

# Configuring and Troubleshooting Cisco Network–Layer Encryption: Background – Part 1

Document ID: 17584

---

## Introduction

### Prerequisites

- Requirements
- Components Used
- Conventions

### Network Layer Encryption Background Information and Configuration

- Cryptography Background
- Definitions
- Preliminary Information
- Caveats

### Cisco IOS Network–Layer Encryption Configuration

- Step 1: Manually Generate DSS Key Pairs
- Step 2: Manually Exchange DSS Public Keys with Peers (out-of-band)
- Sample 1: Cisco IOS Configuration for Dedicated Link
- Sample 2: Cisco IOS Configuration for Multipoint Frame Relay
- Sample 3: Encryption To and Through a Router
- Sample 4: Crypto with DDR
- Sample 5: Encryption of IPX Traffic in an IP Tunnel
- Sample 6: Encrypting L2F Tunnels
- Troubleshooting
- Troubleshooting Cisco 7200 with ESA
- Troubleshooting VIP2 With ESA

### Related Information

---

## Introduction

This document discusses configuring and troubleshooting Cisco Network–Layer Encryption with IPsec and Internet Security Association and Key Management Protocol (ISAKMP) and covers Network–Layer Encryption background information and basic configuration along with IPsec and ISAKMP.

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

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

- Cisco IOS® Software Release 11.2 and later

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

## Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

# Network Layer Encryption Background Information and Configuration

The Network-Layer Encryption feature was introduced in Cisco IOS® Software Release 11.2. It provides a mechanism for secure data transmission and consists of two components:

- **Router Authentication:** Prior to passing encrypted traffic, two routers perform a one-time, two-way authentication using Digital Signature Standard (DSS) public keys to sign random challenges.
- **Network-Layer Encryption:** For IP payload encryption, the routers use Diffie-Hellman key exchange to securely generate a DES(40- or 56-bit session key), Triple DES – 3DES(168-bit), or the more recent Advanced Encryption Standard – AES(128-bit(default), or 192-bit, or 256-bit key), introduced in 12.2(13)T. New session keys are generated on a configurable basis. Encryption policy is set by crypto-maps that use extended IP access lists to define which network, subnet, host, or protocol pairs are to be encrypted between routers.

## Cryptography Background

The field of cryptography is concerned with keeping communications private. The protection of sensitive communications has been the emphasis of cryptography throughout much of its history. Encryption is the transformation of data into some unreadable form. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended, even if they can see the encrypted data. Decryption is the reverse of encryption: it is the transformation of encrypted data back into an intelligible form.

Encryption and decryption require the use of some secret information, usually referred to as a "key". Depending on the encryption mechanism used, the same key might be used for both encryption and decryption; while for other mechanisms, the keys used for encryption and decryption might be different.

A digital signature binds a document to the possessor of a particular key, while a digital timestamp binds a document to its creation at a particular time. These cryptographic mechanisms can be used to control access to a shared disk drive, a high-security installation, or to a pay-per-view television channel.

While modern cryptography is growing increasingly diverse, cryptography is fundamentally based on problems that are difficult to solve. A problem may be difficult because its solution requires knowing the key, such as decrypting an encrypted message or signing some digital document. The problem may also be difficult because it is intrinsically difficult to complete, such as finding a message which produces a given hash value.

As the field of cryptography has advanced, the dividing lines for what is and what is not cryptography have become blurred. Cryptography today might be summed up as the study of techniques and applications that depend on the existence of mathematical problems that are difficult to solve. A cryptanalyst attempts to compromise cryptographic mechanisms, and cryptology is the discipline of cryptography and cryptanalysis combined.

## Definitions

This section defines the related terms used throughout this document.

- **Authentication:** The property of knowing that the data received is actually sent by the claimed sender.

- **Confidentiality:** The property of communicating so that the intended recipients know what is being sent but unintended parties cannot determine what is sent.
- **Data Encryption Standard (DES):** The DES utilizes a symmetric key method, also known as a secret key method. This means that if a block of data is encrypted with the key, the encrypted block must be decrypted with the same key, so both the encryptor and the decrypter must use the same key. Even though the encryption method is known and well published, the best publicly known attack method is through brute force. Keys must be tested against the encrypted blocks to see if they can correctly resolve them. As processors become more powerful, the natural life of DES is nearing its end. For instance, a coordinated effort using spare processing power from thousands of computers across the Internet is able to find the 56-bit key to a DES encoded message in 21 days.

DES is validated every five years by the US National Security Agency (NSA) for meeting the purposes of the US Government. The current approval expires in 1998 and the NSA has indicated that they will not re-certify DES. Moving beyond DES, there are other encryption algorithms which also do not have any known weaknesses other than brute force attacks. For additional information, see DES FIPS 46-2 by the National Institute of Standards and Technology (NIST) .

- **Decryption:** The reverse application of an encryption algorithm to encrypted data, thereby restoring that data to its original, unencrypted state.
- **DSS and Digital Signature Algorithm (DSA):** The DSA was published by the NIST in the Digital Signature Standard (DSS), which is a part of the U.S. government's Capstone project. DSS was selected by NIST, in cooperation with the NSA, to be the digital authentication standard of the U.S. government. The standard was issued on May 19, 1994.
- **Encryption:** The application of a specific algorithm to data so as to alter the appearance of the data making it incomprehensible to those who are not authorized to see the information.
- **Integrity:** The property of ensuring that data is transmitted from source to destination without undetected alteration.
- **Non-repudiation:** The property of a receiver being able to prove that the sender of some data did in fact send the data even though the sender might later desire to deny ever having sent that data.
- **Public Key Cryptography:** Traditional cryptography is based on the sender and receiver of a message knowing and using the same secret key. The sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as "secret-key" or "symmetric cryptography." The main issue is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, or a phone system, or some other transmission medium to prevent the disclosure of the secret key being communicated. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. The generation, transmission, and storage of keys is called key management; all cryptosystems must deal with key management issues. Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users.

The concept of public-key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman in order to solve the key management problem. In their concept, each person gets a pair of keys, one called the public key and the other called the private key. Each person's public key is published while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated and all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal. The only requirement is that public keys are associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message simply by using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public-key cryptography can be used not only for privacy (encryption), but for authentication (digital signatures) as well.

- **Public Key Digital Signatures:** To sign a message, a person performs a computation involving both their private key and the message itself. The output is called the digital signature and is attached to the message, which is then sent. A second person verifies the signature by performing a computation involving the message, the purported signature, and the first person's public key. If the result properly holds in a simple mathematical relation, the signature is verified as being genuine. Otherwise, the signature may be fraudulent or the message might have been altered.
- **Public Key Encryption:** When one person wishes to send a secret message to another person, the first person looks up the second person's public key in a directory, uses it to encrypt the message and sends it off. The second person then uses their private key to decrypt the message and read it. No one listening in can decrypt the message. Anyone can send an encrypted message to the second person but only the second person can read it. Clearly, one requirement is that no one can figure out the private key from the corresponding public key.
- **Traffic Analysis:** The analysis of network traffic flow for the purpose of deducing information that is useful to an adversary. Examples of such information are frequency of transmission, the identities of the conversing parties, sizes of packets, Flow Identifiers used, and so on.

## Preliminary Information

This section discusses some basic Network-Layer Encryption concepts. It contains the aspects of encryption that you should look out for. Initially, these issues may not make sense to you, but it is a good idea to read them over now and be aware of them because they will make more sense after you have worked with encryption for several months.

- It is important to note that encryption occurs only on the output of an interface and decryption occurs only upon input to the interface. This distinction is important when planing your policy. The policy for encryption and decryption is symmetrical. This means that defining one gives you the other automatically. With the crypto maps and their associated extended access lists, only the encryption policy is explicitly defined.

Decryption policy uses the identical information, but when matching packets, it reverses source and destination addresses and ports. This way, the data is protected in both directions of a duplex connection. The *match address x* statement in the **crypto map** command is used to describe packets leaving an interface. In other words, it is describing the encryption of packets. However, packets must also be matched for decryption as they enter the interface. This is done automatically by traversing the access list with the source and destination addresses and ports reversed. This provides symmetry for the connection. The access list pointed to by the **crypto map** should describe traffic in one (outbound) direction only. IP packets not matching the access list you define will be transmitted but not encrypted. A "deny" in the access list indicates that those hosts should not be matched, which means they will not be encrypted. The "deny", in this context, does not mean that the packet is dropped.

- Be very careful of using the word "any" in extended access lists. Using "any" causes your traffic to be dropped unless it is headed to the matching "un-encrypting" interface. In addition, with the IPSec in Cisco IOS Software Release 11.3(3)T, "any" is not allowed.
- Use of the "any" keyword is discouraged in specifying source or destination addresses. Specifying "any" can cause problems with routing protocols, Network Time Protocol (NTP), echo, echo response, and multicast traffic, as the receiving router silently discards this traffic. If "any" is to be used, it should be preceded by "deny" statements for traffic that is not to be encrypted, such as "ntp".
- To save time, make sure you can **ping** the peer router with which you are trying to have an encryption association. Also, have the end devices (that depend on getting their traffic encrypted) ping each other before you spend too much time troubleshooting the wrong problem. In other words, make sure the routing works before trying to do **crypto**. The remote peer may not have a route for the egress interface, in which case you are not able to have an encryption session with that peer (you may be able to use **ip unnumbered** on that serial interface).
- Many WAN point-to-point links use non-routable IP addresses, and Cisco IOS Software Release 11.2 Encryption relies on Internet Control Message Protocol (ICMP) (meaning that it uses the egress

serial interface's IP address for ICMP). This may force you to use **ip unnumbered** on the WAN interface. Always do a **ping** and **traceroute** command to make sure that routing is in place for the two peering (encrypting/decrypting) routers.

- Only two routers are allowed to share a Diffie–Hellman session key. That is, one router cannot exchange encrypted packets to two peers using the same session key; each pair of routers must have a session key that is a result of a Diffie–Hellman exchange between them.
- The crypto engine is either in Cisco IOS, the VIP2 Cisco IOS, or in hardware the Encryption services adapter (ESA) on a VIP2. Without a VIP2, the Cisco IOS crypto engine governs encryption policy on all ports. On platforms using the VIP2, there are multiple crypto engines: one in the Cisco IOS, and one on each VIP2. The crypto engine on a VIP2 governs encryption on the ports that reside on the board.
- Make sure that traffic is set to arrive at an interface prepared to encrypt it. If the traffic can somehow arrive on an interface other than the one with **crypto map** applied, it is silently dropped.
- It helps to have console (or alternate) access to both routers when doing key exchange; it is possible to get the passive side to hang while waiting for a key.
- The **cfb–64** is more efficient to process than **cfb–8** in terms of CPU load.
- The router needs to be running the algorithm that you want to use with the cipher–feedback (CFB) mode that you want to use; defaults for each image are the image name (such as "56") with **cfb–64**.
- Consider changing the key–timeout. The 30–minute default is very short. Try increasing it to one day (1440 minutes).
- IP traffic is dropped during key re–negotiation each time the key expires.
- Select only the traffic that you really want to encrypt (this saves CPU cycles).
- With dial–on–demand routing (DDR), make ICMP interesting or it will never dial out.
- If you want to encrypt traffic other than IP, use a tunnel. With tunnels, apply the crypto maps to both the physical and tunnel interfaces. See Sample 5: Encryption of IPX Traffic in an IP Tunnel for more information.
- The two encryption peer routers do not need to be directly connected.
- A low–end router may give you a "CPU hog" message. This can be ignored because it is telling you that encryption uses a lot of CPU resources.
- Do not place encrypting routers redundantly so that you de–crypt and re–encrypt traffic and waste CPU. Simply encrypt at the two end–points. See Sample 3: Encryption To and Through a Router for more information.
- Currently, encryption of broadcast and multicast packets is not supported. If "secure" routing updates are important to a network design, a protocol with authentication built in should be used, such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), or Routing Information Protocol Version 2 (RIPv2) to ensure update integrity.

## Caveats

**Note:** The Caveats mentioned below have all been resolved.

- A Cisco 7200 router using an ESA for encryption cannot decrypt a packet under one session key and then re–encrypt it under a different session key. Refer to Cisco bug ID CSCdj82613 ( registered customers only) .
- When two routers are connected by an encrypted leased line and an ISDN backup line, if the leased line drops, the ISDN link comes up fine. However, when the leased line comes back up again, the router that placed the ISDN call crashes. Refer to Cisco bug ID CSCdj00310 ( registered customers only) .
- For Cisco 7500 series routers with multiple VIPs, if a **crypto map** is applied to even one interface of any VIP, one or more VIPs crash. Refer to Cisco bug ID CSCdi88459 ( registered customers only) .
- For Cisco 7500 series routers with a VIP2 and ESA, the **show crypto card** command does not display output unless the user is at the console port. Refer to Cisco bug ID CSCdj89070 ( registered customers only) .

# Cisco IOS Network-Layer Encryption Configuration

The working sample Cisco IOS configurations in this document came directly from lab routers. The only alteration made to them was the removal of unrelated interface configurations. All of the material here came from freely available resources on the Internet or in the Related Information section at the end of this document.

All of the sample configurations in this document are from Cisco IOS Software Release 11.3. There were several changes from the Cisco IOS Software Release 11.2 commands, such as the addition of the following words:

- dss in some of the key configuration commands.
- cisco in some of the **show** commands and the **crypto map** commands to distinguish between Cisco's proprietary encryption (as found in Cisco IOS Software Release 11.2 and later) and IPsec which is in Cisco IOS Software Release 11.3(2)T.

**Note:** The IP addresses used in these configuration examples were chosen randomly in Cisco's lab and are intended to be completely generic.

## Step 1: Manually Generate DSS Key Pairs

A DSS key pair (a public and private key) needs to be manually generated on each router participating in the encryption session. In other words, every router must have its own DSS keys in order to participate. An encryption engine can have only one DSS key that uniquely identifies it. The keyword "dss" was added in Cisco IOS Software Release 11.3 in order to distinguish DSS from RSA keys. You can specify any name for the router's own DSS keys (although, it is recommended to use the router hostname). On a less powerful CPU (such as the Cisco 2500 series), key pair generation takes about 5 seconds or less.

The router generates a pair of keys:

- A public key (which is later sent to routers participating in encryption sessions).
- A private key (which is not seen nor exchanged with anyone else; in fact, it is stored in a separate section of NVRAM that cannot be viewed).

Once the router's DSS key pair has been generated, it is uniquely associated with the crypto engine in that router. Key pair generation is shown in the example command output below.

```
dial-5(config)#crypto key generate dss dial5
Generating DSS keys ....
[OK]

dial-5#show crypto key mypubkey dss
crypto public-key dial5 05679919
 160AA490 5B9B1824 24769FCD EE5E0F46 1ABBD343 4C0C4A03 4B279D6B 0EE5F65F
 F64665D4 1036875A 8CF93691 BDF81722 064B51C9 58D72E12 3E1894B6 64B1D145
quit

dial-5#show crypto engine configuration
slot:                0
engine name:         dial5
engine type:         software
serial number:       05679919
platform:            rp crypto engine
crypto lib version:  10.0.0

Encryption Process Info:
```

```
input queue top:    43
input queue bot:   43
input queue count:  0
```

```
dial-5#
```

Because you can generate only one key pair that identifies the router, you may overwrite your original key and need to re-send your public key with every router in the encryption association. This is shown in the example command output below:

```
StHelen(config)#crypto key generate dss barney
% Generating new DSS keys will require re-exchanging
  public keys with peers who already have the public key
  named barney!
Generate new DSS keys? [yes/no]: yes
Generating DSS keys ....
[OK]
```

```
StHelen(config)#
Mar 16 12:13:12.851: Crypto engine 0: create key pairs.
```

## Step 2: Manually Exchange DSS Public Keys with Peers (out-of-band)

Generating the router's own DSS key pair is the first step in establishing an encryption session association. The next step is to exchange public keys with every other router. You can enter these public keys manually by first entering the **show crypto mypubkey** command to display the router's DSS public key. You then exchange these public keys (via email, for example) and, with the **crypto key pubkey-chain dss** command, cut and paste your peer router's public key into the router.

You can also use the **crypto key exchange dss** command to have the routers exchange public keys automatically. If you use the automated method, make sure there are no **crypto map** statements on the interfaces used for the key exchange. A **debug crypto key** is useful here.

**Note:** It is a good idea to **ping** your peer before trying to exchange keys.

```
Loser#ping 19.19.19.20
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 19.19.19.20, timeout is 2 seconds:
```

```
!!!!!
```

```
Loser(config)#crypto key exchange dss passive
```

```
Enter escape character to abort if connection does not complete.
```

```
Wait for connection from peer[confirm]
```

```
Waiting ....
```

```
StHelen(config)#crypto key exchange dss 19.19.19.19 barney
```

```
Public key for barney:
```

```
Serial Number 05694352
```

```
Fingerprint   309E D1DE B6DA 5145 D034
```

```
Wait for peer to send a key[confirm]
```

```
Public key for barney:
```

```
Serial Number 05694352
```

```
Fingerprint   309E D1DE B6DA 5145 D034
```

```
Add this public key to the configuration? [yes/no]:yes
```

```
Mar 16 12:16:55.343: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:55.343: CRYPTO-KE: Sent 4 bytes.
Mar 16 12:16:55.343: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:55.347: CRYPTO-KE: Sent 64 bytes.
```

```
Mar 16 12:16:45.099: CRYPTO-KE: Received 4 bytes.
Mar 16 12:16:45.099: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:45.103: CRYPTO-KE: Received 6 bytes.
Mar 16 12:16:45.103: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:45.107: CRYPTO-KE: Received 50 bytes.
Mar 16 12:16:45.111: CRYPTO-KE: Received 14 bytes.
```

```
Send peer a key in return[confirm]
Which one?
```

```
fred? [yes]:
Public key for fred:
  Serial Number 02802219
  Fingerprint 2963 05F9 ED55 576D CF9D
```

```
Waiting ....
Public key for fred:
  Serial Number 02802219
  Fingerprint 2963 05F9 ED55 576D CF9D
```

```
Add this public key to the configuration? [yes/no]:
```

```
Loser(config)#
Mar 16 12:16:55.339: CRYPTO-KE: Sent 4 bytes.
Mar 16 12:16:55.343: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:55.343: CRYPTO-KE: Sent 4 bytes.
Mar 16 12:16:55.343: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:55.347: CRYPTO-KE: Sent 64 bytes.
Loser(config)#
```

```
Mar 16 12:16:56.083: CRYPTO-KE: Received 4 bytes.
Mar 16 12:16:56.087: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:56.087: CRYPTO-KE: Received 4 bytes.
Mar 16 12:16:56.091: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:56.091: CRYPTO-KE: Received 52 bytes.
Mar 16 12:16:56.095: CRYPTO-KE: Received 12 bytes.
Add this public key to the configuration? [yes/no]: yes
StHelen(config)#^Z
StHelen#
```

Now that public DSS keys have been exchanged, make sure that both routers have each other's public keys and that they match, as shown in the command output below.

```
Loser#show crypto key mypubkey dss
crypto public-key fred 02802219
  79CED212 AF191D29 702A9301 B3E06602 D4FB26B3 316E58C8 05D4930C CE891810
  C0064492 5F6684CD 3FC326E5 679BCA46 BB155402 D443F68D 93487F7E 5ABE182E
quit
```

```
Loser#show crypto key pubkey-chain dss
crypto public-key barney 05694352
  B407A360 204CBFA3 F9A0C0B0 15D6185D 91FD7D3A 3232EBA2 F2D31D21 53AE24ED
  732EA43D 484DEB22 6E91515C 234B4019 38E51D64 04CB9F59 EE357477 91810341
quit
```

```
-----
```

```
StHelen#show crypto key mypubkey dss
crypto public-key barney 05694352
  B407A360 204CBFA3 F9A0C0B0 15D6185D 91FD7D3A 3232EBA2 F2D31D21 53AE24ED
  732EA43D 484DEB22 6E91515C 234B4019 38E51D64 04CB9F59 EE357477 91810341
quit
```

```
StHelen#show crypto key pubkey-chain dss
crypto public-key fred 02802219
  79CED212 AF191D29 702A9301 B3E06602 D4FB26B3 316E58C8 05D4930C CE891810
  C0064492 5F6684CD 3FC326E5 679BCA46 BB155402 D443F68D 93487F7E 5ABE182E
quit
```

## Sample 1: Cisco IOS Configuration for Dedicated Link

After the DSS keys have been generated on each router and the DSS public keys have been exchanged, the **crypto map** command can be applied to the interface. The crypto session begins by generating traffic that matches the access list used by the crypto maps.

```
Loser#write terminal
Building configuration...

Current configuration:
!
! Last configuration change at 13:01:18 UTC Mon Mar 16 1998
! NVRAM config last updated at 13:03:02 UTC Mon Mar 16 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname Loser
!
enable secret 5 $1$AeuFSMx7O/DhpqjLKc2VQVbeC0
!
ip subnet-zero
no ip domain-lookup
crypto map oldstyle 10
  set peer barney
  match address 133
!
crypto key pubkey-chain dss
  named-key barney
  serial-number 05694352
  key-string
    B407A360 204CBFA3 F9A0C0B0 15D6185D 91FD7D3A 3232EBA2 F2D31D21 53AE24ED
    732EA43D 484DEB22 6E91515C 234B4019 38E51D64 04CB9F59 EE357477 91810341
  quit
!
interface Ethernet0
  ip address 40.40.40.41 255.255.255.0
  no ip mroute-cache
!
interface Serial0
  ip address 18.18.18.18 255.255.255.0
  encapsulation ppp
  no ip mroute-cache
  shutdown
!
interface Serial1
  ip address 19.19.19.19 255.255.255.0
  encapsulation ppp
  no ip mroute-cache
  clockrate 2400
  no cdp enable
```

```
crypto map oldstyle
!
ip default-gateway 10.11.19.254
ip classless
ip route 0.0.0.0 0.0.0.0 19.19.19.20
access-list 133 permit ip 40.40.40.0 0.0.0.255 30.30.30.0 0.0.0.255
!
line con 0
  exec-timeout 0 0
line aux 0
  no exec
  transport input all
line vty 0 4
  password ww
  login
!
end
```

Loser#

```
-----
StHelen#write terminal
Building configuration...
```

Current configuration:

```
!
!Last configuration change at 13:03:05 UTC Mon Mar 16 1998
! NVRAM config last updated at 13:03:07 UTC Mon Mar 16 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname StHelen
!
boot system flash c2500-is56-1
enable password ww
!
partition flash 2 8 8
!
no ip domain-lookup
crypto map oldstyle 10
  set peer fred
  match address 144
!
crypto key pubkey-chain dss
  named-key fred
    serial-number 02802219
    key-string
      79CED212 AF191D29 702A9301 B3E06602 D4FB26B3 316E58C8 05D4930C CE891810
      C0064492 5F6684CD 3FC326E5 679BCA46 BB155402 D443F68D 93487F7E 5ABE182E
    quit
!
!
interface Ethernet0
  ip address 30.30.30.31 255.255.255.0
!
interface Ethernet1
  no ip address
  shutdown
!
interface Serial0
  no ip address
  encapsulation x25
  no ip mroute-cache
  shutdown
```

```

!
interface Serial1
 ip address 19.19.19.20 255.255.255.0
 encapsulation ppp
 no ip mroute-cache
 load-interval 30
 compress stac
 no cdp enable
 crypto map oldstyle
!
ip default-gateway 10.11.19.254
ip classless
ip route 0.0.0.0 0.0.0.0 19.19.19.19
access-list 144 permit ip 30.30.30.0 0.0.0.255 40.40.40.0 0.0.0.255
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password ww
 login
!
end

StHelen#

```

## Sample 2: Cisco IOS Configuration for Multipoint Frame Relay

The following sample command output was taken from the HUB Router.

```

Loser#write terminal
Building configuration...

Current configuration:
!
! Last configuration change at 10:45:20 UTC Wed Mar 11 1998
! NVRAM config last updated at 18:28:27 UTC Tue Mar 10 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname Loser
!
enable secret 5 $1$AeuFSMx7O/DhpqjLKc2VQVbeC0
!
ip subnet-zero
no ip domain-lookup
!
crypto map oldstuff 10
 set peer barney
 match address 133
crypto map oldstuff 20
 set peer wilma
 match address 144
!
crypto key pubkey-chain dss
 named-key barney
 serial-number 05694352
 key-string
 1D460DC3 BDC73312 93B7E220 1861D55C E00DA5D8 DB2B04CD FABD297C 899D40E7
 D284F07D 6EEC83B8 E3676EC2 D813F7C8 F532DC7F 0A9913E7 8A6CB7E9 BE18790D
 quit
 named-key wilma

```

```

serial-number 01496536
key-string
  C26CB3DD 2A56DD50 CC2116C9 2697CE93 6DBFD824 1889F791 9BF36E70 7B29279C
  E343C56F 32266443 989B4528 1CF32C2D 9E3F2447 A5DBE054 879487F6 26A55939
quit
!
crypto cisco pregen-dh-pairs 5
!
crypto cisco key-timeout 1440
!
interface Ethernet0
 ip address 190.190.190.190 255.255.255.0
 no ip mroute-cache
!
interface Serial1
 ip address 19.19.19.19 255.255.255.0
 encapsulation frame-relay
 no ip mroute-cache
 clockrate 500000
 crypto map oldstuff
!
!
ip default-gateway 10.11.19.254
ip classless
ip route 200.200.200.0 255.255.255.0 19.19.19.20
ip route 210.210.210.0 255.255.255.0 19.19.19.21
access-list 133 permit ip 190.190.190.0 0.0.0.255 200.200.200.0 0.0.0.255
access-list 144 permit ip 190.190.190.0 0.0.0.255 210.210.210.0 0.0.0.255
!
line con 0
 exec-timeout 0 0
line aux 0
 no exec
 transport input all
line vty 0 4
 password ww
 login
!
end

Loser#

```

The following sample command output was taken from Remote Site A.

```

WAN-2511a#write terminal
Building configuration...

Current configuration:
!
version 11.3
no service password-encryption
!
hostname WAN-2511a
!
enable password ww
!
no ip domain-lookup
!
crypto map mymap 10
 set peer fred
 match address 133
!
crypto key pubkey-chain dss
 named-key fred
  serial-number 02802219
  key-string

```

```

56841777 4F27A574 5005E0F0 CF3C33F5 C6AAD000 5518A8FF 7422C592 021B295D
D95AAB73 01235FD8 40D70284 3A63A38E 216582E8 EC1F8B0D 0256EFF5 0EE89436
quit
!
interface Ethernet0
ip address 210.210.210.210 255.255.255.0
shutdown
!
interface Serial0
ip address 19.19.19.21 255.255.255.0
encapsulation frame-relay
no fair-queue
crypto map mymap
!
ip default-gateway 10.11.19.254
ip classless
ip route 190.190.190.0 255.255.255.0 19.19.19.19
access-list 133 permit ip 210.210.210.0 0.0.0.255 190.190.190.0 0.0.0.255
!
line con 0
exec-timeout 0 0
line 1
no exec
transport input all
line 2 16
no exec
line aux 0
line vty 0 4
password ww
login
!
end

WAN-2511a#

```

The following sample command output was taken from Remote Site B.

```

StHelen#write terminal
Building configuration...

Current configuration:
!
! Last configuration change at 19:00:34 UTC Tue Mar 10 1998
! NVRAM config last updated at 18:48:39 UTC Tue Mar 10 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname StHelen
!
boot system flash c2500-is56-1
enable password ww
!
partition flash 2 8 8
!
no ip domain-lookup
!
crypto map wabba 10
set peer fred
match address 144
!
crypto key pubkey-chain dss
named-key fred
serial-number 02802219
key-string

```

```

56841777 4F27A574 5005E0F0 CF3C33F5 C6AAD000 5518A8FF 7422C592 021B295D
D95AAB73 01235FD8 40D70284 3A63A38E 216582E8 EC1F8B0D 0256EFF5 0EE89436
quit
!
interface Ethernet0
ip address 200.200.200.200 255.255.255.0
!
interface Serial1
ip address 19.19.19.20 255.255.255.0
encapsulation frame-relay
no ip mroute-cache
crypto map wabba
!
ip default-gateway 10.11.19.254
ip classless
ip route 190.190.190.0 255.255.255.0 19.19.19.19
access-list 144 permit ip 200.200.200.0 0.0.0.255 190.190.190.0 0.0.0.255
!
line con 0
exec-timeout 0 0
line aux 0
transport input all
line vty 0 4
password ww
login
!
end

StHelen#

```

The following sample command output was taken from the Frame Relay switch.

```

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname wan-4700a
!
enable password ww
!
no ip domain-lookup
frame-relay switching
!
interface Serial0
no ip address
encapsulation frame-relay
clockrate 500000
frame-relay intf-type dce
frame-relay route 200 interface Serial1 100
!
interface Serial1
no ip address
encapsulation frame-relay
frame-relay intf-type dce
frame-relay route 100 interface Serial0 200
frame-relay route 300 interface Serial2 200
!
interface Serial2
no ip address
encapsulation frame-relay
clockrate 500000
frame-relay intf-type dce
frame-relay route 200 interface Serial1 300

```



```

ip address 18.18.18.19 255.255.255.0
encapsulation ppp
crypto map toworld
!
router rip
network 18.0.0.0
network 180.180.0.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 30.30.30.31
ip route 171.68.118.0 255.255.255.0 10.11.19.254
access-list 133 permit ip 180.180.180.0 0.0.0.255 40.40.40.0 0.0.0.255
access-list 144 permit ip 180.180.180.0 0.0.0.255 30.30.30.0 0.0.0.255
!
line con 0
exec-timeout 0 0
line aux 0
password 7 044C1C
line vty 0 4
login local
!
end

```

wan-4500b#

-----

Loser#**write terminal**  
Building configuration...

Current configuration:

```

!
! Last configuration change at 11:01:54 UTC Wed Mar 18 1998
! NVRAM config last updated at 11:09:59 UTC Wed Mar 18 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname Loser
!
enable secret 5 $1$AeuFSMx7O/DhpqjLKc2VQVbeC0
!
ip subnet-zero
no ip domain-lookup
ip host StHelen.cisco.com 19.19.19.20
ip domain-name cisco.com
!
crypto map towan 10
set peer wan
match address 133
!
crypto key pubkey-chain dss
named-key wan
serial-number 07365004
key-string
A547B701 4312035D 2FC7D0F4 56BC304A 59FA76C3 B9762E4A F86DED86 3830E66F
2ED5C476 CFF234D3 3842BC98 3CA4A5FB 9089556C 7464D2B4 AF7E6AEB 86269A5B
quit
!
interface Ethernet0
ip address 40.40.40.40 255.255.255.0
no ip mroute-cache
!
interface Serial0
ip address 18.18.18.18 255.255.255.0

```

```
encapsulation ppp
no ip mroute-cache
clockrate 64000
crypto map towan
!
interface Serial1
ip address 19.19.19.19 255.255.255.0
encapsulation ppp
no ip mroute-cache
priority-group 1
clockrate 64000
!
!
router rip
network 19.0.0.0
network 18.0.0.0
network 40.0.0.0
!
ip default-gateway 10.11.19.254
ip classless
access-list 133 permit ip 40.40.40.0 0.0.0.255 180.180.180.0 0.0.0.255
!
line con 0
exec-timeout 0 0
line aux 0
no exec
transport input all
line vty 0 4
password ww
login
!
end
```

Loser#

-----  
StHelen#**write terminal**  
Building configuration...

Current configuration:

```
!
! Last configuration change at 11:13:18 UTC Wed Mar 18 1998
! NVRAM config last updated at 11:21:30 UTC Wed Mar 18 1998
!
version 11.3
service timestamps debug datetime msec
no service password-encryption
!
hostname StHelen
!
boot system flash c2500-is56-1
enable password ww
!
partition flash 2 8 8
!
no ip domain-lookup
!
crypto map towan 10
set peer wan
match address 144
!
crypto key pubkey-chain dss
named-key wan
serial-number 07365004
key-string
```

```

A547B701 4312035D 2FC7D0F4 56BC304A 59FA76C3 B9762E4A F86DED86 3830E66F
2ED5C476 CFF234D3 3842BC98 3CA4A5FB 9089556C 7464D2B4 AF7E6AEB 86269A5B
quit
!
interface Ethernet0
no ip address
!
interface Ethernet1
ip address 30.30.30.30 255.255.255.0
!
interface Serial1
ip address 19.19.19.20 255.255.255.0
encapsulation ppp
no ip mroute-cache
load-interval 30
crypto map towan
!
router rip
network 30.0.0.0
network 19.0.0.0
!
ip default-gateway 10.11.19.254
ip classless
access-list 144 permit ip 30.30.30.0 0.0.0.255 180.180.180.0 0.0.0.255
!
line con 0
exec-timeout 0 0
line aux 0
transport input all
line vty 0 4
password ww
login
!
end

StHelen#

```

```

-----
wan-4500b#show crypto cisco algorithms
des cfb-64
40-bit-des cfb-64

```

```

wan-4500b#show crypto cisco key-timeout
Session keys will be re-negotiated every 30 minutes

```

```

wan-4500b#show crypto cisco pregen-dh-pairs
Number of pregenerated DH pairs: 0

```

```

wan-4500b#show crypto engine connections active

```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	Serial0	18.18.18.19	set	DES_56_CFB64	1683	1682
5	Serial0	18.18.18.19	set	DES_56_CFB64	1693	1693

```

wan-4500b#show crypto engine connections dropped-packet

```

Interface	IP-Address	Drop Count
Serial0	18.18.18.19	52

```

wan-4500b#show crypto engine configuration
slot: 0
engine name: wan
engine type: software
serial number: 07365004
platform: rp crypto engine
crypto lib version: 10.0.0

```

Encryption Process Info:

input queue top: 303  
input queue bot: 303  
input queue count: 0

wan-4500b#show crypto key mypubkey dss

crypto public-key wan 07365004  
A547B701 4312035D 2FC7D0F4 56BC304A 59FA76C3 B9762E4A F86DED86 3830E66F  
2ED5C476 CFF234D3 3842BC98 3CA4A5FB 9089556C 7464D2B4 AF7E6AEB 86269A5B  
quit

wan-4500b#show crypto key pubkey-chain dss

crypto public-key loser 02802219  
F0BE2128 752D1A24 F394B355 3216BA9B 7C4E8677 29C176F9 A047B7D9 7D03BDA4  
6B7AFDC2 2DAEF3AB 393EE7C7 802C1A95 B40031D1 908004F9 8A33A352 FF19BC24  
quit

crypto public-key sthelen 05694352

5C401002 404DC5A9 EAED2360 D7007E51 4A4BB8F8 6F9B1554 51D8ACBB D3964C10  
A23848CA 46003A94 2FC8C7D6 0B57AE07 9EB5EF3A BD71482B 052CF06B 90C3C618  
quit

wan-4500b#show crypto map interface serial 1

No crypto maps found.

wan-4500b#show crypto map

Crypto Map "toworld" 10 cisco  
Connection Id = 1 (1 established, 0 failed)  
Peer = loser  
PE = 180.180.180.0  
UPE = 40.40.40.0  
Extended IP access list 133  
access-list 133 permit ip  
source: addr = 180.180.180.0/0.0.0.255  
dest: addr = 40.40.40.0/0.0.0.255

Crypto Map "toworld" 20 cisco  
Connection Id = 5 (1 established, 0 failed)  
Peer = sthelen  
PE = 180.180.180.0  
UPE = 30.30.30.0  
Extended IP access list 144  
access-list 144 permit ip  
source: addr = 180.180.180.0/0.0.0.255  
dest: addr = 30.30.30.0/0.0.0.255

wan-4500b#

-----  
Loser#show crypto cisco algorithms

des cfb-64  
des cfb-8  
40-bit-des cfb-64  
40-bit-des cfb-8

Loser#show crypto cisco key-timeout

Session keys will be re-negotiated every 30 minutes

Loser#show crypto cisco pregen-dh-pairs

Number of pregenerated DH pairs: 10

Loser#show crypto engine connections active

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
61	Serial0	18.18.18.18	set	DES_56_CFB64	1683	1682

Loser#show crypto engine connections dropped-packet

Interface IP-Address Drop Count

Serial0 18.18.18.18 1  
Serial1 19.19.19.19 90

Loser#**show crypto engine configuration**

slot: 0  
engine name: loser  
engine type: software  
serial number: 02802219  
platform: rp crypto engine  
crypto lib version: 10.0.0

Encryption Process Info:

input queue top: 235  
input queue bot: 235  
input queue count: 0

Loser#**show crypto key mypubkey dss**

crypto public-key loser 02802219  
F0BE2128 752D1A24 F394B355 3216BA9B 7C4E8677 29C176F9 A047B7D9 7D03BDA4  
6B7AFDC2 2DAEF3AB 393EE7C7 802C1A95 B40031D1 908004F9 8A33A352 FF19BC24  
quit

Loser#**show crypto key pubkey-chain dss**

crypto public-key wan 07365004  
A547B701 4312035D 2FC7D0F4 56BC304A 59FA76C3 B9762E4A F86DED86 3830E66F  
2ED5C476 CFF234D3 3842BC98 3CA4A5FB 9089556C 7464D2B4 AF7E6AEB 86269A5B  
quit

Loser#**show crypto map interface serial 1**

No crypto maps found.

Loser#**show crypto map**

Crypto Map "towan" 10 cisco  
Connection Id = 61 (0 established, 0 failed)  
Peer = wan  
PE = 40.40.40.0  
UPE = 180.180.180.0  
Extended IP access list 133  
access-list 133 permit ip  
source: addr = 40.40.40.0/0.0.0.255  
dest: addr = 180.180.180.0/0.0.0.255

Loser#

-----  
StHelen#**show crypto cisco algorithms**

des cfb-64

StHelen#**show crypto cisco key-timeout**

Session keys will be re-negotiated every 30 minutes

StHelen#**show crypto cisco pregen-dh-pairs**

Number of pregenerated DH pairs: 10

StHelen#**show crypto engine connections active**

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
58	Serial1	19.19.19.20	set	DES_56_CFB64	1694	1693

StHelen#**show crypto engine connections dropped-packet**

Interface	IP-Address	Drop Count
-----------	------------	------------

Ethernet0	0.0.0.0	1
Serial1	19.19.19.20	80

```

StHelen#show crypto engine configuration
slot: 0
engine name: sthelen
engine type: software
serial number: 05694352
platform: rp crypto engine
crypto lib version: 10.0.0

Encryption Process Info:
input queue top: 220
input queue bot: 220
input queue count: 0

StHelen#show crypto key mypubkey dss
crypto public-key sthelen 05694352
 5C401002 404DC5A9 EAED2360 D7007E51 4A4BB8F8 6F9B1554 51D8ACBB D3964C10
 A23848CA 46003A94 2FC8C7D6 0B57AE07 9EB5EF3A BD71482B 052CF06B 90C3C618
quit

StHelen#show crypto key pubkey-chain dss
crypto public-key wan 07365004
 A547B701 4312035D 2FC7D0F4 56BC304A 59FA76C3 B9762E4A F86DED86 3830E66F
 2ED5C476 CFF234D3 3842BC98 3CA4A5FB 9089556C 7464D2B4 AF7E6AEB 86269A5B
quit

StHelen#show crypto map interface serial 1
Crypto Map "towan" 10 cisco
  Connection Id = 58 (1 established, 0 failed)
  Peer = wan
  PE = 30.30.30.0
  UPE = 180.180.180.0
  Extended IP access list 144
    access-list 144 permit ip
      source: addr = 30.30.30.0/0.0.0.255
      dest: addr = 180.180.180.0/0.0.0.255

StHelen#show crypto map
Crypto Map "towan" 10 cisco
  Connection Id = 58 (1 established, 0 failed)
  Peer = wan
  PE = 30.30.30.0
  UPE = 180.180.180.0
  Extended IP access list 144
    access-list 144 permit ip
      source: addr = 30.30.30.0/0.0.0.255
      dest: addr = 180.180.180.0/0.0.0.255

StHelen#

```

## Sample 4: Crypto with DDR

Because Cisco IOS relies on the ICMP to establish encryption sessions, ICMP traffic must be classified as "interesting" in the dialer list when doing encryption over a DDR link.

**Note:** Compression does work in Cisco IOS Software Release 11.3, but it is not very useful for encrypted data. Because the encrypted data is fairly random-looking, compression only slows things down. But you can leave the feature on for non-encrypted traffic.

In some situations, you will want dial backup to the same router. For example, it is useful when users want to protect against the failure of a particular link in their WAN networks. If two interfaces go to the same peer, the same crypto map can be used on both interfaces. The backup interface must be used in order for this feature to function properly. If a backup design has a router dial into a different box, different crypto maps should be created and the peers set accordingly. Again, the **backup interface** command should be used.

```
dial-5#write terminal
Building configuration...
```

```
Current configuration:
```

```
!
version 11.3
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname dial-5
!
boot system c1600-sy56-1 171.68.118.83
enable secret 5 $1$0NelwDbhBdcN6x9Y5gfuMjqh10
!
username dial-6 password 0 cisco
isdn switch-type basic-ni1
!
crypto map dial6 10
  set peer dial6
  match address 133
!
crypto key pubkey-chain dss
  named-key dial6
    serial-number 05679987
    key-string
      753F71AB E5305AD4 3FCDFB6D 47AA2BB5 656BFCAA 53DBE37F 07465189 06E91A82
      2BC91236 13DC4AA8 7EC5B48C D276E5FE 0D093014 6D3061C5 03158820 B609CA7C
    quit
!
interface Ethernet0
  ip address 20.20.20.20 255.255.255.0
!
interface BRI0
  ip address 10.10.10.11 255.255.255.0
  encapsulation ppp
  no ip mroute-cache
  load-interval 30
  dialer idle-timeout 9000
  dialer map ip 10.10.10.10 name dial-6 4724118
  dialer hold-queue 40
  dialer-group 1
  isdn spid1 919472417100 4724171
  isdn spid2 919472417201 4724172
  compress stac
  ppp authentication chap
  ppp multilink
  crypto map dial6
!
ip classless
ip route 40.40.40.0 255.255.255.0 10.10.10.10
access-list 133 permit ip 20.20.20.0 0.0.0.255 40.40.40.0 0.0.0.255
dialer-list 1 protocol ip permit
!
line con 0
  exec-timeout 0 0
line vty 0 4
  password ww
  login
!
end

dial-5#
```

```
-----
```

```
dial-6#write terminal
Building configuration...
```

```
Current configuration:
```

```
!
version 11.3
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname dial-6
!
boot system c1600-sy56-1 171.68.118.83
enable secret 5 $1$VdPYuA/BIVeEm9UAFEm.PPJFc.
!
username dial-5 password 0 cisco
no ip domain-lookup
isdn switch-type basic-nil
!
crypto map dial5 10
set peer dial5
match address 144
!
crypto key pubkey-chain dss
named-key dial5
serial-number 05679919
key-string
  160AA490 5B9B1824 24769FCD EE5E0F46 1ABBD343 4C0C4A03 4B279D6B 0EE5F65F
  F64665D4 1036875A 8CF93691 BDF81722 064B51C9 58D72E12 3E1894B6 64B1D145
quit
!
!
interface Ethernet0
ip address 40.40.40.40 255.255.255.0
!
interface BRI0
ip address 10.10.10.10 255.255.255.0
encapsulation ppp
no ip mroute-cache
dialer idle-timeout 9000
dialer map ip 10.10.10.11 name dial-5 4724171
dialer hold-queue 40
dialer load-threshold 5 outbound
dialer-group 1
isdn spid1 919472411800 4724118
isdn spid2 919472411901 4724119
compress stac
ppp authentication chap
ppp multilink
crypto map dial5
!
ip classless
ip route 20.20.20.0 255.255.255.0 10.10.10.11
access-list 144 permit ip 40.40.40.0 0.0.0.255 20.20.20.0 0.0.0.255
dialer-list 1 protocol ip permit
!
line con 0
exec-timeout 0 0
line vty 0 4
password ww
login
!
end
```

```
dial-6#
```

## Sample 5: Encryption of IPX Traffic in an IP Tunnel

In this example, IPX traffic in an IP tunnel is encrypted.

**Note:** Only traffic in this tunnel (IPX) is encrypted. All other IP traffic is left alone.

```
WAN-2511a#write terminal
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname WAN-2511a
!
enable password ww
!
no ip domain-lookup
ipx routing 0000.0c34.aa6a
!
crypto public-key wan2516 01698232
  B1C127B0 78D79CAA 67ECAD80 03D354B1 9012C80E 0C1266BE 25AEDE60 37A192A2
  B066D299 77174D48 7FBAB5FC 2B60893A 37E5CB7B 62F6D902 9495733B 98046962
quit
!
crypto map wan2516 10
  set peer wan2516
  match address 133
!
!
interface Loopback1
  ip address 50.50.50.50 255.255.255.0
!
interface Tunnell
  no ip address
  ipx network 100
  tunnel source 50.50.50.50
  tunnel destination 60.60.60.60
  crypto map wan2516
!
interface Ethernet0
  ip address 40.40.40.40 255.255.255.0
  ipx network 600
!
interface Serial0
  ip address 20.20.20.21 255.255.255.0
  encapsulation ppp
  no ip mroute-cache
  crypto map wan2516
!
interface Serial1
  no ip address
  shutdown
!
ip default-gateway 10.11.19.254
ip classless
ip route 0.0.0.0 0.0.0.0 20.20.20.20
access-list 133 permit ip host 50.50.50.50 host 60.60.60.60
!
line con 0
  exec-timeout 0 0
  password ww
```

```
login
line 1 16
line aux 0
password ww
login
line vty 0 4
password ww
login
!
end
```

WAN-2511a#

```
-----
WAN-2516a#write terminal
Building configuration...
```

Current configuration:

```
!
version 11.2
no service pad
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname WAN-2516a
!
enable password ww
!
no ip domain-lookup
ipx routing 0000.0c3b.cc1e
!
crypto public-key wan2511 01496536
C8EA7C21 DF3E48F5 C6C069DB 3A5E1B08 8B830AD4 4F1DABCE D62F5F46 ED08C81D
5646DC78 DDC77EFC 823F302A F112AF97 668E39A1 E2FCDC05 545E0529 9B3C9553
quit
!
crypto map wan2511 10
set peer wan2511
match address 144
!
!
hub ether 0 1
link-test
auto-polarity
!
! <other hub interfaces snipped>
!
hub ether 0 14
link-test
auto-polarity
!
interface Loopback1
ip address 60.60.60.60 255.255.255.0
!
interface Tunnell
no ip address
ipx network 100
tunnel source 60.60.60.60
tunnel destination 50.50.50.50
crypto map wan2511
!
interface Ethernet0
ip address 30.30.30.30 255.255.255.0
ipx network 400
```

```

!
interface Serial0
 ip address 20.20.20.20 255.255.255.0
 encapsulation ppp
 clockrate 2000000
 crypto map wan2511
!
interface Serial1
 no ip address
 shutdown
!
interface BRI0
 no ip address
 shutdown
!
ip default-gateway 20.20.20.21
ip classless
ip route 0.0.0.0 0.0.0.0 20.20.20.21
access-list 144 permit ip host 60.60.60.60 host 50.50.50.50
access-list 188 permit gre any any
!
line con 0
 exec-timeout 0 0
 password ww
 login
line aux 0
 password ww
 login
 modem InOut
 transport input all
 flowcontrol hardware
line vty 0 4
 password ww
 login
!
end

```

WAN-2516a#

WAN-2511a#show ipx route

```

Codes: C - Connected primary network, c - Connected secondary network
       S - Static, F - Floating static, L - Local (internal), W - IPXWAN
       R - RIP, E - EIGRP, N - NLSP, X - External, A - Aggregate
       s - seconds, u - uses

```

3 Total IPX routes. Up to 1 parallel paths and 16 hops allowed.

No default route known.

```

C      100 (TUNNEL),      Tu1
C      600 (NOVELL-ETHER), Et0
R      400 [151/01] via   100.0000.0c3b.cc1e, 24s, Tu1

```

WAN-2511a#show crypto engine connections active

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	Serial0	20.20.20.21	set	DES_56_CFB64	207	207

WAN-2511a#ping 400.0000.0c3b.cc1e

Translating "400.0000.0c3b.cc1e"

Type escape sequence to abort.

Sending 5, 100-byte IPX cisco Echoes to 400.0000.0c3b.cc1e, timeout is 2 seconds:

!!!!

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/35/48 ms
```

```
WAN-2511a#show crypto engine connections active
```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	Serial0	20.20.20.21	set	DES_56_CFB64	212	212

```
WAN-2511a#ping 30.30.30.30
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 30.30.30.30, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
```

```
WAN-2511a#show crypto engine connections active
```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
1	Serial0	20.20.20.21	set	DES_56_CFB64	212	212

```
WAN-2511a#
```

## Sample 6: Encrypting L2F Tunnels

In this example, only encrypting L2F traffic for users dialing in is attempted. Here, "user@cisco.com" calls the local Network Access Server (NAS) named "DEMO2" in their city and gets tunnelled to the home gateway CD. All DEMO2 traffic (along with that of other L2F callers) is encrypted. Because L2F uses UDP port 1701, this is how the access list is constructed, determining which traffic is encrypted.

**Note:** If the encryption association is not already set up, meaning the caller is the first person to call in and create the L2F tunnel, the caller may get dropped because of the delay in setting up the encryption association. This may not happen on routers with enough CPU power. Also, you may want to increase the **keytimeout** so that the encryption set-up and tear-down only occurs during off-peak hours.

The following sample command output was taken from the remote NAS.

```
DEMO2#write terminal
Building configuration...

Current configuration:
!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname DEMO2
!
enable password ww
!
username NAS1 password 0 SECRET
username HomeGateway password 0 SECRET
no ip domain-lookup
vpdn enable
vpdn outgoing cisco.com NAS1 ip 20.20.20.20
!
crypto public-key wan2516 01698232
B1C127B0 78D79CAA 67ECAD80 03D354B1 9012C80E 0C1266BE 25AEDE60 37A192A2
B066D299 77174D48 7FBAB5FC 2B60893A 37E5CB7B 62F6D902 9495733B 98046962
quit
!
crypto map vpdn 10
```

```

set peer wan2516
match address 133
!
crypto key-timeout 1440
!
interface Ethernet0
ip address 40.40.40.40 255.255.255.0
!
interface Serial0
ip address 20.20.20.21 255.255.255.0
encapsulation ppp
no ip mroute-cache
crypto map vpdn
!
interface Serial1
no ip address
shutdown
!
interface Group-Async1
no ip address
encapsulation ppp
async mode dedicated
no peer default ip address
no cdp enable
ppp authentication chap pap
group-range 1 16
!
ip default-gateway 10.11.19.254
ip classless
ip route 0.0.0.0 0.0.0.0 20.20.20.20
access-list 133 permit udp host 20.20.20.21 eq 1701
host 20.20.20.20 eq 1701
!
!
line con 0
exec-timeout 0 0
password ww
login
line 1 16
modem InOut
transport input all
speed 115200
flowcontrol hardware
line aux 0
login local
modem InOut
transport input all
flowcontrol hardware
line vty 0 4
password ww
login
!
end

DEMO2#

```

The following sample command output was taken from the Home gateway.

```

CD#write terminal
Building configuration...

Current configuration:
!
version 11.2
no service pad
no service password-encryption

```

```
service udp-small-servers
service tcp-small-servers
!
hostname CD
!
enable password ww
!
username NAS1 password 0 SECRET
username HomeGateway password 0 SECRET
username user@cisco.com password 0 cisco
no ip domain-lookup
vpdn enable
vpdn incoming NAS1 HomeGateway virtual-template 1
!
crypto public-key wan2511 01496536
  C8EA7C21 DF3E48F5 C6C069DB 3A5E1B08 8B830AD4 4F1DABCE D62F5F46 ED08C81D
  5646DC78 DDC77EFC 823F302A F112AF97 668E39A1 E2FCDC05 545E0529 9B3C9553
quit
!
crypto key-timeout 1440
!
crypto map vpdn 10
  set peer wan2511
  match address 144
!
!
hub ether 0 1
  link-test
  auto-polarity
!
interface Loopback0
  ip address 70.70.70.1 255.255.255.0
!
interface Ethernet0
  ip address 30.30.30.30 255.255.255.0
!
interface Virtual-Template1
  ip unnumbered Loopback0
  no ip mroute-cache
  peer default ip address pool default
  ppp authentication chap
!
interface Serial0
  ip address 20.20.20.20 255.255.255.0
  encapsulation ppp
  clockrate 2000000
  crypto map vpdn
!
interface Serial1
  no ip address
  shutdown
!
interface BRI0
  no ip address
  shutdown
!
ip local pool default 70.70.70.2 70.70.70.77
ip default-gateway 20.20.20.21
ip classless
ip route 0.0.0.0 0.0.0.0 20.20.20.21
access-list 144 permit udp host 20.20.20.20 eq 1701 host 20.20.20.21 eq 1701
!
line con 0
  exec-timeout 0 0
  password ww
  login
```

```

line aux 0
  password ww
  login
  modem InOut
  transport input all
  flowcontrol hardware
line vty 0 4
  password ww
  login
!
end

```

## Troubleshooting

It is generally best to begin each troubleshooting session by gathering information using the following **show** commands. An asterisk (\*) indicates an especially useful command. Please also see IP Security Troubleshooting – Understanding and Using debug Commands for additional information.

Certain **show** commands are supported by the Output Interpreter Tool ( registered customers only) , which allows you to view an analysis of **show** command output.

**Note:** Before issuing **debug** commands, please see Important Information on Debug Commands.

Commands	
<b>show crypto cisco algorithms</b>	<b>show crypto cisco key-timeout</b>
<b>show crypto cisco pregen-dh-pairs</b>	<b>* show crypto engine</b>
<b>show crypto engine connections dropped-packet</b>	<b>connections active</b> <b>show crypto engine</b>
<b>show crypto key mypubkey dss</b>	<b>configuration</b> <b>* show crypto key</b> <b>pubkey-chain dss</b>
<b>show crypto map interface serial 1</b>	<b>* show crypto map</b>
<b>debug crypto engine</b>	<b>* debug crypto sess</b>
<b>debug cry key</b>	<b>clear crypto connection</b>
<b>crypto zeroize</b>	<b>no crypto public-key</b>

- **show crypto cisco algorithms**

– You must enable all Data Encryption Standard (DES) algorithms that are 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 attempts 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 fails. If at least one common DES algorithm is enabled at both ends, the encrypted session can proceed.

**Note:** The extra word cisco shows up in Cisco IOS Software Release 11.3 and is needed to distinguish between IPSec and Cisco proprietary encryption found in Cisco IOS Software Release 11.2.

```
Loser#show crypto cisco algorithms
```

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

- **show crypto cisco key-timeout** – 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 DES (session) key must be generated for encrypted communication to continue. Use this command to change the time that an encrypted communication session lasts before it expires (times out).

```
Loser#show crypto cisco key-timeout
Session keys will be re-negotiated every 30 minutes
```

Use these commands to determine the length of time before the DES keys are renegotiated.

```
StHelen#show crypto conn
Connection Table
PE                UPE                Conn_id New_id Algorithm      Time
0.0.0.1           0.0.0.1           4       0       DES_56_CFB64      Mar 01 1993 03:16:09
                    flags:TIME_KEYS
```

```
StHelen#show crypto key
Session keys will be re-negotiated every 30 minutes
```

```
StHelen#show clock
*03:21:23.031 UTC Mon Mar 1 1993
```

- **show crypto cisco pregen-dh-pairs** – Each encrypted session uses a unique pair of DH numbers. Every time a new session is established, 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 set-up slow, especially for low-end routers.

To accelerate session set-up, 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 is 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 is always 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 is setting up multiple encrypted sessions so frequently that a pregenerated reserve of one or two DH number pairs is depleted too quickly.

```
Loser#show crypto cisco pregen-dh-pairs
Number of pregenerated DH pairs: 10
```

- **show crypto cisco connections active**

The following is sample command output.

```
Loser#show crypto engine connections active
ID   Interface      IP-Address  State  Algorithm      Encrypt  Decrypt
16   Serial1        19.19.19.19 set    DES_56_CFB64   376     884
```

- **show crypto cisco engine connections dropped-packet**

The following is sample command output.

```
Loser#show crypto engine connections dropped-packet
Interface      IP-Address  Drop Count
Serial1        19.19.19.19 39
```

- **show crypto engine configuration** (was **show crypto engine brief** in Cisco IOS Software Release 11.2.)

The following is sample command output.

```
Loser#show crypto engine configuration
slot: 0
engine name: fred
engine type: software
serial number: 02802219
platform: rp crypto engine
crypto lib version: 10.0.0

Encryption Process Info:
input queue top: 465
input queue bot: 465
input queue count: 0
```

- **show crypto key mypubkey dss**

The following is sample command output.

```
Loser#show crypto key mypubkey dss
crypto public-key fred 02802219
79CED212 AF191D29 702A9301 B3E06602 D4FB26B3 316E58C8 05D4930C CE891810
C0064492 5F6684CD 3FC326E5 679BCA46 BB155402 D443F68D 93487F7E 5ABE182E
quit
```

- **show crypto key pubkey-chain dss**

The following is sample command output.

```
Loser#show crypto key pubkey-chain dss
crypto public-key barney 05694352
B407A360 204CBFA3 F9A0C0B0 15D6185D 91FD7D3A 3232EBA2 F2D31D21 53AE24ED
732EA43D 484DEB22 6E91515C 234B4019 38E51D64 04CB9F59 EE357477 91810341
quit
```

- **show crypto map interface serial 1**

The following is sample command output.

```
Loser#show crypto map interface serial 1
Crypto Map "oldstyle" 10 cisco
Connection Id = 16 (8 established, 0 failed)
Peer = barney
PE = 40.40.40.0
UPE = 30.30.30.0
Extended IP access list 133
access-list 133 permit ip
source: addr = 40.40.40.0/0.0.0.255
dest: addr = 30.30.30.0/0.0.0.255
```

Note the time disparity when you use the **ping** command.

```
wan-5200b#ping 30.30.30.30

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.30, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/54/56 ms
wan-5200b#
-----
wan-5200b#ping 30.30.30.31
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 30.30.30.31, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/53/56 ms
-----
```

```
wan-5200b#ping 19.19.19.20
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 19.19.19.20, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
-----
```

- **show crypto map interface serial 1**

The following is sample command output.

```
Loser#show crypto map
Crypto Map "oldstyle" 10 cisco
      Connection Id = 16          (8 established,      0 failed)
      Peer = barney
      PE = 40.40.40.0
      UPE = 30.30.30.0
      Extended IP access list 133
        access-list 133 permit ip
          source: addr = 40.40.40.0/0.0.0.255
          dest:   addr = 30.30.30.0/0.0.0.255
```

- **debug crypto engine**

The following is sample command output.

```
Loser#debug crypto engine
Mar 17 11:49:07.902: Crypto engine 0: generate alg param

Mar 17 11:49:07.906: CRYPTO_ENGINE: Dh phase 1 status: 0
Mar 17 11:49:07.910: Crypto engine 0: sign message using crypto engine
Mar 17 11:49:09.894: CRYPTO_ENGINE: packets dropped: State = 0
Mar 17 11:49:11.758: Crypto engine 0: generate alg param

Mar 17 11:49:12.246: CRYPTO_ENGINE: packets dropped: State = 0
Mar 17 11:49:13.342: CRYPTO_ENGINE 0: get syndrome for conn id 25
Mar 17 11:49:13.346: Crypto engine 0: verify signature
Mar 17 11:49:14.054: CRYPTO_ENGINE: packets dropped: State = 0
Mar 17 11:49:14.402: Crypto engine 0: sign message using crypto engine
Mar 17 11:49:14.934: Crypto engine 0: create session for conn id 25
Mar 17 11:49:14.942: CRYPTO_ENGINE 0: clear dh number for conn id 25
Mar 17 11:49:24.946: Crypto engine 0: generate alg param
```

- **debug crypto sessgmt**

The following is sample command output.

```
StHelen#debug crypto sessgmt

Mar 17 11:49:08.918: IP: s=40.40.40.40 (Serial1), d=30.30.30.30, len 328,
      Found an ICMP connection message.

Mar 17 11:49:08.922: CRYPTO: Dequeued a message: CIM
Mar 17 11:49:08.926: CRYPTO-SDU: Key Timeout, Re-exchange Crypto Keys
Mar 17 11:49:09.978: CRYPTO: Verify done. Status=OK
Mar 17 11:49:09.994: CRYPTO: DH gen phase 1 status for conn_id 22 slot 0:OK
Mar 17 11:49:11.594: CRYPTO: DH gen phase 2 status for conn_id 22 slot 0:OK
Mar 17 11:49:11.598: CRYPTO: Syndrome gen status for conn_id 22 slot 0:OK
Mar 17 11:49:12.134: CRYPTO: Sign done. Status=OK
Mar 17 11:49:12.142: CRYPTO: ICMP message sent: s=19.19.19.20, d=19.19.19.19
```

```

Mar 17 11:49:12.146: CRYPTO-SDU: act_on_nnc_req: NNC Echo Reply sent
Mar 17 11:49:12.154: CRYPTO: Create encryption key for conn_id 22 slot 0:OK
Mar 17 11:49:15.366: CRYPTO: Dequeued a message: CCM
Mar 17 11:49:15.370: CRYPTO: Syndrome gen status for conn_id 22 slot 0:OK
Mar 17 11:49:16.430: CRYPTO: Verify done. Status=OK
Mar 17 11:49:16.434: CRYPTO: Replacing -23 in crypto maps with 22 (slot 0)
Mar 17 11:49:26.438: CRYPTO: Need to pregenerate 1 pairs for slot 0.
Mar 17 11:49:26.438: CRYPTO: Pregenerating DH for conn_id 32 slot 0
Mar 17 11:49:28.050: CRYPTO: DH phase 1 status for conn_id 32 slot 0:OK
                        ~~ <----- This is good -----> ~~

```

If the wrong peer set on the Crypto Map, you receive this error message.

```

Mar  2 12:19:12.639: CRYPTO-SDU:Far end authentication error:
                    Connection message verify failed

```

If the crypto algorithms do not match, you receive this error message.

```

Mar  2 12:26:51.091: CRYPTO-SDU: Connection
                    failed due to incompatible policy

```

If the DSS key is missing or invalid, you receive this error message.

```

Mar 16 13:33:15.703: CRYPTO-SDU:Far end authentication error:
                    Connection message verify failed

```

- **debug crypto key**

The following is sample command output.

```

StHelen#debug crypto key
Mar 16 12:16:45.795: CRYPTO-KE: Sent 4 bytes.
Mar 16 12:16:45.795: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:45.799: CRYPTO-KE: Sent 6 bytes.
Mar 16 12:16:45.799: CRYPTO-KE: Sent 2 bytes.
Mar 16 12:16:45.803: CRYPTO-KE: Sent 64 bytes.

Mar 16 12:16:56.083: CRYPTO-KE: Received 4 bytes.
Mar 16 12:16:56.087: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:56.087: CRYPTO-KE: Received 4 bytes.
Mar 16 12:16:56.091: CRYPTO-KE: Received 2 bytes.
Mar 16 12:16:56.091: CRYPTO-KE: Received 52 bytes.
Mar 16 12:16:56.095: CRYPTO-KE: Received 12 bytes.

```

- **clear crypto connection**

The following is sample command output.

```

wan-2511#show crypto engine connections act
ID      Interface      IP-Address  State  Algorithm      Encrypt  Decrypt
  9      Serial0          20.20.20.21 set    DES_56_CFB64   29       28

wan-2511#clear crypto connection 9
wan-2511#
*Mar  5 04:58:20.690: CRYPTO: Replacing 9 in crypto maps with 0 (slot 0)
*Mar  5 04:58:20.694: Crypto engine 0: delete connection 9
*Mar  5 04:58:20.694: CRYPTO: Crypto Engine clear conn_id 9 slot 0: OK
wan-2511#
wan-2511#show crypto engine connections act
ID      Interface      IP-Address  State  Algorithm      Encrypt  Decrypt

wan-2511#

```

- **crypto zeroize**

The following is sample command output.

```
wan-2511#show crypto mypubkey
crypto public-key wan2511 01496536
 11F43C02 70C0ADB7 5DD50600 A0219E04 C867A5AF C40A4FE5 CE99CCAB A8ECA840
 EB95FBEE D727ED5B F0A6F042 BDB5529B DBB0698D DB0B2756 F6CABE8F 05E4B27F
quit

wan-2511#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
wan-2511(config)#crypto zeroize
Warning! Zeroize will remove your DSS signature keys.
Do you want to continue? [yes/no]: yes
% Keys to be removed are named wan2511.
Do you really want to remove these keys? [yes/no]: yes
% Zeroize done.

wan-2511(config)#^Z
wan-2511#
wan-2511#show crypto mypubkey
wan-2511#
```

- **no crypto public-key**

The following is sample command output.

```
wan-2511#show crypto pubkey
crypto public-key wan2516 01698232
 B1C127B0 78D79CAA 67ECAD80 03D354B1 9012C80E 0C1266BE 25AEDE60 37A192A2
 B066D299 77174D48 7FBAB5FC 2B60893A 37E5CB7B 62F6D902 9495733B 98046962
quit

wan-2511#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
wan-2511(config)#crypto public-key ?
WORD Peer name

wan-2511(config)#
wan-2511(config)#no crypto public-key wan2516 01698232
wan-2511(config)#^Z
wan-2511#
wan-2511#show crypto pubkey
wan-2511#
```

## Troubleshooting Cisco 7200 with ESA

Cisco also provides a hardware assist option to do encryption on the Cisco 7200 series routers, which is called the ESA. The ESA is in the form of a port adapter for the VIP2-40 card or a standalone port adapter for the Cisco 7200. This arrangement allows the use of either a hardware adapter or the VIP2 software engine to encrypt and decrypt data that comes into or leaves through the interfaces on the Cisco 7500 VIP2 card. The Cisco 7200 allows hardware assist to encrypt traffic for any interfaces on the Cisco 7200 chassis. Using an encryption assist saves precious CPU cycles that can be used for other purposes, such as routing or any of the other Cisco IOS functions.

On a Cisco 7200, the standalone port adapter is configured exactly the same as the Cisco IOS software crypto engine, but has a few extra commands that are only used for hardware and for deciding which engine (software or hardware) will do the encryption.

First, prepare the router for hardware encryption:

```
wan-7206a(config)#
%OIR-6-REMCARD: Card removed from slot 3, interfaces disabled
```

\*Mar 2 08:17:16.739: ...switching to SW crypto engine

wan-7206a#**show crypto card 3**

Crypto card in slot: 3

Tampered: No  
Xtracted: Yes  
Password set: Yes  
DSS Key set: Yes  
FW version 0x5049702  
wan-7206a#

wan-7206a(config)#

wan-7206a(config)#**crypto zeroize 3**

Warning! Zeroize will remove your DSS signature keys.

Do you want to continue? [yes/no]: **yes**

% Keys to be removed are named hard.

Do you really want to remove these keys? [yes/no]: **yes**

[OK]

Enable or disable hardware encryption as shown below:

wan-7206a(config)#**crypto esa shutdown 3**  
...switching to SW crypto engine

wan-7206a(config)#**crypto esa enable 3**  
There are no keys on the ESA in slot 3- ESA not enabled.

Next, generate keys for the ESA before you enable it.

wan-7206a(config)#**crypto gen-signature-keys hard**  
% 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:

Re-enter password:

Generating DSS keys ....

[OK]

wan-7206a(config)#

wan-7206a#**show crypto mypubkey**

crypto public-key hard 00000052

EE691A1F BD013874 5BA26DC4 91F17595 C8C06F4E F7F736F1 AD0CACEC 74AB8905

DF426171 29257F8E B26D49B3 A8E11FB0 A3501B13 D3F19623 DCCE7322 3D97B804

quit

wan-7206a#

wan-7206a(config)#**crypto esa enable 3**

...switching to HW crypto engine

wan-7206a#**show crypto engine brie**

crypto engine name: hard

crypto engine type: ESA

serial number: 00000052

crypto engine state: installed

crypto firmware version: 5049702

crypto engine in slot: 3

wan-7206a#

## Troubleshooting VIP2 With ESA

The ESA hardware port adapter on the VIP2 card is used to encrypt and decrypt data that comes into or leaves through the interfaces on the VIP2 card. As with the Cisco 7200, using an encryption assist saves precious CPU cycles. In this case, the **crypto esa enable** command does not exist because the ESA port adapter does the encryption for the ports on the VIP2 card if the ESA is plugged in. The **crypto clear-latch** needs to be applied to that slot if the ESA port adapter was just installed for the first time, or removed then reinstalled.

```
Router#show crypto card 11
```

```
Crypto card in slot: 11
```

```
Tampered:          No
Xtracted:          Yes
Password set:      Yes
DSS Key set:       Yes
FW version         0x5049702
Router#
```

Because the ESA crypto module was extracted, you will get the following error message until you do a **crypto clear-latch** command on that slot, as shown below.

```
----
```

```
*Jan 24 02:57:09.583: CRYPTO: Sign done. Status= Extraction latch set. Request not allowed
```

```
-----
```

```
Router(config)#crypto clear-latch ?
<0-15>  Chassis slot number
```

```
Router(config)#crypto clear-latch 11
% Enter the crypto card password.
Password:
Router(config)#^Z
```

If you forget a previously assigned password, 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 are prompted to create a new password. An example is shown below.

```
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show crypto card 11
```

```
Crypto card in slot: 11
```

```
Tampered:          No
Xtracted:          No
Password set:      Yes
DSS Key set:       Yes
FW version         0x5049702
Router#
```

```
-----
```

```
Router#show crypto engine brief
crypto engine name:  TERT
crypto engine type:  software
serial number:       0459FC8C
crypto engine state: dss key generated
crypto lib version:  5.0.0
crypto engine in slot: 6
```

```
crypto engine name:  WAAA
crypto engine type:  ESA
serial number:       00000078
crypto engine state: dss key generated
crypto firmware version: 5049702
crypto engine in slot: 11
```

Router#

-----

Router(config)#**crypto zeroize**

Warning! Zeroize will remove your DSS signature keys.

Do you want to continue? [yes/no]: **yes**

% Keys to be removed are named TERT.

Do you really want to remove these keys? [yes/no]: **yes**

% Zeroize done.

Router(config)#crypto zeroize 11

Warning! Zeroize will remove your DSS signature keys.

Do you want to continue? [yes/no]: **yes**

% Keys to be removed are named WAAA.

Do you really want to remove these keys? [yes/no]: **yes**

[OK]

Router(config)#^Z

Router#**show crypto engine brief**

```
crypto engine name:  unknown
crypto engine type:  software
serial number:       0459FC8C
crypto engine state: installed
crypto lib version:  5.0.0
crypto engine in slot: 6
```

```
crypto engine name:  unknown
crypto engine type:  ESA
serial number:       00000078
crypto engine state: installed
crypto firmware version: 5049702
crypto engine in slot: 11
```

Router#

-----

Router(config)#**crypto gen-signature-keys VIPESA 11**

% 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:

Re-enter password:

Generating DSS keys ....

[OK]

Router(config)#

\*Jan 24 01:39:52.923: Crypto engine 11: create key pairs.

^Z

Router#

-----

Router#**show crypto engine brief**

```
crypto engine name:  unknown
crypto engine type:  software
serial number:       0459FC8C
crypto engine state: installed
crypto lib version:  5.0.0
crypto engine in slot: 6
```

```
crypto engine name:  VIPESA
crypto engine type:  ESA
```

```
serial number:          00000078
crypto engine state:    dss key generated
crypto firmware version: 5049702
crypto engine in slot: 11
```

```
Router#
```

```
-----
Router#show crypto engine connections active 11
```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
2	Serial11/0/0	20.20.20.21	set	DES_56_CFB64	9996	9996

```
Router#
```

```
Router#clear crypto connection 2 11
```

```
Router#
```

```
*Jan 24 01:41:04.611: CRYPTO: Replacing 2 in crypto maps with 0 (slot 11)
```

```
*Jan 24 01:41:04.611: Crypto engine 11: delete connection 2
```

```
*Jan 24 01:41:04.611: CRYPTO: Crypto Engine clear conn_id 2 slot 11: OK
```

```
Router#show crypto engine connections active 11
```

```
No connections.
```

```
Router#
```

```
*Jan 24 01:41:29.355: CRYPTO ENGINE: Number of connection entries
```

```
received from VIP 0
```

```
-----
```

```
Router#show crypto mypub
```

```
% Key for slot 11:
```

```
crypto public-key VIPESA 00000078
```

```
CF33BA60 56FCEE01 2D4E32A2 5D7ADE70 6AF361EE 2964F3ED A7CE08BD A87BF7FE
```

```
90A39F1C DF96143A 9B7B9C78 5F59445C 27860F1E 4CD92B6C FBC4CBCC 32D64508
```

```
quit
```

```
Router#show crypto pub
```

```
crypto public-key wan2516 01698232
```

```
C5DE8C46 8A69932C 70C92A2C 729449B3 FD10AC4D 1773A997 7F6BA37D 61997AC3
```

```
DBEDBEA7 51BF3ADD 2BB35CB5 B9126B4D 13ACF93E 0DF0CD22 CFAAC1A8 9CE82985
```

```
quit
```

```
Router#
```

```
-----
interface Serial11/0/0
```

```
ip address 20.20.20.21 255.255.255.0
```

```
encapsulation ppp
```

```
ip route-cache distributed
```

```
no fair-queue
```

```
no cdp enable
```

```
crypto map test
```

```
!
```

```
-----
```

```
Router#show crypto eng conn act 11
```

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
3	Serial11/0/0	20.20.20.21	set	DES_56_CFB64	761	760

```
Router#
```

```
*Jan 24 01:50:43.555: CRYPTO ENGINE: Number of connection
```

```
entries received from VIP 1
```

```
Router#
```

---

## Related Information

- **Configuring and Troubleshooting Cisco Network-Layer Encryption: IPsec and ISAKMP – Part 2**

- **Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, 2nd Edition, 1995.**
  - **DES FIPS 46–2 at National Institute of Standards and Technology (NIST)**
  - **DSS FIPS 186 at National Institute of Standards and Technology (NIST)**
  - **RSA Laboratories' Frequently Asked Questions About Today's Cryptography**
  - **IETF Security Standards**
  - **Configuring Internet Key Exchange Security Protocol**
  - **Configuring IPsec Network Security**
  - **IPsec Support Page**
  - **Technical Support – Cisco Systems**
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Jan 14, 2008

Document ID: 17584

---