Configuring Encryption Services

The Encryption Service Adapter (ESA) is a high-performance data encryption module that offloads some of the encryption processing from the Catalyst 4224 main processor and improves performance. The ESA implements data encryption and authentication algorithms on the Catalyst 4224 through a software service called a crypto engine.

The ESA includes a public key math processor and a hardware random number generator. These features support public key cryptography for key generation, exchange, and authentication. The ESA can encrypt and authenticate two full-duplex T1 or two E1 communication links.

Each data line can be channelized with a separate encryption context. The ESA uses Public Key (PK) technology based on the concept of the Protected Entity (PE) and employs IPSec Data Encryption Standard (DES) 56-bit and 3(Triple) DES 168-bit encryption to ensure that secure data and information can be transferred between similarly equipped hosts on your network.

This section details how to configure the ESA and includes the following topics:

Configuring the Encryption Service Adapter

Configuring the ESA requires four steps, as outlined below:

- Step 1: Configure the T1 Channel Group, page 11-2
- Step 2: Configure the Internet Key Exchange Security Protocol, page 11-3
- Step 3: Configure IPSec Network Security, page 11-5
- Step 4: Configure Encryption on the T1 Channel Group Serial Interface, page 11-8

Step 1: Configure the T1 Channel Group

The first step toward configuring the ESA is to establish a T1 connection. You must define the characteristics of a configuration group (such as speed and slot number).

To configure the T1 channel group, follow this procedure:

	Task	Command
Step 1	Specify a controller and enter controller configuration mode.	`Gateway(config)# controller {t1|e1}slot|port`
Step 2	Specify the clock source for a link. **line** specifies that the link uses the recovered clock from the link and is the default setting. Generally, this setting is most reliable. **internal** specifies that the DS1 link uses the internal clock. **loop-timed** specifies that the T1 or E1 interface takes the clock from the Rx (line) and uses it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command.	`Gateway(config-controller)# clock source {line|internal|loop-timed}`
Step 3	Select frame clock.	`Gateway(config-controller)# frame-clock-select{priority} {E1/T1} {slot/port}`

	Task	Command
Step 4	Specify the framing type for the T1 or E1 data line. **sf** specifies Super Frame as the T1 frame type. **esf** specifies Extended Super Frame as the T1 frame type.	`Gateway(config-controller)#` **`framing {sf|esf}`**
Step 5	Specify the line code format. **ami** specifies alternate mark inversion (AMI) as the line-code type. Valid for T1 or E1 controllers; the default for T1 lines. **b8zs** specifies B8ZS as the line-code type. Valid for T1 controller only. **hdb3** specifies high-density bipolar 3 (hdb3) as the line-code type. Valid for E1 controller only; the default for E1 lines.	`Gateway(config-controller)#` **`linecode {ami|b8zs|hdb3}`**
Step 6	Specify the channel group and time slots to be mapped.	`Gateway(config-controller)#` **`channel-group`** *`channel_number`* **`timeslots`** *`range`*
Step 7	Return to global configuration mode.	`Gateway(config-controller)#` **`exit`**

Step 2: Configure the Internet Key Exchange Security Protocol

The second step is to establish an Internet Key Exchange (IKE) Security Protocol for encryption.

The Internet Key Exchange (IKE) protocol is a key management protocol standard that is used in conjunction with the IPSec standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets. IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard. (For more information on IPSec, see the "Step 3: Configure IPSec Network Security" section on page 11-5.)

To configure an IKE Security Protocol, follow this procedure:

	Task	Command
Step 1	Create an IKE policy[1] with a unique priority number and enter Internet Security Association and Key Management Protocol (ISAKMP[2]) policy configuration mode. **Note** You can configure multiple policies on each peer[3]. At least one of these policies must contain exactly the same encryption, authentication, and other parameters as one of the policies on the remote peer.	`Gateway(config)#` **`crypto isakmp policy`** *`priority`*
Step 2	Specify the authentication method to be used in an IKE policy.	`Gateway(config-isakmp)#` **`authentication`** {**`rsa-sig`**|**`rsa-encr`**|**`pre-share`**}
Step 3	Return to global configuration mode.	`Gateway(config-isakmp)#` **`exit`**
Step 4	Configure the authentication key for each peer that shares a key.	`Gateway(config)#` **`crypto isakmp key`** *`keystring`* **`address`** *`peer_address|peer_hostname`*

1. You must create IKE policies at each peer. An IKE policy defines a combination of security parameters to be used during the IKE negotiation. IKE negotiations must be protected, so each IKE negotiation begins by each peer agreeing on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations. After the two peers agree upon a policy, the security parameters of the policy are identified by a security association established at each peer, and these security associations apply to all subsequent IKE traffic during the negotiation.
2. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.
3. In the context of this document, a peer refers to a Catalyst 4224 or other device that participates in IPSec and IKE.

For information on how to create a private or public key and to download a certificate, visit the following website:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt4/scdipsec.htm

Step 3: Configure IPSec Network Security

The third step is to define how the T1 data will be handled. This requires that you use IPSec (IP Security Protocol) security.

IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

To configure IPSec network security, follow this procedure:

	Task	Command
Step 1	Specify the lifetime of a security association[1]. As a general rule, the shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPSec security associations can be set up more quickly. The default lifetimes are 3600 seconds (one hour) and 4608000 kilobytes (10 megabytes per second for one hour).	`Gateway(config)# crypto ipsec security-association lifetime seconds seconds kilobytes`
Step 2	Specify a transform set[2] and enter transform-set configuration mode. To define a transform set, specify one to three "transforms"—each *transform* represents an IPSec security protocol (ESP or AH) plus the algorithm you want to use. When the particular transform set is used during negotiations for IPSec security associations, the entire transform set (the combination of protocols, algorithms and other settings) must match a transform set at the remote peer.	`Gateway(config)# crypto ipsec transform-set transform_set_name transform1 [transform2 [transform3]]`
Step 3	Return to global configuration mode.	`Gateway(cfg-crypto-trans)# exit`

	Task	Command
Step 4	Create a crypto map[3] denoted by *map-name*. Enter crypto map configuration mode, unless you use the dynamic keyword. *seq-num* is the number you assign to the crypto map entry. **ipsec-isakmp** indicates that IKE will be used to establish the IPSec security associations for protecting the traffic specified by this crypto map entry. **dynamic** is an optional argument specifying that this crypto map entry references a preexisting dynamic crypto map. Dynamic crypto maps are policy templates used in processing negotiation requests from a peer IPSec device. If you use this keyword, none of the crypto map configuration commands will be available. *dynamic-map-name* specifies the name of the dynamic crypto map set that should be used as the policy template.	`Gateway(config)# crypto map map_name seq_num ipsec-isakmp [dynamic dynamic_map_name] [discover]`
Step 5	Specify the same remote IPSec peer that you specified in Step 4 in the previous procedure, "Step 2: Configure the Internet Key Exchange Security Protocol" section on page 11-3.	`Gateway(config-crypto map)# set peer hostname|ip_address`
Step 6	For this crypto map entry, specify the same transform set that you specified in Step 2 of this procedure.	`Gateway(config-crypto map)# set transform-set transform_set_name`
Step 7	Specify an extended access list for a crypto map entry. This value should match the access-list-number or name argument of the extended access list.	`Gateway(config-crypto map)# match address [access_list_id | name]`

	Task	Command
Step 8	Return to global configuration mode.	`Gateway(cfg-crypto-trans)# exit`
Step 9	Create an access list.[4] access_list_number denotes an IP list number from 1 through 99. **permit** or **deny** specifies permit or deny condition for this list. IP-address is the IP address to which the router compares the address being tested. wild-mask is the wildcard mask bits for the address in 32-bit, dotted decimal notation.	`Gateway(config)# access-list access_list_number {permit | deny} {type_code wild_mask | address mask}`

1. A security association (SA) describes how two or more entities will utilize security services to communicate securely. For example, an IPSec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPSec connection. Both IPSec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPSec SA is established either by IKE or by manual user configuration.
2. A transform set represents a specific combination of security protocols and algorithms. During the IPSec security association negotiation, the peers search for a transform set that is the same on both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers' IPSec security associations.
3. With IPSec you define what traffic should be protected between two IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in order, and the Catalyst 4224 attempts to match the packet to the access list specified in that entry.
4. Packet filtering helps control packet movement through the network. Such control can help limit network traffic and restrict network use by certain users or devices. To permit or deny packets from crossing specified interfaces, Cisco provides access lists. An access list is a sequential collection of permit and deny conditions that apply to IP addresses.

Step 4: Configure Encryption on the T1 Channel Group Serial Interface

The fourth step is to configure a T1 serial interface with an IP address and a crypto map.

To configure encryption on the T1 channel group, follow this procedure:

	Task	Command
Step 1	Select the serial interface and enter interface configuration mode.	`Gateway (config)# interface serial slot|port:timeslot`
Step 2	Specify an IP address followed by the subnet mask for this interface.	`Gateway (config-if)# ip address address mask`
Step 3	Assign a crypto map to this interface.	`Gateway (config-if)# crypto map map_name`
Step 4	Return to global configuration mode.	`Gateway(config-if)# exit`
Step 5	Return to the enable mode.	`Gateway(config)# exit`
Step 6	Display the current operating configuration, including any changes just made.	`Gateway# show running-config`
Step 7	Display the configuration currently stored in nonvolatile random-access memory (NVRAM).	`Gateway# show startup-config`
Step 8	At the enable prompt, write your changes to NVRAM. **Note** The results of the **show running-config** and **show startup-config** commands differ if you have made changes to the configuration but have not yet written them to NVRAM.	`Gateway# copy running-config startup-config`

For complete information about configuration commands and about configuring LAN and WAN interfaces on your switch, refer to the Cisco IOS configuration guides and command references.

Verifying the Configuration

After configuring the new interface, use the following commands to verify that it is operating correctly:

- Use **show version** to display the router hardware configuration. Check that the list includes the new interface.
- Use **show controllers** to display all network modules and their interfaces.
- Use **show interfaces** [*type slot/port*] to display the details of a specified interface. Verify that the first line of the display shows the correct slot and port number and that the interface and line protocol are in the correct state (up or down).
- Use **show protocols** to display the protocols configured for the entire router and for individual interfaces. If necessary, add or remove protocol routing on the router or its interfaces.
- Use **show running-config** to display the running configuration.
- Use **show startup-config** to display the configuration stored in NVRAM.
- Use **ping** to send an echo request to a specified IP address.

Note Encryption is enabled by default when you install the ESA hardware. If you need to enable encryption, use the **no crypto engine accel** command. This command is useful for debugging problems with the ESA or for testing features available only with software encryption.

Sample Configurations

This section contains the following topics:

Encrypting Traffic Between Two Networks

The sample configurations in this section show you how to encrypt traffic between a private network (10.103.1.x) and a public network (98.98.98.x) using IPSec. The 98.98.98.x network knows the 10.103.1.x network by the private addresses. The 10.103.1.x network knows the 98.98.98.x network by the public addresses.

Configuration File for the Public Gateway

```
gateway-2b# show running config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname gateway-2b
!
ip subnet-zero
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 1
hash md5
authentication pre-share
crypto isakmp key cisco123 address 95.95.95.2
!
crypto ipsec transform-set rtpset esp-des esp-md5-hmac
!
crypto map rtp 1 ipsec-isakmp
set peer 95.95.95.2
set transform-set rtpset
match address 115
!
interface Ethernet0/0
ip address 98.98.98.1 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0/1
ip address 99.99.99.2 255.255.255.0
no ip directed-broadcast
```

```
no ip route-cache
no ip mroute-cache
crypto map rtp
!
interface Ethernet0/2
no ip address
no ip directed-broadcast
shutdown
!
interface Ethernet0/3
no ip address
no ip directed-broadcast
shutdown
!
ip classless
ip route 0.0.0.0 0.0.0.0 99.99.99.1
no ip http server
!
access-list 115 permit ip 98.98.98.0 0.0.0.255 10.103.1.0 0.0.0.255
access-list 115 deny ip 98.98.98.0 0.0.0.255 any
!
line con 0
transport input none
line aux 0
line vty 0 4
login
!
end
```

Configuration File for the Private Gateway

```
gateway-6a# show running config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname gateway-6a
!
enable secret 5 $1$S/yK$RE603ZNv8N71GDYDbdMWd0
enable password ww
!
```

```
ip subnet-zero
!
ip audit notify log
ip audit PO max-events 100
isdn switch-type basic-5ess
isdn voice-call-failure 0
!

crypto isakmp policy 1
hash md5
authentication pre-share
crypto isakmp key cisco123 address 99.99.99.2
!
crypto ipsec transform-set rtpset esp-des esp-md5-hmac
crypto map rtp 1 ipsec-isakmp
set peer 99.99.99.2
set transform-set rtpset
match address 115
!
interface Ethernet0/0
no ip address
no ip directed-broadcast
!
interface Serial0/0
no ip address
no ip directed-broadcast
no ip mroute-cache
shutdown
!
interface Ethernet0/1
no ip address
no ip directed-broadcast
!
interface Serial0/1
no ip address
no ip directed-broadcast
shutdown
!
interface BRI1/0
no ip address
no ip directed-broadcast
shutdown
isdn switch-type basic-5ess
!
interface Ethernet1/0
no ip address
no ip directed-broadcast
shutdown
```

```
!
interface Serial1/0
no ip address
no ip directed-broadcast
shutdown
!
interface TokenRing1/0
no ip address
no ip directed-broadcast
shutdown
ring-speed 16
!
interface Ethernet3/0
ip address 95.95.95.2 255.255.255.0
no ip directed-broadcast
ip nat outside
no ip route-cache
no ip mroute-cache
crypto map rtp
!
interface Ethernet3/1
no ip address
no ip directed-broadcast
shutdown
!
interface Ethernet3/2
ip address 10.103.1.75 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface Ethernet3/3
no ip address
no ip directed-broadcast
shutdown
!
ip nat pool FE30 95.95.95.10 95.95.95.10 netmask 255.255.255.0
ip nat inside source route-map nonat pool FE30 overload
ip classless
ip route 0.0.0.0 0.0.0.0 95.95.95.1
ip route 171.68.120.0 255.255.255.0 10.103.1.1
no ip http server
!
access-list 110 deny ip 10.103.1.0 0.0.0.255 98.98.98.0 0.0.0.255
access-list 110 permit ip 10.103.1.0 0.0.0.255 any
access-list 115 permit ip 10.103.1.0 0.0.0.255 98.98.98.0 0.0.0.255
access-list 115 deny ip 10.103.1.0 0.0.0.255 any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
```

```
route-map nonat permit 10
match ip address 110
!
tftp-server flash:cgateway-io3s56i-mz.120-7.T
!
line con 0
transport input none
line 65 72
line aux 0
line vty 0 4
password WW
login
!
end
```

Exchanging Encrypted Data Through an IPSec Tunnel

This section contains sample configuration files for two peer Catalyst 4224s set up to exchange encrypted data through a secure IPSec tunnel over a channelized T1 interface channel group, serial 1/0:0.

Configuration File for Peer 1

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Rose
!
logging buffered 100000 debugging
enable password lab
!
ip subnet-zero
no ip domain-lookup
!
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key pre-shared address 6.6.6.2
!
crypto ipsec security-association lifetime seconds 86400
!
crypto ipsec transform-set transform-1 esp-des
```

```
!
crypto map cmap 1 ipsec-isakmp
 set peer 6.6.6.2
 set transform-set transform-1
 match address 101
!
controller T1 1/0
 framing esf
 linecode b8zs
 channel-group 0 timeslots 1-23 speed 64
 channel-group 1 timeslots 24 speed 64
!
controller T1 1/1
 channel-group 0 timeslots 1-23 speed 64
 channel-group 1 timeslots 24 speed 64
!
process-max-time 200
!
interface FastEthernet0/0
 ip address 111.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 speed 10
!

interface Serial0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface FastEthernet0/1
 ip address 4.4.4.1 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 speed 10
!
interface Serial1/0:0
 bandwidth 1472
 ip address 6.6.6.1 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 load-interval 30
 no fair-queue
 crypto map cmap
```

```
!
interface Serial1/0:1
 no ip address
 no ip directed-broadcast
 fair-queue 64 256 0
!
interface Serial1/1:0
 no ip address
 no ip directed-broadcast
!
interface Serial1/1:1
 no ip address
 no ip directed-broadcast
 fair-queue 64 256 0
!
router rip
 network 4.0.0.0
 network 6.0.0.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 111.0.0.1
no ip http server
!
access-list 101 deny   udp any eq rip any
access-list 101 deny   udp any any eq rip
access-list 101 permit ip 6.6.6.0 0.0.0.255 6.6.6.0 0.0.0.255
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password lab
 login
!
end
```

Configuration File for Peer 2

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Peony
!
```

```
logging buffered 100000 debugging
enable password lab
!
ip subnet-zero
no ip domain-lookup
!
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key pre-shared address 6.6.6.1
!
crypto ipsec security-association lifetime seconds 86400
!
crypto ipsec transform-set transform-1 esp-des
!
crypto map cmap 1 ipsec-isakmp
 set peer 6.6.6.1
 set transform-set transform-1
 match address 101
!
controller T1 1/0
 framing esf
 linecode b8zs
 channel-group 0 timeslots 1-23 speed 64
 channel-group 1 timeslots 24 speed 64
!
controller T1 1/1
 channel-group 0 timeslots 1-23 speed 64
 channel-group 1 timeslots 24 speed 64
!
process-max-time 200
!
interface FastEthernet0/0
 ip address 172.0.0.13 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 load-interval 30
 no keepalive
 speed 10
!
interface FastEthernet0/1
 ip address 3.3.3.2 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 speed 10
!
interface Serial1/0:0
```

```
 bandwidth 1472
 ip address 6.6.6.2 255.0.0.0
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 load-interval 30
 no fair-queue
 crypto map cmap
!

interface Serial1/0:1
 no ip address
 no ip directed-broadcast
 fair-queue 64 256 0
!
interface Serial1/1:0
 no ip address
 no ip directed-broadcast
!
interface Serial1/1:1
 no ip address
 no ip directed-broadcast
 fair-queue 64 256 0
!
router rip
 network 3.0.0.0
 network 6.0.0.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 111.0.0.1
no ip http server
!
access-list 101 deny   udp any eq rip any
access-list 101 deny   udp any any eq rip
access-list 101 permit ip 6.6.6.0 0.0.0.255 6.6.6.0 0.0.0.255
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!!
end
```