Global Information Assurance Certification Paper

Configuration of Tunnel Mode IPSec VPN Using Cisco Routers

Fouzan M. Pal

SANS GSEC Practical Version 1.4b Option 1
November 3, 2003

1 Summary

For businesses today, the need to share data between different branch offices is greater than ever. The internet provides an economical, pre-existing infrastructure for accomplishing this but is plagued by security threats. IPSec provides a secure method for organizations to share data over the internet by implementing security at the network layer using the commonly implemented Internet Protocol. Cisco, the largest manufacturer of IP routers, offers IPSec implementation in its routers.

The purpose of this paper is to present and explain the steps necessary to configure tunnel mode IPSec between two Cisco routers. In order to provide a thorough understanding of the configuration steps, an overview of the relevant features of IPSec is presented first. As key management forms an important part of the configuration process, a two-step approach is taken to help the reader understand the implementation of IPSec on Cisco routers. First, an example of IPSec with manual keying is presented; then, a more involved example of IPSec using IKE is provided. Relevant portions of the final configurations for each peer are also presented at the end of each example.

2 Introduction

With the growth and versatility of the internet, security has become a primary focus of companies large and small. Organizations often need to transfer proprietary data between geographically separated branch offices. Whereas leased lines provide a secure way of doing this, they are not economically feasible for small or mid-sized businesses. A secure way to communicate over the pre-existing infrastructure of the internet is the only viable solution for such cost conscious businesses.

Security can be built in different layers of the OSI model. Link-layer security for example offers great protection but is only feasible for a private network not separated by large geographic distances. On the world wide web of the internet, security must be implemented on higher layers. One solution is presented by the various security offerings at the application layer. However, these technologies are cumbersome and inefficient since each application must implement its own application-specific security architecture. The solution lies in offering security on the layer that is common to the vast infrastructure of the internet – the network layer. Since the expansive architecture of the internet primarily shares the same protocol, the Internet Protocol, to interconnect nodes and hosts at the network

layer, it is most desirable that a security solution be implemented uniformly at this layer. IPSec offers exactly such a solution.

3 Assumptions

IPSec is a comprehensive and flexible suite of protocols which spans a number of documents. The detailed set of IPSec requirements is presented in a series of Requests For Comments (RFCs 2401-2412) published by the Internet Engineering Task Force (IETF). It is not the purpose of this paper to present the nuts and bolts of the IPSec architecture. However, every attempt is made to explain the relevant features of IPSec as they relate to configuring tunnel mode IPSec on Cisco routers. Similarly, an explanation of the basics of configuring Cisco devices is not a feasible sub-topic for this paper. Therefore, it is assumed that the reader has a fundamental knowledge of configuring Cisco routers.

4 Conventions

The commands used in configuring Cisco routers are presented in `Courier` font type. The following conventions are used when presenting Cisco IOS command syntax:

- the actual command text is in **`bold`** face;
- `{ }` designate choice;
- `|` designate OR when listing the choices;
- `[ ]` designate optional parameters;
- entries that should be replaced with appropriate values are *`italicized.`*

This convention is similar to that used in Cisco's documentation.[1]

5 Relevant Overview of IPSec

IPSec stands for Internet Protocol Security. It is a suite of protocols developed by the IETF to allow for the implementation of security features in data traversing over the IP protocol. It accomplishes this using three main features as part of the suite of protocols: 1) a key exchange feature known as Internet Key Exchange (IKE), 2) an authentication-only feature known as Authentication Header (AH), and 3) a combined authentication and encryption feature known as Encapsulating Security Payload (ESP).

IKE, defined by RFC 2409, is a hybrid protocol which defines the method for exchanging keys in a secure fashion to negotiate and use security associations (for use in IPSec) in a secure manner. Even though the IPSec protocol standard allows for the use of other key exchange protocols, it describes IKE as the default protocol for automated key management.[2] For the purposes of our discussion, IKE can be viewed as being synonymous with the Internet Security Association and Key Management Protocol (ISAKMP) since the Cisco

[1] Cisco, "About the Cisco IOS Software Documentation", p. xii.
[2] Kent, "Security Architecture for the Internet Protocol", p. 27.

commands used to configure key exchange settings use the keyword `isakmp`. However, it is important to understand that IKE is a combination of the relevant features of ISAKMP, Oakley Key Determination Protocol (Oakley), and Secure Key Exchange Mechanism (SKEME).[3]

AH, defined by RFC 2402, provides support for data integrity (i.e. data has not been modified in transit), data authentication (i.e. data is indeed from the claimed source), and protection against replay attacks (i.e. attacker can not take an authenticated packet and resend it to the destination, tricking the destination into thinking that the packet is legitimate).

ESP, defined by RFC 2406, provides support for data confidentiality (i.e. data can not be read by an unintended recipient) in addition to supporting the services provided by AH. Due to the high security risk associated with the Internet nowadays, ESP, which incorporates greater security features, is a more likely choice for protecting an organization's data as it traverses from one point to another over the internet.

It is relevant to understand that IPSec offers two modes of operation when employing AH or ESP to protect IP data: 1) Transport Mode and 2) Tunnel Mode. Transport mode is typically used for end-to-end communications between two hosts where the hosts are responsible for implementing IPSec for any communication that is to be secured. In this mode, the original IP header is maintained and the AH and/or ESP header is added between the original IP header and the payload; the AH and/or ESP trailer is appended to the end of the original IP packet. Tunnel mode is used when communication is to take place through gateways (e.g. Cisco routers) which are to create secure "tunnels" through which the insecure traffic from the hosts can travel in a secure fashion. In this mode, the whole of the original IP packet (including the original header) is encapsulated and a new IP header followed by the AH and/or ESP header is added to the front of the original IP packet; the AH and/or ESP trailer is appended to the end of the original IP packet.[4] Since implementing IPSec on separate hosts can be cumbersome as the number of hosts grows, tunnel mode operation is often the preferred choice. This paper discusses the configuration of tunnel mode IPSec configuration using Cisco routers.

The other aspect which is relevant to the configuration of IPSec on Cisco routers relates to Security Association (SA). SAs form an integral part of IPSec. An SA can be viewed as the set of parameters used to define the security requirements for communication in a particular direction (incoming or outgoing). A particular SA can make use of AH, or ESP, but not both, and is defined by a Security Parameter Index (SPI), an IP destination address, and an AH or ESP identifier.[5]

---

[3] Harkins, p. 1-2.
[4] For a clear and concise coverage of tunnel mode vs. transport mode, see Stallings, p. 21-23.
[5] Kent, "Security Architecture for the Internet Protocol", p. 8.

When going through the configuration examples presented in this paper, it is recommended that the reader keep mental notes to trace the Cisco IOS terminology for establishing various features of IPSec with the generic IPSec terminology as described in the RFCs. Often times, this is not a one-to-one relationship due to the fact that the organization of Cisco IOS commands is dependent on a number of other services in addition to just IPSec.

6 Test Configuration

To illustrate the IPSec manual key and IKE examples, we will be using the test configuration shown in Figure 1.

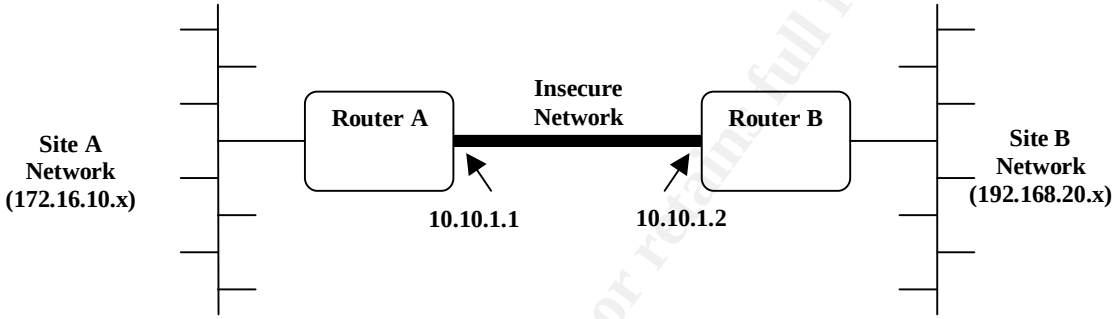

Figure 1: Test network diagram

Note: The example configurations discussed in this paper assume Cisco 3600 series router with IOS mainline version 12.1.[6]

7 Steps for Configuring IPSec Using Manual Key Management

The general procedure that we will use for configuring IPSec on Cisco routers using manual key management is as follows:

- Create access list
- Configure transform set
- Configure crypto map with manual key management
- Apply crypto map to interface

7.1 Create Access Lists

Access lists are commonly used on Cisco routers to filter incoming or outgoing traffic based on various criteria. As a packet enters or leaves an interface of a router, it is matched against the rules specified in the access list(s) for the given interface. Based on this matching process, the packet is either permitted or denied entry into or exit out of the interface.

[6] "IPSec was first available starting with IOS version 11.3T. It was introduced into the mainline IOS, starting with version 12.0." (Quiggle, p. 298)

It is important to understand that when access lists are used in the context of IPSec, they function to determine whether IPSec processing is applied to an incoming/outgoing packet or not -- not to block or permit the packet.

The command for creating a numbered access list is as follows:

```
Router(config)# access-list access-list-number {deny | permit} protocol source-address source-wildcard destination-address destination-wildcard [eq port-number] [log]
```

In our example, we create the access list on each router as follows:

```
RouterA(config)# access-list 110 permit ip 192.168.20.0 0.0.0.255 172.16.10.0 0.0.0.255

RouterB(config)# access-list 120 permit ip 172.16.10.0 0.0.0.255 192.168.20.0 0.0.0.255
```

When referenced by the IPSec commands (as shown in section 7.3.2), these lists will serve the purpose of specifying the packets which should have IPSec processing applied to them.

7.2 Configure Transform Set

Transform sets define the IPSec security policies that will be applied to the desired traffic as it exits or enters the interface. IPSec standard specifies the use of security associations in determining what security policies are applied to the desired traffic; transform sets can be thought of as defining these security associations for use with crypto maps.

When configuring transform sets, we must specify the protocol of choice (AH authentication, ESP authentication, or ESP encryption) and the mode of operation (tunnel or transport).

7.2.1 Define Transform Set Protocol

The command for defining the protocol used in the transform set is as follows:

```
Router(config)# crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]
```

Note that you can choose up to three transform protocols (i.e. one AH, one ESP authentication, and one ESP encryption).

The choices allowed for specifying the security protocols are shown in Table 1:

Transform Type	Transform	Description

AH Transform *(Pick up to one.)*	**ah-md5-hmac**	AH with the MD5 (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
ESP Encryption Transform *(Pick up to one.)*	**esp-des**	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm
ESP Authentication Transform *(Pick up to one.)*	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform *(Pick up to one.)*	**comp-lzs**	IP compression with the LZS algorithm.

Table 1: Allowed Transform Protocol Combinations [7]

A detailed coverage of the transform algorithms listed above is beyond the scope of this paper. Suffice it to say, it is helpful to view them in two broad categories:

1) Authentication algorithms which use hashing techniques to authenticate the information contained in the packet. Examples of these algorithms are Message Digest 5 (MD5) and Secure Hash Algorithm (SHA), both of which are variants of Hashed Message Authentication Codes (HMAC).
2) Encryption algorithms which use keys to encrypt the data contained in the packet so that it can not be read by anyone other than the intended recipient. Examples of these algorithms are Data Encryption Standard (DES) and Triple DES (3DES).

In our example, we choose to encrypt our data and define the transform set as follows:

```
RouterA(config)# crypto ipsec transform-set
RouterATransform esp-des

RouterB(config)# crypto ipsec transform-set
RouterBTransform esp-des
```

Note: We have selected the DES encryption algorithm for illustrative purposes only due to its minimum key length requirements when applying manual key management (see section 7.3.5). Back in 1998, DES was proven to be an insecure encryption algorithm and is not recommended for use in real life.[8]

[7] Cisco, "Configuring IPSec Network Security", p. SC-348.
[8] Electronic Frontier Foundation.

As shown above, the same transform set protocols are specified at the two ends of the link.

7.2.2 Specify Transform Set Mode

The command for specifying the mode in which IPSec should operate is as follows:

```
Router(cfg-crypto-tran)# mode [tunnel | transport]
```

In our example, we choose the tunnel mode and specify the transform set mode as follows:

```
RouterA(cfg-crypto-tran)# mode tunnel
RouterB(cfg-crypto-tran)# mode tunnel
```

Note that since we want our Cisco routers to act as gateways which will apply IPSec to intended traffic going through them, we choose to operate in the tunnel mode.

7.3 Configure Crypto Map with Manual Key Management

Crypto maps can be viewed as the glue that ties the various pieces of IPSec configurations in Cisco routers together to create a comprehensive security relationship. As such, they form an important part of configuring IPSec on Cisco routers.

It is important to understand the composition of crypto maps in relation to security associations. A particular crypto map can contain one or more crypto map entries depending on the complexity of the security association between the peers. The crypto map entries are distinguished by the sequence numbers assigned to them. Any traffic that is to be passed through the IPSec-protected interfaces is evaluated against the crypto map entries (contained in the crypto map applied to that interface) starting from the lowest sequence number (highest priority) to the highest sequence number (lowest priority). These crypto map entries specify the security association which must be used in exchanging traffic between the two peers. In order for communication to take place between the two peers, at least one crypto map entry on the first peer's interface must be compatible with at least one crypto map entry on the second peer's interface. (Note: Multiple remote peers can be defined using crypto maps but that is beyond the scope of this paper.)

Crypto maps are where the differentiation between the use of manual key management and IKE is made. The steps we have described prior to this point are common regardless of the type of key management used. As described above, in the interest of taking a progressive approach to understanding IPSec and its configuration on Cisco routers, we will first walk through the complete example of configuring manual key management and then delve into the more complicated example of configuring IKE for key management.

Due to the comprehensive functionality of crypto maps, they can be a bit confusing to understand and it helps to break out the configuration of crypto maps into the following steps:

- Create a crypto map with manual key management
- Specify the data traffic to secure
- Specify the peer node
- Specify the transform set to use
- Define security keys

7.3.1 Create a Crypto Map with Manual Key Management

A crypto map is created by specifying the name of the map, the sequence number of the map, and the type of key management that will be used between the two peers. The command for creating a crypto map with manual key management is as follows:

```
Router(config)# crypto map map-name seq-num ipsec-manual
```

In our example, we create the crypto map with manual key management as follows:

```
RouterA(config)# crypto map RouterACryptoMap 10 ipsec-
manual

RouterB(config)# crypto map RouterBCryptoMap 20 ipsec-
manual
```

Note that this command puts us in the crypto map mode so that subsequent specifications can be made in relation to the crypto map that was created.

7.3.2 Specify the Data Traffic to Secure

In order for IP security to be applied, we must specify the traffic that is to be secured. Earlier, we created an access list to specify the traffic of interest. Now we must apply that access list to the crypto map and create a definition of what traffic will have security parameters applied to it. The command for applying the access list to the crypto map is as follows:

```
Router(config-crypto-map)# match address access-list-number
```

In our example, we apply the access list to the crypto map as follows:

```
RouterA(config-crypto-map)# match address 110

RouterB(config-crypto-map)# match address 120
```

7.3.3 Specify the Peer Node

Just as the crypto map needs to know what traffic to apply the security parameters to, it also must know the IP address of the peer node. The command for associating the IP address of the peer node with the crypto map is as follows:

```
Router(config-crypto-map)# set peer {hostname | ip-address}
```

In our example, we associate the IP address of the peer node with the crypto map as follows:

```
RouterA(config-crypto-map)# set peer 10.10.1.2

RouterB(config-crypto-map)# set peer 10.10.1.1
```

Note that the IP address specified in this command must be the address of the peer's interface on which IPSec is applied.

7.3.4 Specify the Transform Set to Use

The next step entails associating the security rules to the crypto map. Earlier, we defined the security rules by creating a transform set. Now we must apply that transform set to the crypto map. The command for associating the transform set with the crypto map is as follows:

```
Router(config-crypto-map)# set transform-set transform-set-
name
```

In our example, we associate the transform-set to the crypto map as follows:

```
RouterA(config-crypto-map)# set transform-set
RouterATransform

RouterB(config-crypto-map)# set transform-set
RouterBTransform
```

7.3.5 Define Security Keys

The final step in the configuration of crypto maps consists of defining the security keys. In the case of manual key management, we must specify the keys manually for the authentication as well as encryption of the data depending on the type of security desired. We must also specify the SPI. The command for specifying the AH authentication key manually as follows:

```
Router(config-crypto-map)# set session-key {inbound |
outbound} ah SPI hex-key
```

The key must be specified for each direction, inbound and outbound. Also, one peer's inbound key must match the other peer's outbound key for the two peers

to communicate successfully over the IPSec tunnel. Finally, the SPIs must also match in a similar fashion.

Since we did not select any authentication in our transform set, we do not apply this command.

The command for specifying the ESP encryption keys is as follows:

```
Router(config-crypto-map)# set session-key {inbound |
outbound} esp SPI cipher hex-key [authenticator hex-key]
```

The key must be specified for each direction, inbound and outbound. Also, one peer's inbound key must match the other peer's outbound key for the two peers to communicated successfully over the IPSec tunnel. The SPIs must also match in a similar fashion. Finally, the `authenticator` key word and its associated hex key is only specified if the transform set previously selected includes ESP authentication (in addition to the encryption).

In our example, we previously selected encryption type esp-des in our transform set. So we specify the encryption key manually for each direction on the two peers as follows:

```
RouterA(config-crypto-map)# set session-key inbound esp 410
cipher 9876543210abcdef

RouterA(config-crypto-map)# set session-key outbound esp
420 cipher fedcba0123456789

RouterB(config-crypto-map)# set session-key inbound esp 420
cipher fedcba0123456789

RouterB(config-crypto-map)# set session-key outbound esp
410 cipher 9876543210abcdef
```

Note that different encryption algorithms have different minimum key length requirements. For DES, the minimum key length requirement is 16 hex digits.

This completes the configuration of the crypto map.

7.4 Apply Crypto Map to Interface

The final step in configuring IPSec on Cisco routers consists of applying the crypto map to the interface on which secure communication is to take place. The command for applying the crypto map to the desired interface (after entering into the desired interface) is as follows:

```
Router(config-if)# crypto map map-name
```

In our example, we apply the crypto map to the serial interface as follows:

```
RouterA(config-if) crypto map RouterACryptoMap

RouterB(config-if) crypto map RouterBCryptoMap
```

7.5 Example Configuration Script for IPSec using Manual Key Management

This completes the IPSec configuration for both routers using manual key management. The relevant portions of the configurations for each router are shown below.

```
RouterA# show running-config
Building configuration...

.
.
.

crypto ipsec transform-set RouterATransform esp-des

!
crypto map RouterACryptoMap 10 ipsec-manual
 set peer 10.10.1.2
 set session-key inbound esp 410 cipher 9876543210abcdef
 set session-key outbound esp 420 cipher fedcba0123456789
 set transform-set RouterATransform
 match address 110

.
.
.

interface Serial0/0
 description IpSecToHostileNetwork
 ip address 10.10.1.1 255.0.0.0
 crypto map RouterACryptoMap

.
.
.

access-list 110 permit ip 192.168.20.0 0.0.0.255
172.16.10.0 0.0.0.255

.
.
```

```
.
```

Figure 2: Relevant sections of Router A's configuration using manual key management.

```
RouterB# show running-config
Building configuration...

.
.
.

crypto ipsec transform-set RouterBTransform esp-des

!
crypto map RouterBCryptoMap 20 ipsec-manual
 set peer 10.10.1.1
 set session-key inbound esp 420 cipher fedcba0123456789
 set session-key outbound esp 410 cipher 9876543210abcdef
 set transform-set RouterBTransform
 match address 120

.
.
.

interface Serial0/0
 description IpSecToHostileNetwork
 ip address 10.10.1.2 255.0.0.0
 crypto map RouterBCryptoMap

.
.
.

access-list 120 permit ip 172.16.10.0 0.0.0.255
192.168.20.0 0.0.0.255

.
.
.
```

Figure3: Relevant sections of Router B's configuration using manual key management.

We now explain the steps involved in configuring key management using IKE.

8 Steps for Configuring IPSec Using IKE

Majority of the steps described above (and listed in section 7) for configuring crypto IPSec using manual key management are also applicable to configuring IPSec using IKE. The general procedure that we will use to configure IPSec on Cisco routers using IKE is as follows:

- Create access list
- Configure transform set
- Configure crypto map with IKE
- Apply crypto map to interface

From the above steps, the main difference between configuration using manual key management and that using IKE lies in the configuration of crypto maps. For the sake of brevity, and to emphasize the differences between the two approaches, only the sections that are different between the two approaches are presented here.

8.1 Configure Crypto Map with IKE

Similar to the manual key management case, we will break down the configuration of crypto maps using IKE into the following steps:

- Create a crypto map with IKE
- Specify the data traffic to secure
- Specify the peer node
- Specify the transform set to use
- Configure IKE

Note that from the above list, all the steps are identical except for the first and the last step.

8.1.1 Create a Crypto Map with IKE

A crypto map is created by specifying the name of the map, the sequence number of the map, and the type of key management that will be used between the two peers. The command for creating a crypto map with IKE is as follows:

```
Router(config)# crypto map map-name seq-num ipsec-isakmp
```

In our example, we create the crypto map with IKE as follows:

```
RouterA(config)# crypto map RouterACryptoMap 10 ipsec-
isakmp

RouterB(config)# crypto map RouterBCryptoMap 20 ipsec-
isakmp
```

8.1.2 Configure IKE

IKE uses its own set of keys to create a security association to use in exchanging keys between the two ends prior to even setting up any IPSec parameters. Each of the two peers can create multiple policies for use in communicating with the other end. At least one of the policies on one end must be compatible with at least one of the policies on the other. In order to compare them, the policies at each end must be exchanged. For this to happen in a secure manner, authentication and encryption parameters must be agreed upon prior to any exchange of information.

The steps to configuring the security keys using IKE are as follows:

- Enable IKE
- Create IKE Policy
- Define Key

8.1.2.1 Enable IKE

IKE is enabled by default on Cisco routers with IPSec; if it is not, the command to enable it is as follows:

```
Router(config)# crypto isakmp enable
```

8.1.2.2 Create IKE Policy

In creating an IKE policy for negotiation of security associations for exchanging keys, a number of parameters must be defined at each of the peers. The steps for defining these parameters are as follows:

- Define Policy Priority
- Specify Encryption Algorithm
- Specify Hash Algorithm
- Define Authentication Method
- Specify Diffie-Hellman Group Identifier
- Specify Security Association's Lifetime

The selection of these parameters depends on one's security needs as well as compatibility with the peer's parameters. It is important to pay attention to the selection of these parameters as they must match between the two peers in order for IKE to function. (The exception is the lifetime parameter where, the lifetime of the remote peer's policy must be less than or equal to the lifetime of the policy being compared, for a match to occur. If this is true, the shorter lifetime of the remote peer's policy is used.[9])

8.1.2.2.1 Define Policy Priority

Multiple policies can be created when negotiating IKE between two peers. Each policy has to have a priority associated with it. These policies are then compared

[9] Cisco, "Configuring Internet Key Exchange Security Protocol", p. 415.

with the peer's policies in order of priority. The command for defining a policy is as follows:

```
Router(config)# crypto isakmp policy priority
```

In our example, we define the policy as follows:

```
RouterA(config)# crypto isakmp policy 10

RouterB(config)# crypto isakmp policy 20
```

Note that this command will put us into the ISAKMP mode so that all subsequent parameters can be applied to the policy that was defined here.

8.1.2.2.2 Specify Encryption Algorithm

Cisco allows the use of the standard DES (56-bit) encryption as well as the more robust Triple DES (168-bit) encryption. The command for specifying the encryption algorithm to be used in IKE is as follows:

```
Router(config-isakmp)# encryption {des | 3des}
```

In our example, we choose to use Triple DES and specify the encryption algorithm as follows:

```
RouterA(config-isakmp)# encryption 3des

RouterB(config-isakmp)# encryption 3des
```

8.1.2.2.3 Specify Hash Algorithm

Cisco allows the use of the SHA or MD5 hash algorithms. The command for specifying the hash algorithm to be used in IKE is as follows:

```
Router(config-isakmp)# hash {sha | md5}
```

In our example, we choose to use SHA and specify the hash algorithm as follows:

```
RouterA(config-isakmp)# hash sha

RouterB(config-isakmp)# hash sha
```

8.1.2.2.4 Specify Authentication Method

Cisco provides three methods for authenticating the exchange of keys: 1) RSA signatures, 2) RSA encrypted nonces, and 3) pre-shared keys. The first two require the use of a Certificate Authority (CA) server while the last one requires that the two peers already have knowledge of the keys previously. The command for specifying the authentication method is as follows:

```
Router(config-isakmp)# authentication {rsa-sig | rsa-encr |
pre-share}
```

In our example, we select the pre-share authentication method and specify the authentication method as follows:

```
RouterA(config-isakmp)# authentication pre-share

RouterB(config-isakmp)# authentication pre-share
```

Note that pre-shared keys should not be used in a large network since sharing of keys can become cumbersome if too many peers are involved; in such cases, the implementation of a Public Key Infrastructure (PKI) is recommended. However, for a small organization which can not afford to invest in a comprehensive PKI infrastructure, pre-shared method for authenticating keys is suitable.

8.1.2.2.5 Specify Diffie-Hellman Group Identifier

Diffie-Hellman algorithm is a way to generate a key for use in a particular communication session between two peers based on public and private information held by each peer. As such, it provides for greater security as compared to other key establishment protocols which rely on simply transporting the key which was generated by input from only one of the two communicating peers.

Cisco IOS provides two levels of security: 1) a 768-bit or 2) 1024-bit. (Add-on feature to Cisco IOS has been introduced to provide greater security by using group 5 but at a higher CPU cost.)[10]

The command for specifying the Diffie-Hellman group is as follows:

```
Router(config-isakmp)# group {1 | 2}
```

This is an optional parameter for which the default value is 1. We will accept this default value as part of our configuration.

8.1.2.2.6 Specify Security Association's Lifetime

Lifetime parameter specifies how long a particular policy should be used until a new one is negotiated. The smaller the lifetime, the more secure the link. However, smaller lifetimes result in longer setup time since negotiations are required more often.

The command for specifying the security association lifetime is as follows:

[10] Cisco, "Diffie-Hellman Group 5", p. 1.

```
Router(config-isakmp)# lifetime seconds
```

This is an optional parameter for which the default value is 86,400 seconds. We will accept this default value as part of our configuration.

8.1.2.3 Define Key

Finally, since we chose to use the pre-shared option of exchanging keys, we have to specify the pre-shared key that we will be using for each peer. The command for specifying the pre-shared key is as follows:

```
Router(config)# crypto isakmp key keystring address peer-
address

RouterA(config)# crypto isakmp key thiskey address
10.10.1.2

RouterB(config)# crypto isakmp key thiskey address
10.10.1.1
```

8.2 Example Configuration Script for IPSec using IKE

The configuration of the remaining parameters is the same as that described in the section for manual key management.

This completes the IPSec configuration for both routers using IKE. The relevant portions of the configurations for each router are shown below.

```
RouterA# show running-config
Building configuration...

.
.
.

crypto isakmp policy 10
 encryption 3des
 hash sha
 authentication pre-share
crypto isakmp key thiskey address 10.10.1.2

!
crypto ipsec transform-set RouterATransform esp-des

!
crypto map RouterACryptoMap 10 ipsec-isakmp
 set peer 10.10.1.2
 set transform-set RouterATransform
 match address 110
```

```
.
.
.

interface Serial0/0
 description IpSecToHostileNetwork
 ip address 10.10.1.1 255.0.0.0
 crypto map RouterACryptoMap

.
.
.

access-list 110 permit ip 192.168.20.0 0.0.0.255
172.16.10.0 0.0.0.255

.
.
.
```

Figure 4: Relevant sections of Router A's configuration using IKE.

```
RouterB# show running-config
Building configuration...

.
.
.

crypto isakmp policy 20
 encryption 3des
 hash sha
 authentication pre-share
 crypto isakmp key thiskey address 10.10.1.1

!
crypto ipsec transform-set RouterBTransform esp-des

!
crypto map RouterBCryptoMap 20 ipsec-isakmp
 set peer 10.10.1.1
 set transform-set RouterBTransform
 match address 120

.
.
.
```

```
interface Serial0/0
 description IpSecToHostileNetwork
 ip address 10.10.1.2 255.0.0.0
 crypto map RouterBCryptoMap

.
.
.

access-list 120 permit ip 172.16.10.0 0.0.0.255
192.168.20.0 0.0.0.255

.
.
.
```

Figure 5: Relevant sections of Router B's configuration using IKE.

9 Conclusion

In this paper, we have explained the method of configuring tunnel-mode IPSec on Cisco routers to create a VPN over an insecure network (i.e. internet) for the purposes of providing secure communications between two sites. We have seen that at a small scale, IPSec can be configured using manual key management. When the number of interconnected sites is large, an automated key management protocol, IKE, is more feasible. In both cases, Cisco routers offer a comprehensive solution for configuring IPSec to allow businesses to securely share data over the pre-existing internet infrastructure, thus providing an economical and cost-effective alternative to leased lines.

References

Cisco Systems, Inc. “About the Cisco IOS Software Documentation.” http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121sup/121csum1/csdabt1.pdf (October 2003)

Kent, S., Atkinson, R. “Security Architecture for the Internet Protocol.” Network Working Group Request for Comments 2401. November 1998. http://www.ietf.org/rfc/rfc2401.txt (October 2003)

Harkins, D., Carrel, D. “The Internet Key Exchange (IKE).” Network Working Group Request for Comments 2409. November 1998. http://www.ietf.org/rfc/rfc2409.txt (October 2003)

Stallings, William. “IP Security.” The Internet Protocol Journal. Volume 3, Number 1 (March 2000): 11-26.

Quiggle, Adam. Implementing Cisco VPNs. Berkeley: McGraw-Hill, 2001. 281 – 362.

Cisco Systems, Inc. “Configuring IPSec Network Security.” http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt4/scdipsec.pdf (October 2003).

Electronic Frontier Foundation. “Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design – How Federal Agencies Subvert Privacy.” http://www.eff.org/descracker/ (October 2003).

Cisco Systems, Inc. “Configuring Internet Key Exchange Security Protocol.” http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt4/scdike.pdf (October 2003).

Cisco Systems, Inc. “Diffie-Hellman Group 5.” http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtgroup5.pdf (October 2003).

Cisco Systems, Inc. “Configuring an IPSec Tunnel Between Routers with Duplicate LAN Subnets.” http://www.cisco.com/warp/public/707/same-ip.pdf (October 2003).

Cisco Systems, Inc. “Configuring IPSec Manual Keying Between Routers.” http://www.cisco.com/warp/public/707/manual.pdf (October 2003).

Maughan, D., Schertler, M., Schneider, M., Turner, J. “Internet Security Association and Key Management Protocol (ISAKMP).” Network Working Group

Request for Comments 2408. November 1998. http://www.ietf.org/rfc/rfc2408.txt (October 2003)

Kent, S., Atkinson, R. “IP Encapsulating Security Payload (ESP).” Network Working Group Request for Comments 2406. November 1998. http://www.ietf.org/rfc/rfc2406.txt (October 2003)

Kent, S., Atkinson, R. “IP Authentication Header.” Network Working Group Request for Comments 2402. November 1998. http://www.ietf.org/rfc/rfc2402.txt (October 2003)

Sharma, Rajesh K., Mogha, Rashim. Cisco Security Bible. Hungry Minds, Inc., 2002. 427 - 447, 449 - 469.

Ives, Millie. “Implementing Site-To-Site IPSec VPNs Using Cisco Routers.” May 4, 2001. http://www.giac.org/practical/gsec/Millie_Ives_GSEC.pdf (October 2003).