Global Information Assurance Certification Paper

An Overview of Digital Certificates and How They Are Used in VPN Authentication

Practical assignment version 1.4b
Option 1

By Steve Pitts

March 17, 2003

Abstract:
The purpose of this paper is to provide the reader with an overview of digital certificates and how they are used to authenticate VPNs. An example is provided illustrating the use of certificates issued by an RSA KEON CA server to authenticate a tunnel between a Cisco VPN3000 and a Cisco 2651 router. This discussion pertains to IPSec VPNs which utilize IKE.

Introduction:
In order to establish a secure connection between VPN gateways, an authentication method must be employed to validate the identity of the participants. Proper device authentication prevents unauthorized access to the VPN network and provides origin authentication. Two common means of authenticating VPN tunnels are pre-shared keys and digital certificates. Both are considered strong authentication methods. A pre-shared key is simply an alphanumeric string known by the remote parties connected to the VPN. The key is established and provided to the remote parties by some out-of-band means and configured in the VPN gateways before the tunnel connection is attempted. (1) The pre-shared key is never actually transmitted over the VPN network, but is instead incorporated into the formation of key material exchanged during Main Mode. Pre-shared key authentication is very common and is easier to configure than Digital Certificates. However, digital certificates offer several advantages for VPNs and are considered a stronger authentication method. (2) When dealing with large VPN networks, managing large numbers of pre-shared keys can become cumbersome. Digital certificates are a much more scalable authentication solution. Since pre-shared keys are bound to an IP address, and since the pre-shared key is a factor in the formation of the keying material, VPN gateways must maintain an index matching pre-shared keys with addresses. Digital certificates are not bound to an IP address, but instead can use unique, static identification information validated by the Certificate Authority. This allows remote users with a dynamically assigned IP address to authenticate using identification information contained in the certificate, and the keying material is formed independent of information in the certificate. The role of the Certificate Authority helps re-enforce the trust relationship between peers participating in the VPN. Another advantage of using digital certificates is CRLs. Certificates of compromised devices can be placed on a Certificate Revocation List published by the CA. The CA's clients will no longer authenticate to a device presenting a certificate contained in the Certificate Revocation List. The certificate system provides a convenient means for administrators to assign and revoke access privileges to users. Since certificates are tamper-proof and cannot be altered, there is no need to protect or hide them. There are disadvantages to using digital certificates. Employing digital certificates is inherently complicated. Some complications are due to inconsistent terminology among vendors and interoperability issues. A system for issuing, storing and revoking certificates must be managed, the VPN gateways must be configured to operated with the certificates and administrators must be trained to support the certificate infrastructure. Before beginning this discussion, it is important to distinguish

between data authentication and end-entity authentication. Data authentication is performed during VPN processes to maintain integrity of the data stream being sent over the VPN tunnel. End-entity authentication is the process of verifying the identity of the participants of the communication. I will now attempt to clarify some of the concepts involved with digital certificate authentication of VPNs.

What is a digital certificate?

A digital certificate can be thought of as an electronic passport used to validate the identity of the owner. The owner presents the certificate during secure communication session initiation to provide proof of identity to remote parties. An end user obtains a certificate by generating a certificate request and sending the request to a trusted Certificate Authority that processes the request, validates the identity of the end user, and issues a certificate to the end user. Certificate Authorities are the pillar of the infrastructure since they are responsible for vouching for the identity of the requestor and certifying that the requestor is a valid user of the system. The CA is trusted throughout the system and therefore, any entity the CA certifies is also trusted The certificate itself is an electronic document created using public key cryptography containing identity and encryption information for the certificate authority and the owner of the certificate. Digital Certificates are used in a wide variety of electronic communication roles such as VPN tunnel authentication, secure e-mail certificates, SSL, TLS and SET. Digital Certificates are a core component of the X.509 standard. X509 defines authentication protocols using public-key certificates and digital signatures. (3)

Public/Private key cryptography:

Digital certificates are based on the concept of public/private key cryptography, an asymmetrical encryption method utilizing a public and a private key pair. The keys are created by a mathematical process as a matched pair. Either key can perform encryption or decryption, but a message encrypted with a particular key cannot be decrypted by that same key. That message can only be decrypted by the other key in the key pair. A message encrypted by the public key can only be decrypted by the matching private key and vice versa. In public/private cryptographic systems, public keys are exchanged freely while private keys must be kept secret. When I wish to send an encrypted message to a remote party, I will obtain their public key. I will then encrypt my message using their public key and send it. Only the remote party holding the matching private key can decrypt the message. (4) The security is based on the difficulty of calculating the private key given the public key. The most widely used algorithm is the RSA algorithm created by Rivest, Shamir and Adleman. The public/private key system alone provides confidentiality, but does not provide origin authentication since there is no means to verify who authored the message. Digital Certificates bind the identity of the public key to its owner. By doing so, certificate systems add a level of security to communications by providing remote party authentication and non-repudiation. (5)

What are the types of digital certificates?
There are several types of certificates used for a variety of applications. For this discussion we will be concerned with X509 Certificates. In VPNs, certificates are typically employed as Root Certificates and End-Entity Certificates. Root certificates belong to Certificate Authorities. End-Entity Certificates are installed in VPN gateways. Root certificates may also be called CA certificates. End-Entity certificates may also be called Identity, Client or Self- Certificates.

How do certificates work?
Assume there are two entities, Host A and Host B which desire to use certificates to authenticate a VPN. They will each need to enroll with a trusted CA, obtain the CA's certificate and obtain self-certificates. First, each entity will enroll with the CA and obtain the CA's certificate. The CA certificate contains identity information for the CA and the CA public key. To obtain a self-certificate, Host A and B must each generate a public/private key pair. Each Host will then submit his public key and identification information to a trusted Certificate Authority. The Certificate Authority will validate the user's identity and assemble the user's identification and public key information into a digital document. The CA will then "sign" this document. The CA signs the document by hashing the certificate contents with its signing algorithm. The hash is then encrypted using the CA's private key and included in the certificate. The CA will then issue the certificate to Host A and B. When Host A wishes to authenticate a session with Host B, Host A will send its self-certificate to Host B along with information about the CA that issued the certificate. Since Host B subscribes to the same CA, Host B will have the CA certificate containing the CA public key and information specifying the signing algorithm used by the CA. Host B can then use the CA public key to decrypt the self certificate of Host A. Host B can now run the CA signing algorithm and re-create a hash of Host A's certificate. If the re-created hash of Host A's self-certificate matches the hash created by the CA, the certificate is deemed valid. (6)

Functions of Certificate Authorities:
CAs validate the identity of clients and match unique client identity with the client's public key. CAs service certificate requests from clients, issue, deny, store and revoke certificates. Certificates are issued in a variety of ways including a manual cut and paste method, TFTP and HTTP. Some CA servers support the Simple Certificate Enrollment Protocol that allows the certificate request and issuing process to take place automatically. Manual methods allow the CA to operate off-line. SCEP requires network access to the CA. The CA maintains a list of revoked certificates contained in a Certificate Revocation List or CRL, which can be published and loaded into client devices.

CA chains:
CAs can exist in a hierarchy or "trust chain" in which a Root CA stands at the top of a network containing other subordinate CAs. The Root CA must have a self-signed CA certificate. The subordinate CAs must have CA certificates signed by

the Root CA. Both parties in a secure communication link must recognize and trust the CA that issued the identity certificates presented by the participants in the session. In smaller networks, this is easy. However in larger networks, it is not practical for all users to recognize all CAs. In such cases it is necessary for a trust chain to link subordinate CAs to a Root CA which is recognized by both parties. (7) For example, an enterprise may operate a standalone "self signed" CA server to issue certificates to clients within the enterprise. This will allow devices within the enterprise to authenticate to each other using the enterprise CA. However, to allow secure communication sessions to be created to outside parties which may not have access to the enterprise CA, the enterprise CA must be chained to an external trusted CA recognized by the outside parties and the enterprise CA. Examples of CAs widely used for this purpose are VeriSign, Thwate, E-trust, RSA, etc.

Contents of a certificate:
The contents of a certificate are specified by the X.509 standard for directory services. A typical certificate will contain:

Version number:	version of the certificate format
Serial number:	unique number associated with the certificate
Signature algorithm:	algorithm used to sign the certificate
Issuer information:	Information about the CA that created and issued the certificate
Validity Period:	the certificate is invalid beyond this time envelope
Subject Name:	the identity information about the holder of the certificate, in the LDAP format C=US, CN=MyCompany
Public Key:	the Public Key of the holder of the certificate
Subject Alt Name:	optional additional subject identity information Common Subject Alt name formats are IP Address, Fully Qualified Domain Name or User Fully Qualified Domain Name, RFC822 name or Email address, LDAP or Directory Name
Signature:	a hash of the contents of the certificate encrypted with the CA Private key

Note that the exact contents may vary depending on the CA server configuration and the type of certificate and optional extensions included by the CA. (8)

SCEP:
Originally developed by Verisign for Cisco, Simple Certificate Enrollment Protocol (SCEP) may be used in VPN networks to facilitate certificate management. (9) SCEP uses HTTP to exchange certificate registration messages between CAs and network devices. SCEP supports key distribution, certificate enrollment, certificate queries, revocation and CRL queries. (10) Enrollment requests may be

authenticated manually or automatically using pre-shared secrets. Automatic enrollment allows the entire process to take place without the intervention of the CA administrator. MD5 fingerprints are employed in the SCEP messages to prevent "man-in-the-middle" attacks. (11) Manual enrollment requires the CA administrator to process each certificate request and elect to issue, defer or deny the certificate to the requestor. During the manual enrollment process, the requestor will send SCEP polling messages to the CA to retrieve the certificate. (12) The polling interval can be used by the CA to confirm the identity of the requestor by out of band means. (13)

Certificate Revocation and CRLs:
Certain events may require that a certificate be revoked before the end of its validity period. The X.509 standard utilizes the Certificate Revocation List to publish revoked certificate information. Certificates revoked by the CA are placed on a Certificate Revocation List that is periodically published by the CA. The CRL contains the certificate serial number, revocation date, issuer name, update fields and reason for revocation. To ensure authenticity, the contents of the CRL are signed by the private key of the CA. (14) It is the responsibility of subscribers to the CA retrieve the CRL from the CA. Certificate clients refer to the CRL during the authentication process and will drop connections attempts from devices whose certificates appear in the CRL. Valid reasons for certificate revocation include compromise of the CA, compromise of the certificate holder's private key, and affiliation or privilege change of the certificate holder. Certificates issued to end-entities contain a CRL Distribution Point (CDP). The CDP contains information that tells the end-entity where to get CRLs. (15)

Applications of Digital Certificates in SSL:
Digital certificates play an important role in providing security for Web transactions used in e-commerce. Web servers employ the Secure Sockets Layer (SSL) protocol to authenticate and encrypt transactions that carry sensitive data such as credit card information. Authentication provides assurance to web users that personal data is sent only to trusted Web sites. Encryption provides confidentiality. SSL employs X509 certificates to support these mechanisms. Web servers utilize a site Certificate issued by a Certificate Authority. The site certificate contains the server's ID information as well as its public key. The server will also possess a private key belonging to the public/private key pair. Web browsers typically contain a set of certificates issued by well-known trusted authorities which is pre-loaded during the installation of the client operating system. Using these certificates, SSL can provide server side authentication as well as client side authentication. In most cases, web browsers use server side authentication to obtain proof of the authenticity of web sites. Clients initiate a session to the server by sending a hello message. The hello message contains information about the client including preferences for encryption and compression methods. The server will respond to the client with similar information. If the server and client support mutually agreeable methods, an SSL session can be established. Once the session is established, the server will send

its site certificate to the client. The web browser now has access to the server's public key. The Client uses the web server's public key to verify the web server's certificate. In addition to verifying the validity dates and CA signature in the site certificate, the browser will also perform an identity check by comparing the URL that submitted the certificate to the URL specified in the ID information contained in the certificate. If these match, the browser will trust the site certificate. Once this authentication process is complete the web browser will create a symmetric encryption key. The browser then uses the server's public key to encrypt the symmetric encryption key. The browser then sends the key to the server. As explained earlier in the discussion of public/private key encryption, only the authenticated server that possesses the matching private key will be able to decrypt the encryption key sent by the web browser. Now that the server and the web browser have exchanged the symmetric encryption key, they will use the key to encrypt subsequent communications. (16) It is interesting to note that the convenience of this nearly transparent process exposes the web user to vulnerabilities. Pre-loaded site certificates encourage the user to let someone else decide who to trust. Also, most applications of SSL do not perform CRL checking. (17) This weakness can allow a rouge web site to use a compromised site certificate to entice unwary users into illegitimate session establishment.

How certificates are used in VPNs:

In order to create a framework for understanding the role of digital certificates in VPN authentication, it is necessary to briefly discuss some mechanisms employed to establish an IPSec VPN using Main Mode. IPSec is a set of protocols used to provide security for IP communication. Secure IP sessions are established using Security Associations (SAs) that specify the security services to be applied to data traffic. These include the encryption and authentication methods, algorithms, keys and key lifetimes, and SA lifetimes. IKE is a mechanism used to automatically establish and maintain SAs and exchange key information for IPSec. Internet Security Association and Key Management Protocol (ISAKMP) is incorporated into IKE and defines the processes used to authenticate peers, exchange keys and build SAs. ISAKMP requires strong authentication methods to ensure the integrity of the session establishment. (18) The ISAKMP processes occur in two phases. During the first phase, a management tunnel is established to protect Phase 2 negotiations. During the second phase, IPSec SAs are established. The IKE management tunnel and the IPSec tunnel each have their own set of security parameters that must be

negotiated by the peers. It is during the creation of the management tunnel in phase 1 of Main Mode in which remote party authentication occurs, so I will focus on the phase 1 exchanges and how they vary depending on use of pre-shared keys or digital certificates for remote authentication.

Phase 1 message exchanges for pre-shared key authentication:

1st exchange:
During the first two-way message exchange, the data encryption, authentication methods and parameters to be used for the management tunnel are negotiated and agreed upon by the peers participating in the VPN. Cookies are also exchanged. After the first exchange, the peers generate their Diffie-Hellman public values. The messages of this first exchange carry the same information for both remote authentication methods. (19)

2nd exchange:
During the second set of messages, the peers share their Diffie-Hellman public values and nonces. The cookies and nonces are pseudo-random numbers inserted into the messages. These help prevent replay attacks and maintain continuity of the session. They are also used to help prevent Denial of Service attacks. (20) Once these values have been exchanged, the primary root key SKEYID can be established which is used to create subsequent keys for IKE data encryption, authentication and IPSec. How the SKEYID is formed depends upon the remote authentication method being used. When using pre-shared key authentication, the SKEYID is defined as:

SKEYID = prf (pre-shared key, Nonce_I | Nonce_R)

Note that the pre-shared key is incorporated into the creation of the primary root key SKEYID. (21)

3rd exchange:
By the third exchange, each peer has enough information to begin encryption. The last exchange of Main Mode is transferred across this encrypted channel. In this last exchange, the peers share an ID payload and a Hash payload. The Hash payload contains a hash performed over several pieces of information exchanged between peers during the previous transactions including the SKEYID and serves to validate these initial exchanges. (22) The ID payload contains identification information. Note that in the previous exchanges, the only identification information the peers have about each other is their IP address. A third party could intercept phase 1 session traffic and spoof the IP address of one of the peers and initiate a phase 1 session in order to gain access to the VPN. However, unless the third party has the correct pre-shared key, the exchanged hashes will not match and the authentication will fail. Only parties having the correct pre-shared key will be able to create a correct hash payload. The Hash

links the information of the first two exchanges with the pre-shared key which only the peers know.

Phase 1 message exchanges for digital certificate authentication:

1st exchange:
The messages in the first exchange are the same for both authentication methods.

2nd exchange:
During the second exchange, the peers will share Diffie-Hellman public values and nonces. Certificate requests will also be contained in the second set of messages. (23) The SKEYID is slightly different in that it is formed using the nonces and the Diffie-Hellman shared value:

SKEYID = prf (Nonce_I | Nonce_R, DH_key)

Information contained in the certificates is not incorporated into the formation of the SKEYID.

3rd exchange:
As before, the 3rd exchange is encrypted and will contain a similarly constructed Hash payload and an ID payload. Certificates will also be exchanged at this time in a certificate payload. Before transmission, the Hash payload is signed by the private key of the peer. Once each party has completed the exchange, they can use the public key contained in the remote parties self-certificate to verify the validity of the signed Hash payload and authenticate the session. The subject name contained in the self-certificate must match the identification claimed in the ID payload. The encryption of the third exchange protects the identification information of the peers. In addition to origin authentication, digital certificates provide non-repudiation since the peer cannot deny participating in the phase 1 session. (24) After the successful completion of phase 1, phase 2 will establish IPSec and complete the VPN tunnel.

Before staring on the VPN configuration, a basic configuration is set up which permits IP traffic between the gateways. After verifying IP routing with a ping from Host A to Host B, we are at a good staring point for the VPN configuration. For the sake of brevity, I will omit this basic aspect of the configuration details. When setting up VPNs, it is frequently easier to configure the tunnel to use pre-shared keys. Once the tunnel is established with pre-shared keys, it is then easy to load certificates into the gateways and make the simple changes required to authenticate using certificates. This process is described below. We start by configuring the VPN3000 for IKE/IPSec using pre-shared keys.

VPN Configuration:

Setup:
Network Diagram: (Figure 1)
An IP network incorporating the VPN gateways is configured as shown in the diagram below:

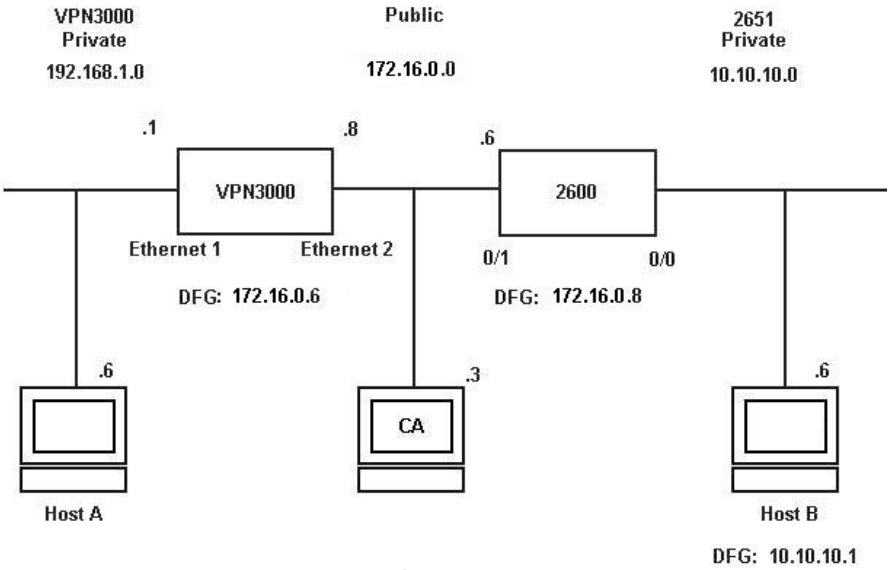

Figure 1: IP Network incorporating the VPN Gateways

VPN3000 IKE Proposals (Figure 2):
Configuration
System
Tunneling Protocols
IPSEC
IKE protocols:

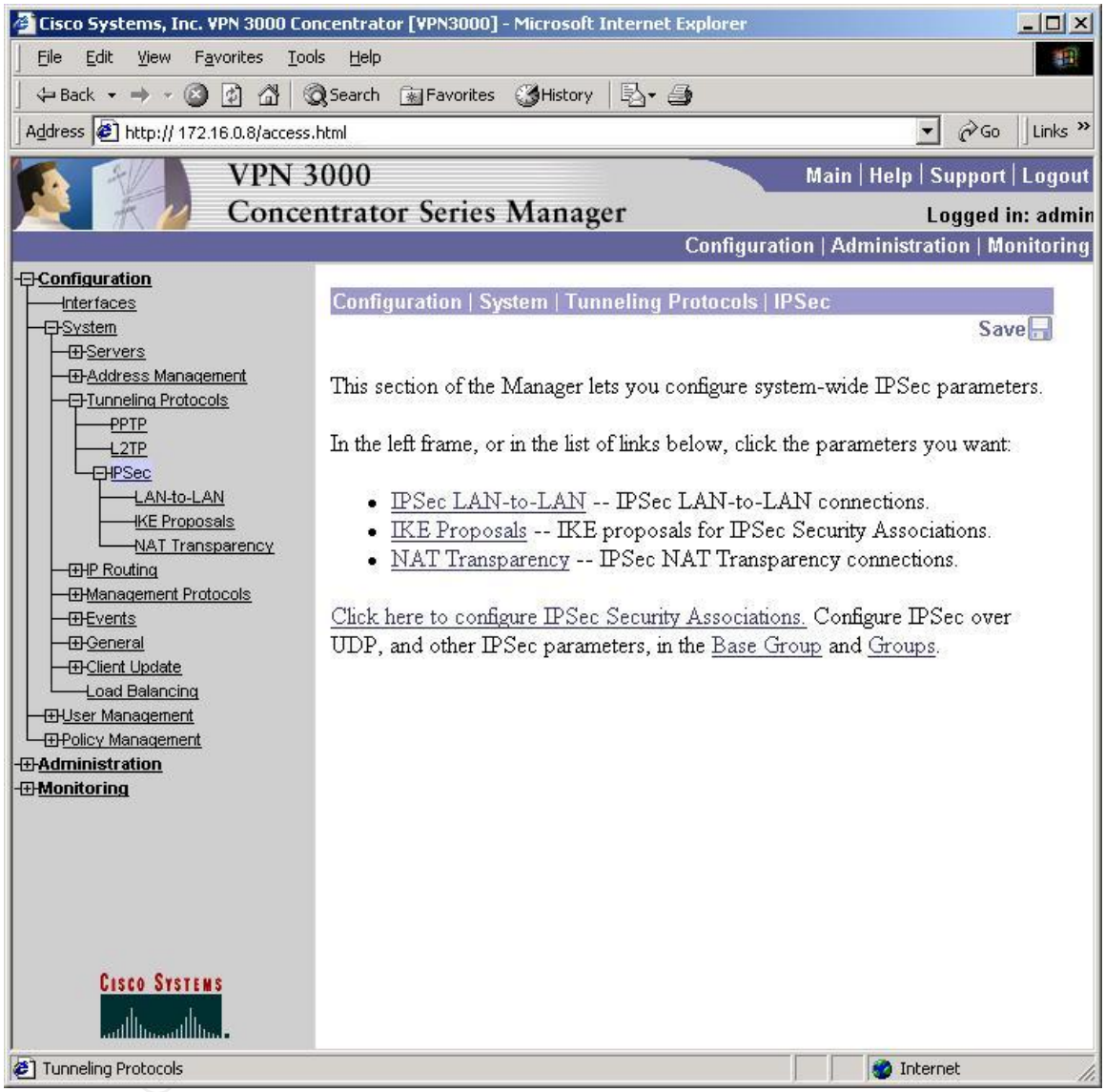

Figure 2: IKE Proposals

VPN3000 IKE Proposals/Phase 1 Parameters: (Figure 3)
This is the menu for configuring Phase 1 parameters. A new proposal can be added, or an existing proposal can be modified

Proposal Name:	IKE-DES-MD5
Authentication Mode:	pre-shared keys
Authentication Algorithm	Md5/HMAC-128
Encryption Algorithm	Des-56
Diffie-Hellman Group	group 1 768 bits
Lifetime Measurement	time
Data Lifetime	10000 (not used)
Time Lifetime	300

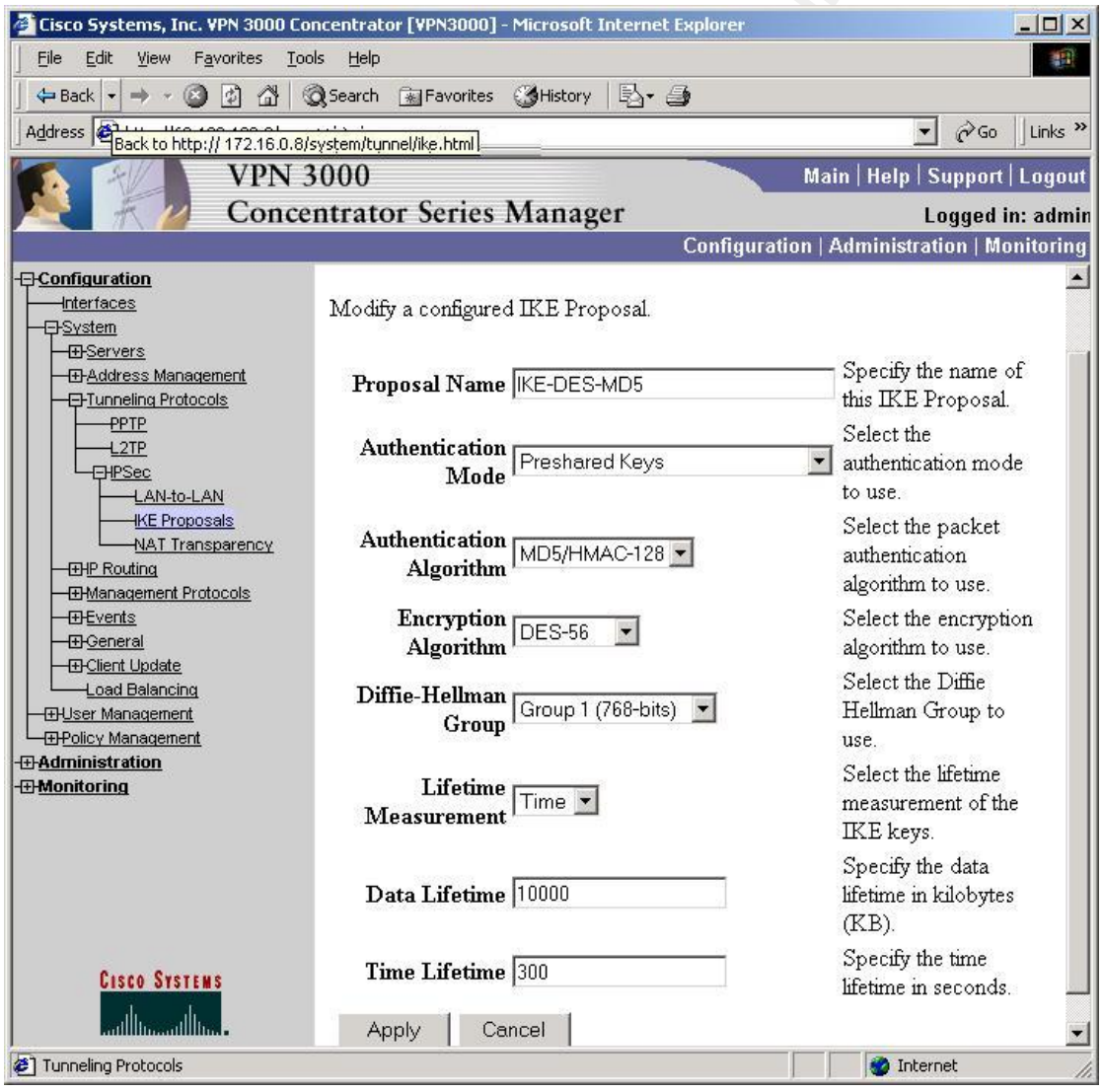

Figure 3: Phase 1 Parameters

VPN 3000 LAN-To-LAN / Phase 2 Parameters: (Figure 4).
This menu contains most of the phase 2 parameters:

name:	vpn3000
Interface	Ethernet 2
Peer	172.16.0.6
Digital Certificate	none
pre-shared key	1234567890123456
Authentication	esp/md5/hmac-128
encryption	des-56
Ike proposal	ike-des-md5
local network	192.168.1.0 0.0.0.255
remote network	10.10.10.0 0.0.0.255

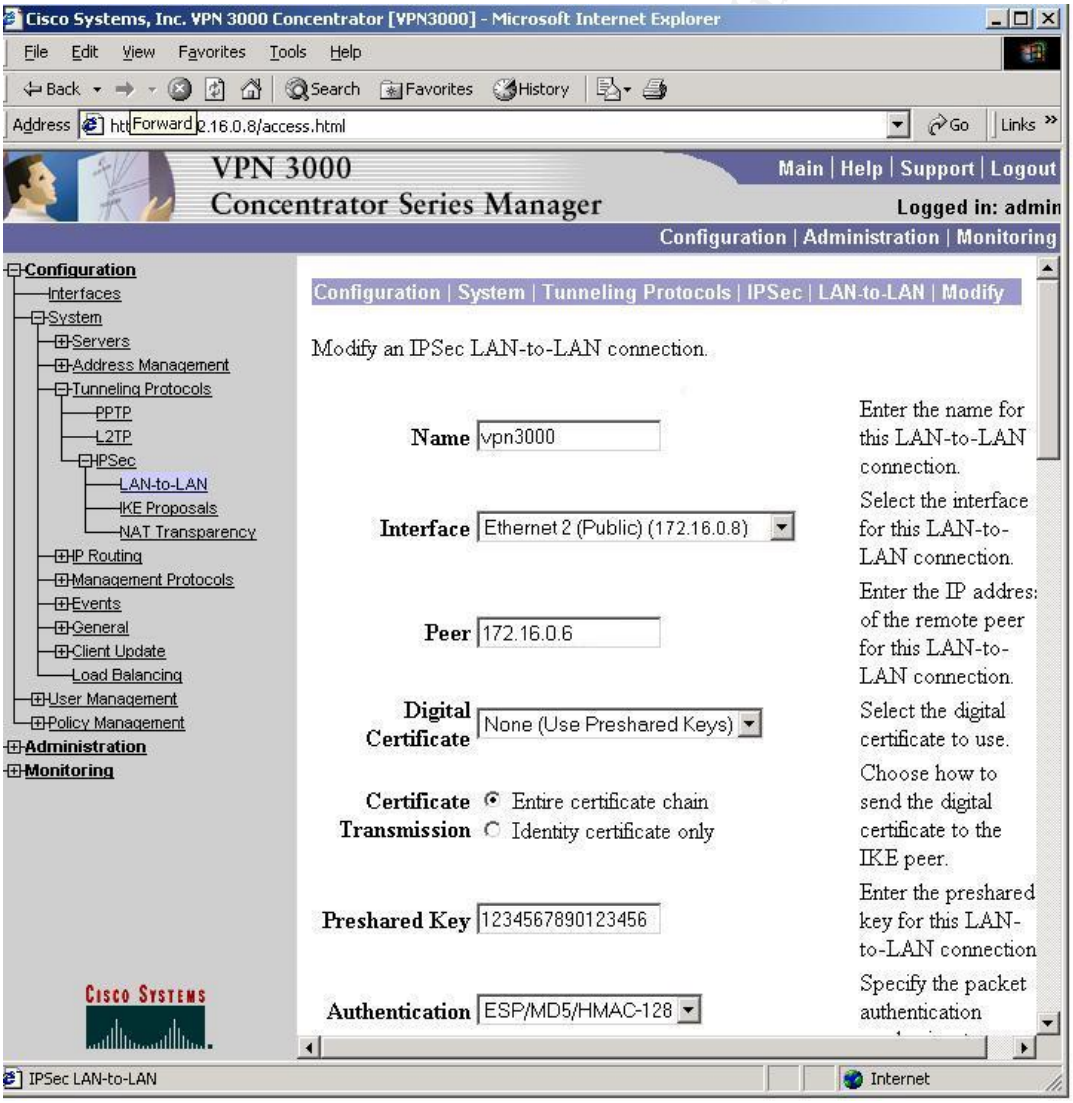

Figure 4: Phase Two Parameters

VPN 3000 Security Association menu : (Figure 5)
Configuration
policy management
traffic management
SAs
L2L:vpn3000
Perfect Forward Secrecy Group1 768 bits

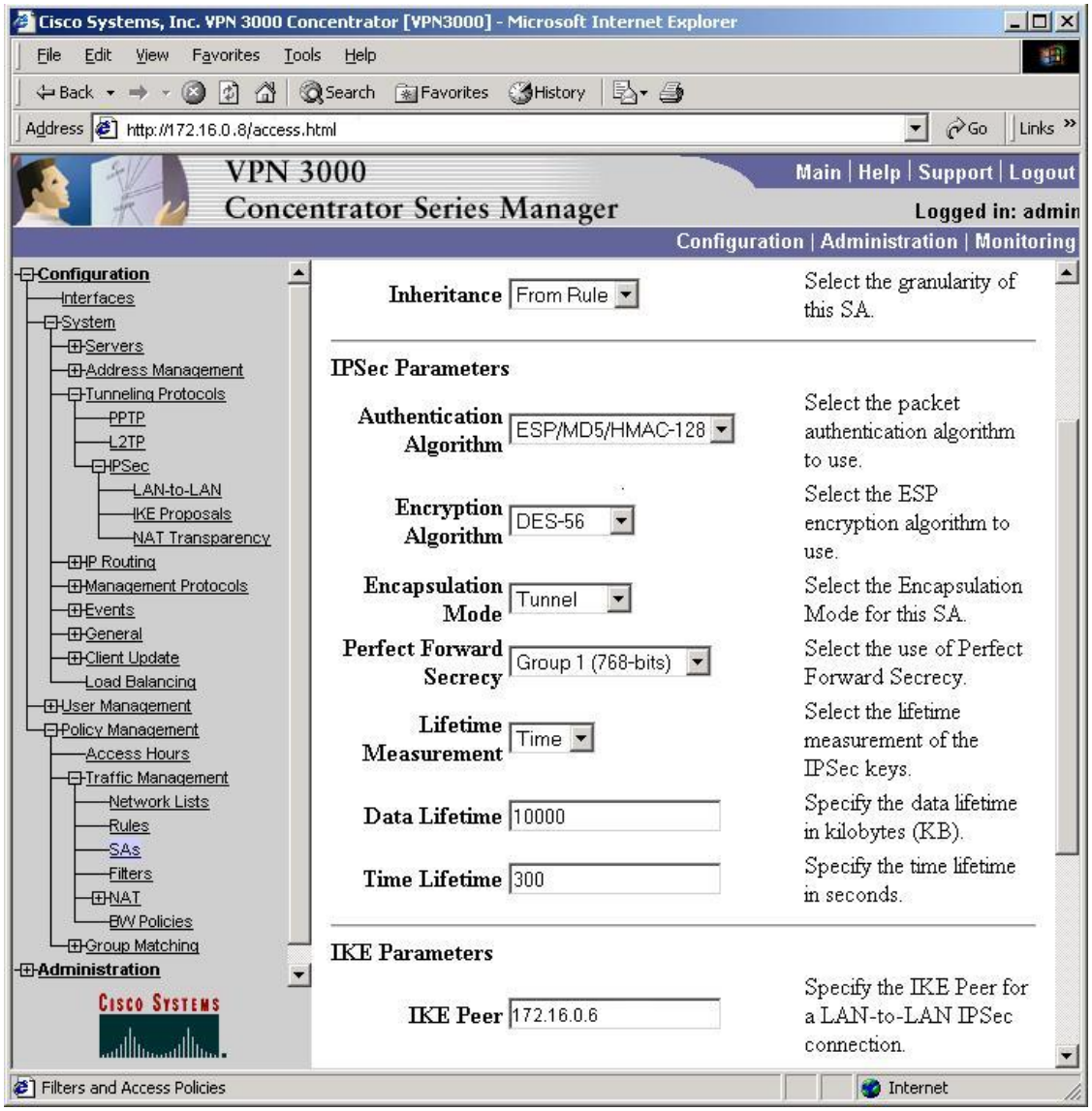

Figure 5: SAs

This completes the VPN3000 configuration for pre-shared keys.

2600 Preshared Key VPN Configuration:
A similar matching configuration will be built for the Cisco Router:

First, some useful debugs:

```
2621#debug crypto isakmp
Crypto ISAKMP debugging is on
2621#debug crypto verbose
verbose debug output debugging is on
2621#debug crypto pki transactions
Crypto PKI Trans debugging is on
2621#debug crypto ipsec
Crypto IPSEC debugging is on
2621#
```

Create an access list that will define traffic destined for the VPN:

```
2621(config)#ip access-list extended list150
2621(config-ext-nacl)#permit ip host 10.10.10.6 host 192.168.1.6
2621(config-ext-nacl)#exit
```

Set IKE phase 1 parameters including origin auth method, encryption and data auth method, DH group pre-shared key, and the lifetime of SA.

```
2621(config)#crypto isakmp enable
2621(config)#crypto isakmp policy 100
2621(config-isakmp)#authentication pre-share
2621(config-isakmp)#encryption des
2621(config-isakmp)#hash md5
2621(config-isakmp)#group 1
2621(config-isakmp)#lifetime 300
2621(config)#crypto isakmp identity address
2621(config)#crypto isakmp key 1234567890123456 address 172.16.0.8
255.255.255.240
```

Define IKE phase 2 parameters including the IPSEC transform set: esp with authentication using DES and MD5 SA lifetime, IKE peer, perfect forward secrecy, and map the access-list to the crypto-map.

```
2621(config)#crypto ipsec transform-set vpn3000 esp-des esp-md5-hmac
2621(cfg-crypto-trans)#mode tunnel
2621(cfg-crypto-trans)#exit
2621(config)#crypto ipsec security-association lifetime seconds 300
2621(config)#crypto map vpn3000 100 ipsec-isakmp
2621(config-crypto-map)#set peer 172.16.0.8
2621(config-crypto-map)#set transform-set vpn3000
2621(config-crypto-map)#exit
2621(config)#crypto map vpn3000 local-address fastEthernet 0/1
2621(config)#crypto map vpn3000 100 ipsec-isakmp
2621(config-crypto-map)#set pfs group1
2621(config-crypto-map)#match address list150
2621(config-crypto-map)#exit
```

Assign the crypto map to the WAN interface of the router.

```
2621(config)#interface fastEthernet 0/1
2621(config-if)#crypto map vpn3000
2621(config-if)#exit
2621(config)#exit
2621#
```

At this point, both devices are configured to support a pre-shared key VPN. A ping From Host A to Host B will cause the gateways to initiate VPN negotiations. When the tunnel is complete, a ping response should be noted from the originating machine. Using the show crypto ipsec sa and the show crypto ISAKMP SA commands on the 2621 Router will verify the successful completion of the VPN tunnel:

```
2621#show crypto isakmp sa
dst             src             state          conn-id    slot
172.16.0.8    172.16.0.6    QM_IDLE              1       0

2621#show crypto ipsec sa

interface: FastEthernet0/1
    Crypto map tag: vpn3000, local addr. 172.16.0.6

   local  ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer: 172.16.0.8
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 2, #pkts encrypt: 2, #pkts digest 2
    #pkts decaps: 2, #pkts decrypt: 2, #pkts verify 2
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts
     decompress     failed: 0
    #send errors 1, #recv errors 0

     local crypto endpt.: 172.16.0.6, remote crypto endpt.:
     172.16.0.8
     path mtu 1500, media mtu 1500
     current outbound spi: 26D26B2D

     inbound esp sas:
      spi: 0xD03B8E04(3493563908)
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 420, flow_id: 1, crypto map: vpn3000
        sa timing: remaining key lifetime (k/sec): (4607999/259)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x26D26B2D(651324205)
```

```
        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 421, flow_id: 2, crypto map: vpn3000
        sa timing: remaining key lifetime (k/sec): (4607999/259)
        IV size: 8 bytes
        replay detection support: Y

     outbound ah sas:

     outbound pcp sas:


   local  ident (addr/mask/prot/port): (10.10.10.6/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port):
(192.168.1.6/255.255.255.255/0/0)
   current_peer: 172.16.0.8
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress
failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 172.16.0.6, remote crypto endpt.:  172.16.0.8
     path mtu 1500, media mtu 1500
     current outbound spi: 0

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

2621#
```

Once the configuration is verified we are ready to reconfigure the VPN to use digital certificates.

Configuration of the RSA KEON CA Server:
The installation of the CA server is done using default settings. Once the server is installed an initial jurisdiction must be established under the system CA.

KEON CA :Create new CA Jurisdiction: (Figure 6).

CA operations
server system CA create
issuer: self (indicates this will be a self signed CA)
Jurisdiction create new jurisdiction
Nickname: Neptune
Common Name NEPTUNE
organizational unit IPSECLAB
organization PQ
country US
signing algorithm SHA-1
key size 1024 bits

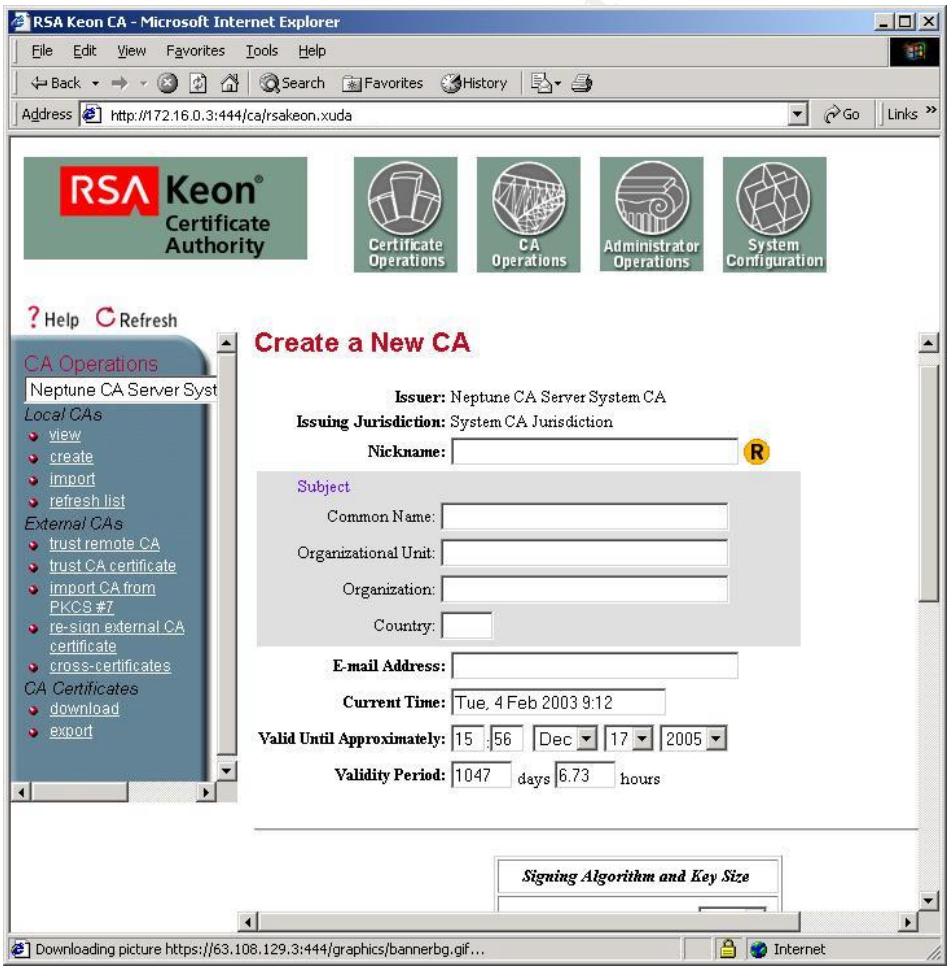

Figure 6: CA creation

KEON CA: Initial Jurisdiction Configuration: (Figure 7)
CA operations
NEPTUNE
View
Jurisdiction Configuration
Configure
Sections:Extension Profiles
General Profile Policy
Check requestor can select
Vettor can override
profile Choices: VPN/IPSEC
Sections:SCEP Autovetting
Enable autovetting of SCEP requests
Ip Address List:
172.16.0.6,172.16.0.8
save

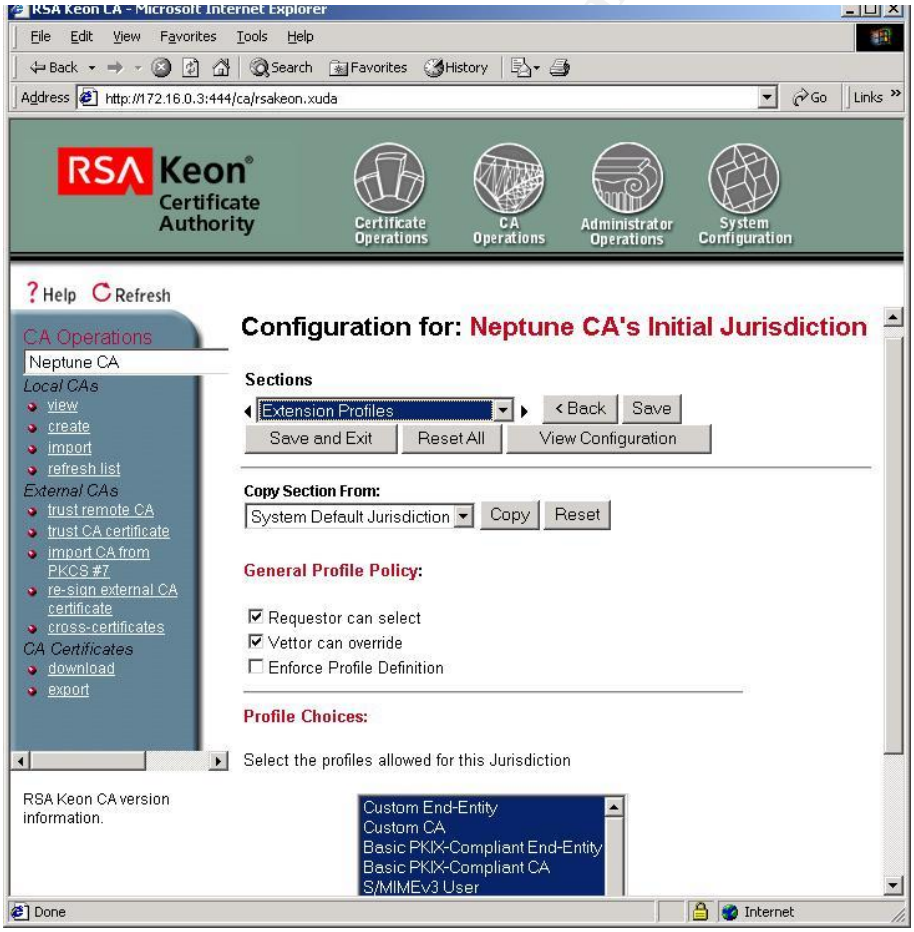

Figure 7: Initial Jurisdiction Configuration

KEON CA: SCEP Autovetting: (Figure 8)
CA operations
NEPTUNE
Jurisdiction Configuration
Configure
Sections:SCEP Autovetting
Enable autovetting of SCEP requests
Ip Address List:
172.16.0.6,172.16.0.8
save

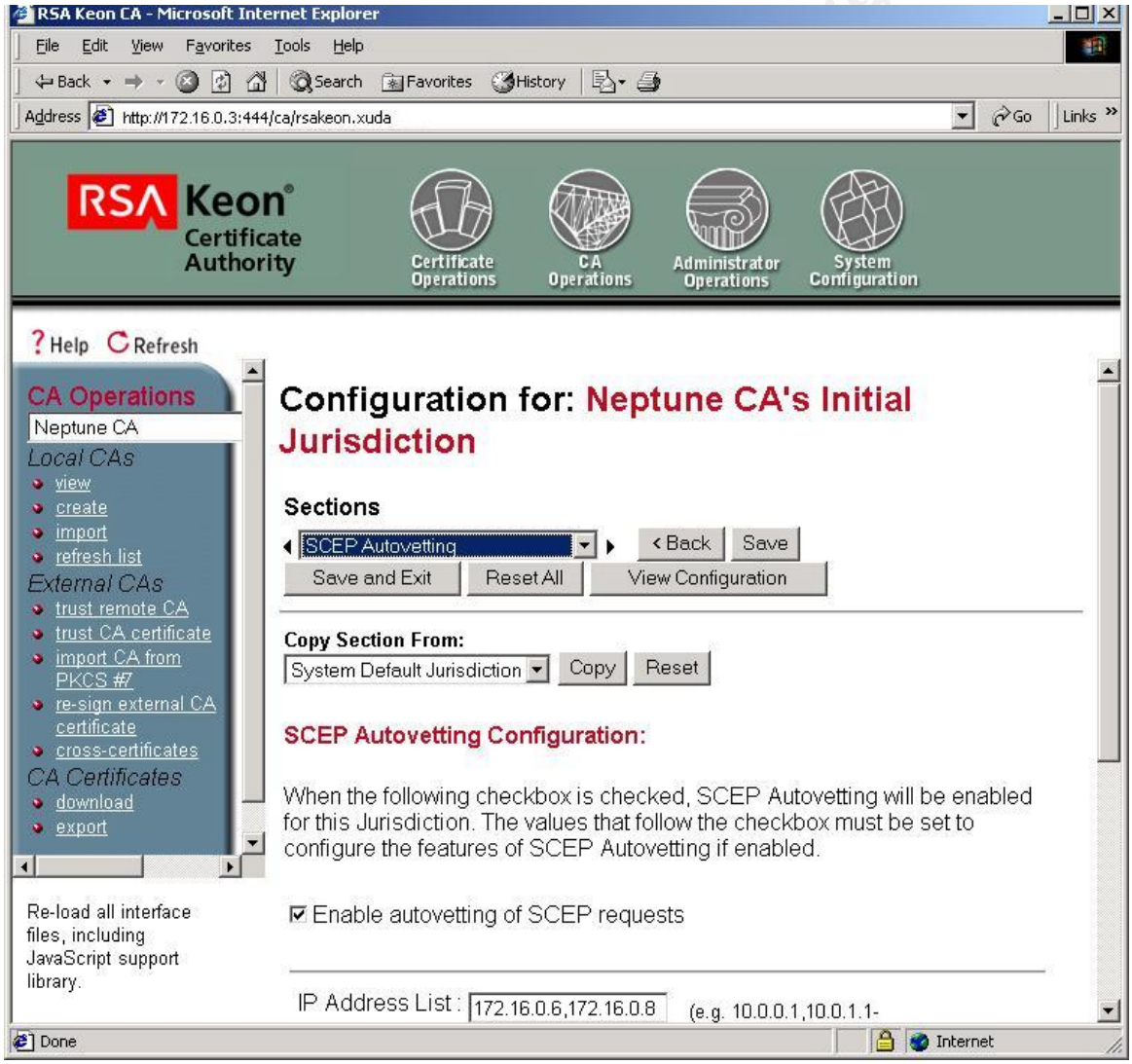

Figure 8: SCEP Autovetting

During this process, we have created a self-signed CA jurisdiction that will serve as the Root CA. We have also configured the CA to automatically process certificate requests from clients, and issue the identity certificates without intervention from an administrator. Note that the VPN/IPSEC extension is required for SCEP operation of the CA. The CA is now configured to process certificates using SCEP. After configuring the CA server, we need to obtain the Jurisdiction ID of the CA. This id number is used when specifying the CA's SCEP enrollment URL on the VPN gateways.

KEON CA :Certification Authority details: (Figure 9)
CA Operations
Neptune
View
Issuing Jurisdiction ID: 8bae1b88222ca2ee3dc5ad9008f8075c343aff30

Figure 9: Obtaining the Jurisdiction ID

Time Settings:
Before loading certificates into the VPN gateways, it is essential to synchronize the time and date settings on the Gateways and the CA server. If the time settings are not correct, the VPN gateways will reject the certificate during the loading process. The CA server uses the Date/Time setting of the host machine.

In the Cisco router:
2621#clock set 13:28:00 4 February 2003

VPN3000 Time and Date Settings (Figure 10)
In the VPN3000:
Configuration
System
General

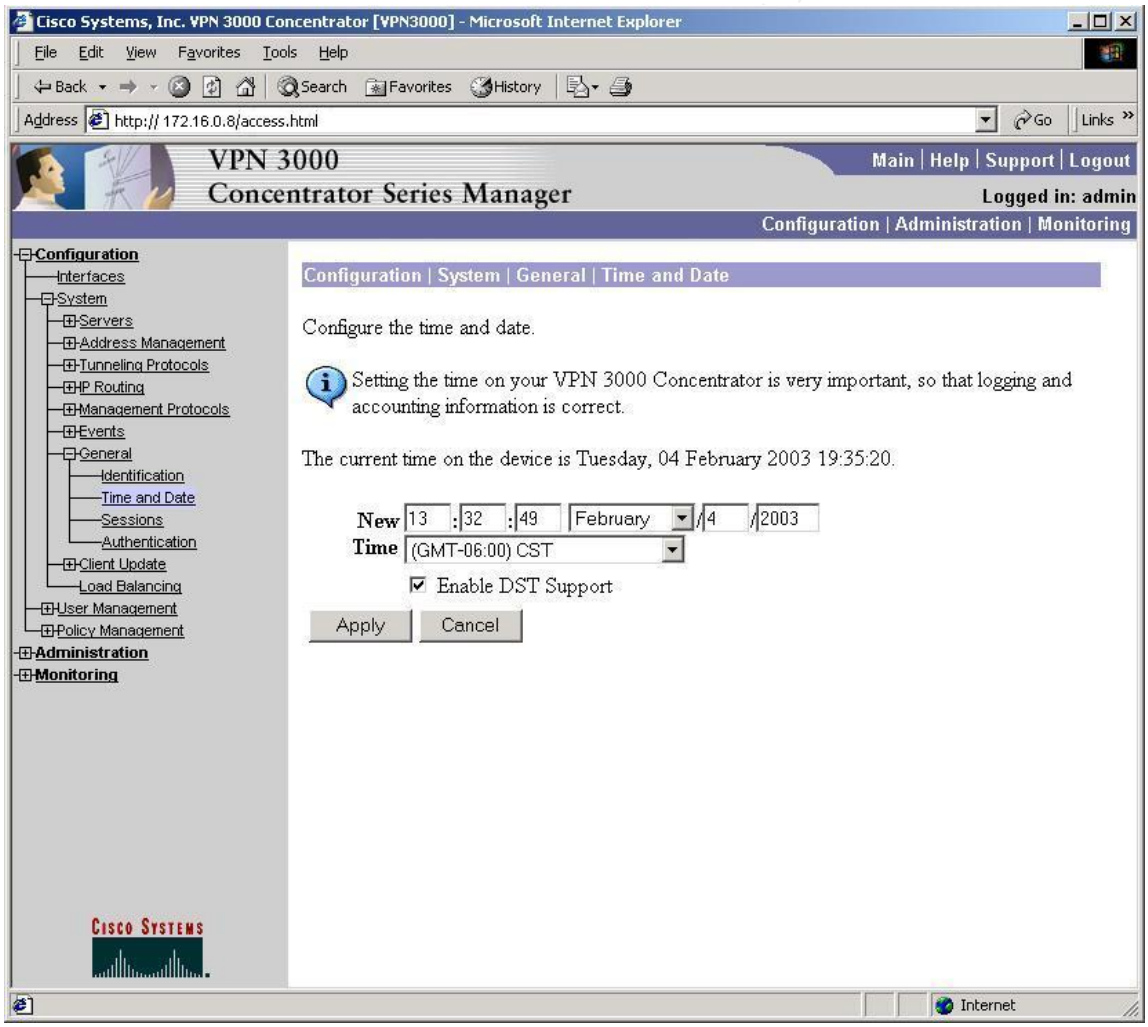

Figure 10:VPN 3000 Time and Date Settings

Loading Certificates on the 2600:

In order to support digital certificate authentication, the router must be configured to recognize a certificate authority. For this example, we are using a trust chain with a single self signed CA. We will also ignore CRl checking. The Cisco Trustpoint commands define the CA and obtain the root certificate of the CA. The Crypto CA enroll command causes the router to generate an identity certificate request and submit it to the CA, and then retrieve it from the CA using the SCEP protocol.

```
#config t
Enter configuration commands, one per line.  End with CNTL/Z.

Define the certificate Authority

2621(config)#crypto ca trustpoint NEPTUNE
2621(ca-trustpoint)#enrollment url
http://172.16.0.3:446/8bae1b88222ca2ee3dc5ad9008f8075c343aff30
2621(ca-trustpoint)#enrollment retry period 60
2621(ca-trustpoint)#enrollment retry count 100
2621(ca-trustpoint)#crl op
2621(ca-trustpoint)#crl optional
2621(ca-trustpoint)#exit
Obtain the CA certificate:

2621(config)#crypto ca authenticate NEPTUNE
Certificate has the following attributes:
Fingerprint: 40E3154D 7EB9E837 0F6F4AE4 0566E3D7
% Do you accept this certificate? [yes/no]: yes
1w3d: CRYPTO_PKI: Sending CA Certificate Request:
GET
/8bae1b88222ca2ee3dc5ad9008f8075c343aff30/pkiclient.exe?operation=GetCAC
ert&message=NEPTUNE HTTP/1.0

1w3d: CRYPTO_PKI: can not resolve server name/IP address
1w3d: CRYPTO_PKI: Using unresolved IP Address 172.16.0.3
1w3d: CRYPTO_PKI: http connection opened
1w3d: CRYPTO_PKI: HTTP response header:
 HTTP/1.1 200 OK

Date: Mon, 03 Feb 2003 20:44:27 GMT
Server: Apache/1.3.19-rev2 (Win32) mod_ssl/2.8.1 SSL-C
Connection: close
Content-Type: application/x-x509-ca-cert
Content-Type indicates we have received a CA certificate.

1w3d: Received 513 bytes from server as CA certificate:
1w3d: CRYPTO_PKI: transaction GetCACert completed
1w3d: CRYPTO_PKI: CA certificate received.
1w3d: CRYPTO_PKI: CA certificate received.
1w3d: CRYPTO_PKI: crypto_pki_authenticate_tp_cert()
1w3d: CRYPTO_PKI: trustpoint NEPTUNE authentication status = 2
Trustpoint CA certificate accepted.

Next, a key pair must be generated for the router:
```

```
2621(config)#crypto key generate rsa
 Choose the size of the key modulus in the range of 360 to 2048 for your
 General Purpose Keys. Choosing a key modulus greater than 512 may take
 a few minutes.

How many bits in the modulus [512]:
2w0d: %SSH-5-DISABLED: SSH 1.5 has been disabled1024
% Generating 1024 bit RSA keys ...[OK]

2w0d: %SSH-5-ENABLED: SSH 1.5 has been enabled
2621#show crypto key pubkey-chain rsa
Codes: M - Manually configured, C - Extracted from certificate

Code Usage   IP-Address       Name
C    Signing                 X.500 DN name:
                             CN = Neptune
                             OU = IPSECLAB
                             O = PQ
                             C = US

C    General                 VPN3000w.com

2621#show crypto key mypubkey rsa
% Key pair was generated at: 20:30:15 UTC Feb 3 2003
Key name: 2621.IPSECLAB.com
 Usage: General Purpose Key
 Key Data:
  30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181
00AD04C2
  0D493529 085A0EFC 5C3EF7AF 051A7AD7 33FD0BF4 52BAE786 CE4A17DB
D495A173
  E71E1ECC 5865C6A5 4C66DAD5 2C985966 C6ED3AE5 CD04C3C4 D363AAAB
6D986596
  C3282E35 A48EDED1 3437F2A1 178C1B36 F52089DF F10EF167 D0D93F7A
A05BD2CE
  5451B4B3 01B9F11A 8160BCBE F569F554 D56B2E6E 5DA18C66 FDAD7D89
0D020301 0001
% Key pair was generated at: 20:30:26 UTC Feb 3 2003
Key name: 2621.IPSECLAB.com.server
 Usage: Encryption Key
 Key Data:
  307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00A0F67D
E44299FB
  F3584DEA 0A929CF6 0B49A51A 4108C0EE C959E351 549E3573 6EAEF20B
0B821A5F
  14026A23 F724C375 8BC62F37 320ED020 FE0BBB11 795C8C30 2717E8B5
C3F83580
  3893AF4B ED1F6814 49AFD151 431A057F 72F4E53A 82E2EE69 37020301 0001
2621#
```

Now the request can be generated for an identity certificate using SCEP to send the request to the CA and retrieve the ID cert.

```
2621(config)#crypto ca enroll NEPTUNE
%
```

```
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the
   configuration. Please make a note of it.


Password:
Re-enter password:

% The subject name in the certificate will be: 2621.IPSECLAB.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: 686DB299
% Include an IP address in the subject name? [no]: yes
Enter Interface name or IP Address[]: fastethernet0/1
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the
fingerprint.
2621(config)#    Fingerprint:  E634B08F 32B6B2DF 603BEE58 72BA62DD

1w3d: CRYPTO_PKI: Sending CA Certificate Request: GET
/8bae1b88222ca2ee3dc5ad9008f8075c343aff30/pkiclient.exe?operation=GetCAC
ert&message=NEPTUNE HTTP/1.0

1w3d: CRYPTO_PKI: can not resolve server name/IP address
1w3d: CRYPTO_PKI: Using unresolved IP Address 172.16.0.3
1w3d: CRYPTO_PKI: http connection opened
1w3d: CRYPTO_PKI: HTTP response header:
 HTTP/1.1 200 OK

Date: Mon, 03 Feb 2003 20:45:09 GMT
Server: Apache/1.3.19-rev2 (Win32) mod_ssl/2.8.1 SSL-C
Connection: close
Content-Type: application/x-x509-ca-cert
Content-Type indicates we have received a CA certificate.

1w3d: Received 513 bytes from server as CA certificate:
1w3d: CRYPTO_PKI: transaction PKCSReq completed
1w3d: CRYPTO_PKI: status:
1w3d: CRYPTO_PKI: can not resolve server name/IP address
1w3d: CRYPTO_PKI: Using unresolved IP Address 172.16.0.3
1w3d: CRYPTO_PKI: http connection opened
1w3d: CRYPTO_PKI:  received msg of 2712 bytes
1w3d: CRYPTO_PKI: HTTP response header:
 HTTP/1.1 200 OK

Date: Mon, 03 Feb 2003 20:45:11 GMT
Server: Apache/1.3.19-rev2 (Win32) mod_ssl/2.8.1 SSL-C
Connection: close
Content-Type: application/x-pki-message
1w3d: CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL
1w3d: CRYPTO_PKI: status = 100: certificate is granted
1w3d: CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL
```

```
1w3d: CRYPTO_PKI: All enrollment requests completed.
1w3d: CRYPTO_PKI: All enrollment requests completed.
1w3d: %CRYPTO-6-CERTRET: Certificate received from Certificate Authority
1w3d: CRYPTO_PKI: All enrollment requests completed.
1w3d: CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL
2621(config)#
1w3d: CRYPTO_PKI: All enrollment requests completed.
2621(config)#
2621(config)#exit
2621#
1w3d: %SYS-5-CONFIG_I: Configured from console by console
2621#

Verify the certificates in the router

2621#show crypto ca certificates

The routers CA cert:
Certificate
  Status: Available
  Certificate Serial Number: 9E66B7212CFC2DC980212F24BF2B9CDC
  Certificate Usage: General Purpose
  Issuer:
    CN = Neptune
     OU = IPSECLAB
     O = PQ
     C = US
  Subject:
    Name: 2621.IPSECLAB.com
    IP Address: 172.16.0.6
    Serial Number: 686DB299
    OID.1.2.840.113549.1.9.2 =<16> 2621.IPSECLAB.com +
OID.1.2.840.113549.1.9.8 = 172.16.0.6 + OID.2.5.4.5 = 686DB299
  CRL Distribution Point:
    http://neptune:447/Neptune%20CA.crl
  Validity Date:
    start date: 20:45:12 UTC Feb 3 2003
    end   date: 20:45:12 UTC Feb 3 2004
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoint: NEPTUNE

The root certificate of the CA

CA Certificate
  Status: Available
  Certificate Serial Number: C120FE2B658A927F2DA72FBC86019E8D
  Certificate Usage: General Purpose
  Issuer:
    CN = Neptune
     OU = IPSECLAB
     O = PQ
     C = US
  Subject:
    CN = Neptune
     OU = IPSECLAB
```

```
     O = PQ
     C = US
  Validity Date:
    start date: 16:59:40 UTC Dec 17 2002
    end   date: 21:59:40 UTC Dec 17 2005
  Associated Trustpoint: NEPTUNE
2621#
```

Loading Certificates on the VPN 3000:

The first step in configuring the VPN3000 to use digital certificates is to load the root certificate of the CA. Since our network provides direct LAN connectivity to the CA server, we will use SCEP. For cases where the CA server is off-line we could also use the cut and paste method. The information below gives the VPN3000 the enrollment URL and "nickname" of the CA server. Note that we must append "pkiclient.exe" on the URL. The KEON CA server uses port 446 for SCEP.

VPN3000 CA Enrollment: (Figure 11)
Administration
Certificate Management
Installation
Install CA certificate
SCEP (Simple Certificate Enrollment Protocol)

URL: http://172.16.0.3:446/8bae1b88222ca2ee3dc5ad9008f8075c343aff30/pkiclient.exe

CA Descriptor: NEPTUNE

Retrieve

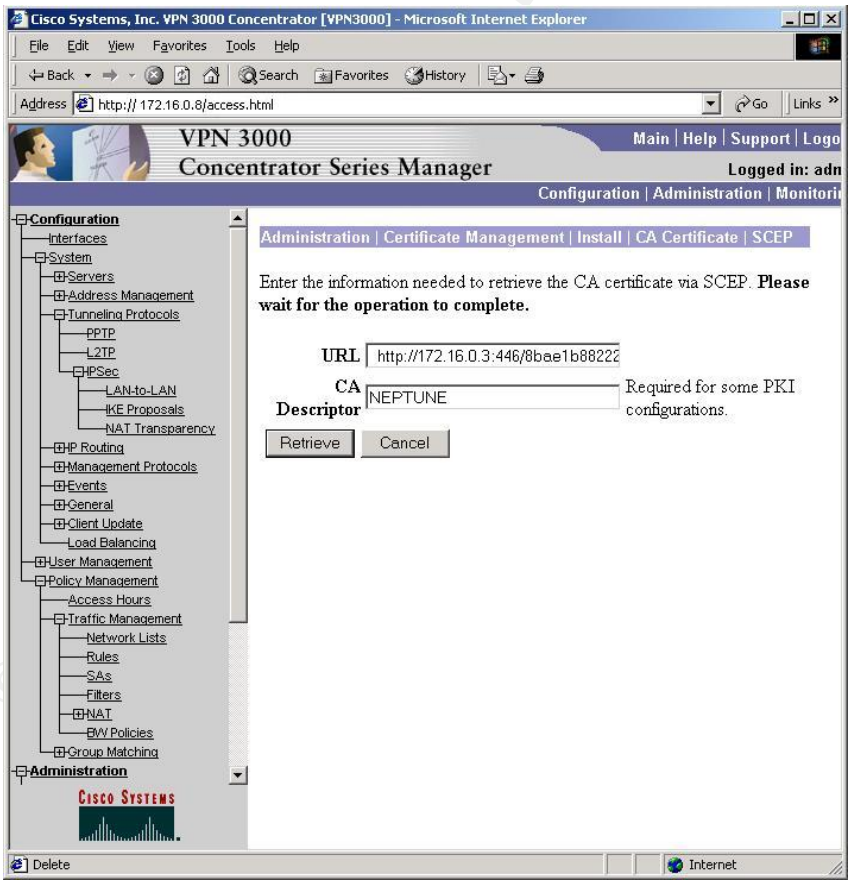

Figure 11: CA Enrollment

The VPN3000 will obtain the root certificate of the CA.

To obtain a Self-Certificate from the KEON

VPN 3000 Self-Certificate parameters: (Figure 12)
Administration
Certificate Management
Enrollment
Identity certificate
Enroll via SCEP at NEPTUNE
Enroll

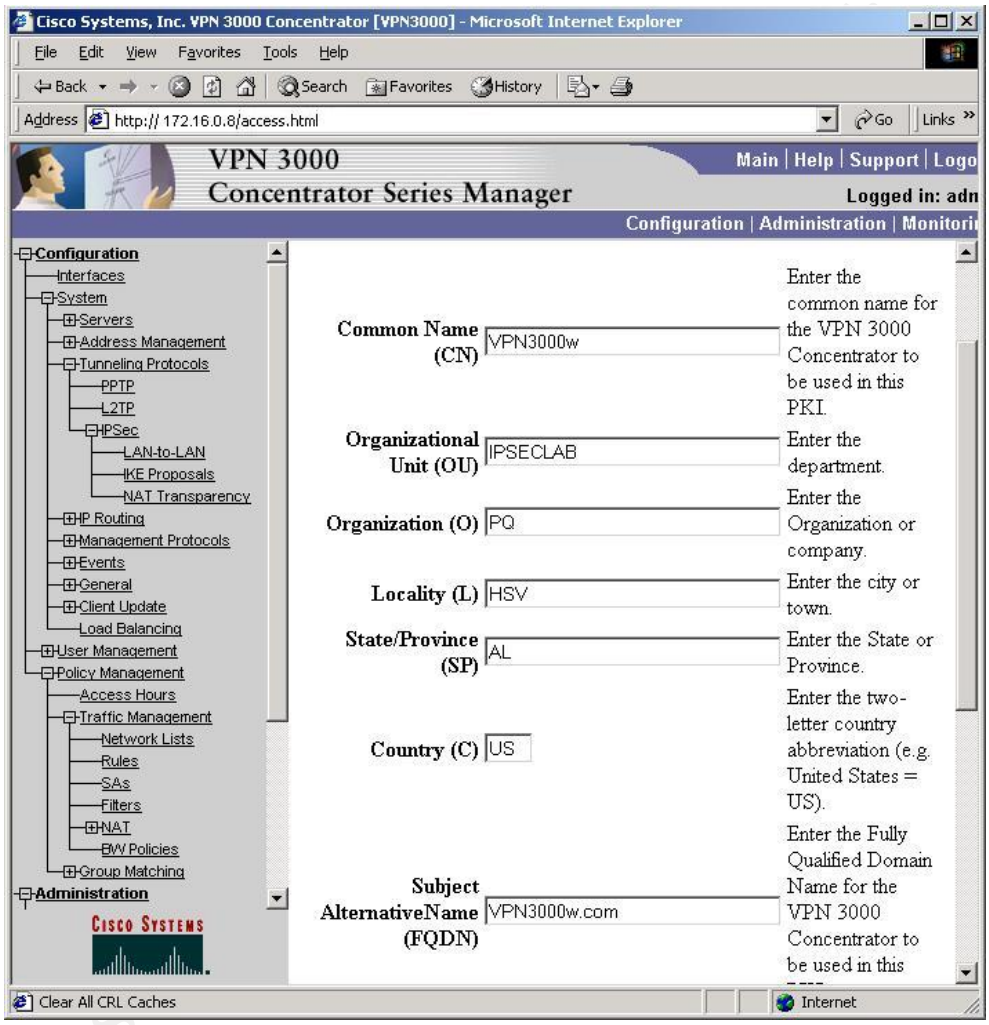

Figure 12: Self-Certificate Parameters

Fill in the identity certificate Information: Common Name=VPN3000w, organizational unit, organization, etc. For this example, I am using a key size of 1024 bits.

Once the VPN 3000 has retrieved its Identity Certificate, we can verify the CA certificate and the Identity certificate by examining the Certificate Management page.

VPN 3000 Certificate Management (Figure 13)

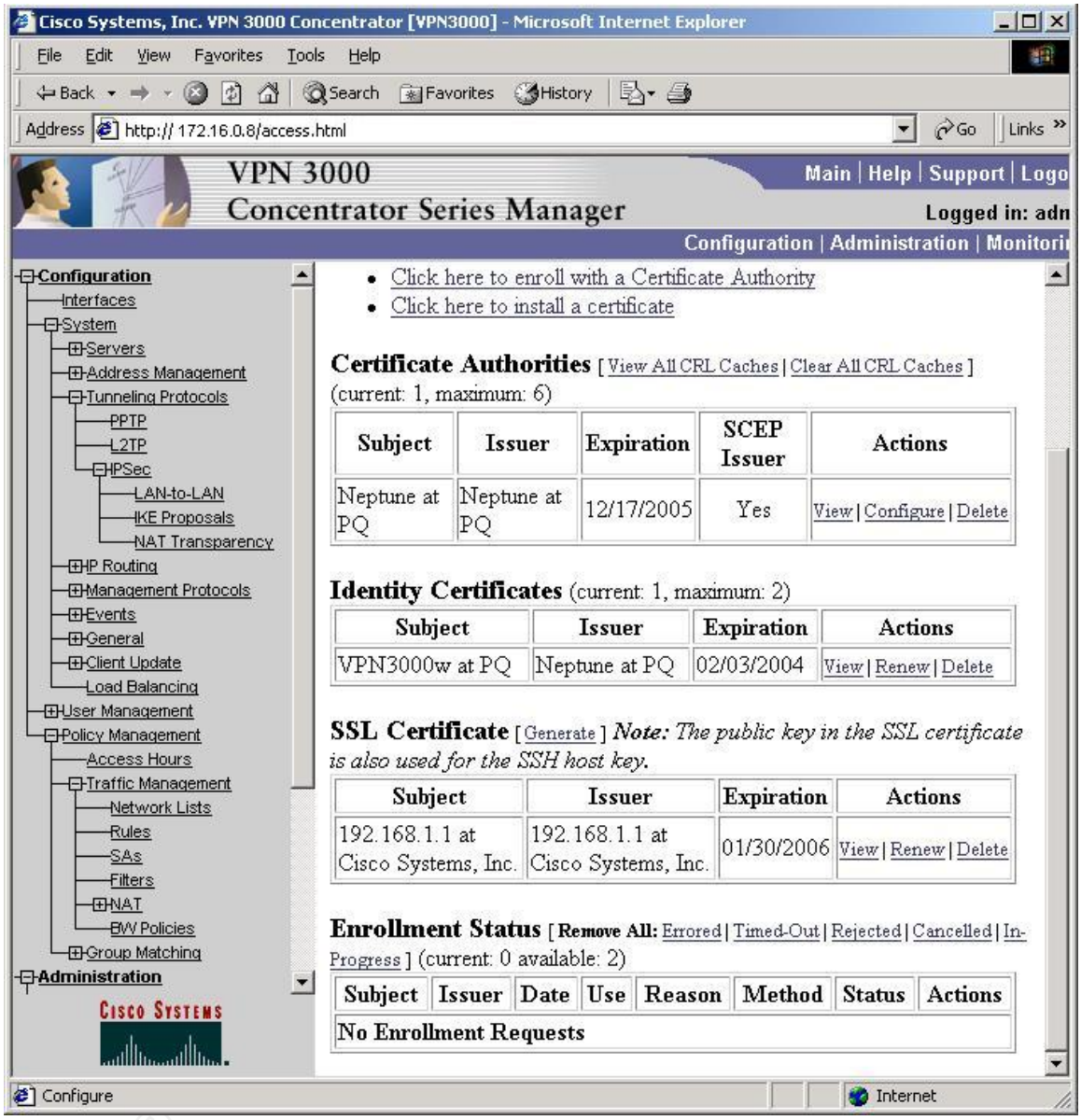

Figure 13: VPN 3000 Certificates

To use the certificates to authenticate our VPN tunnel, we must make the appropriate changes in the VPN3000 IKE proposal and in the LAN-to-LAN menu

In the IKE proposal, specify:

Authentication Mode: RSA Digital Certificate

VPN3000 IKE Proposals: (Figure 14)

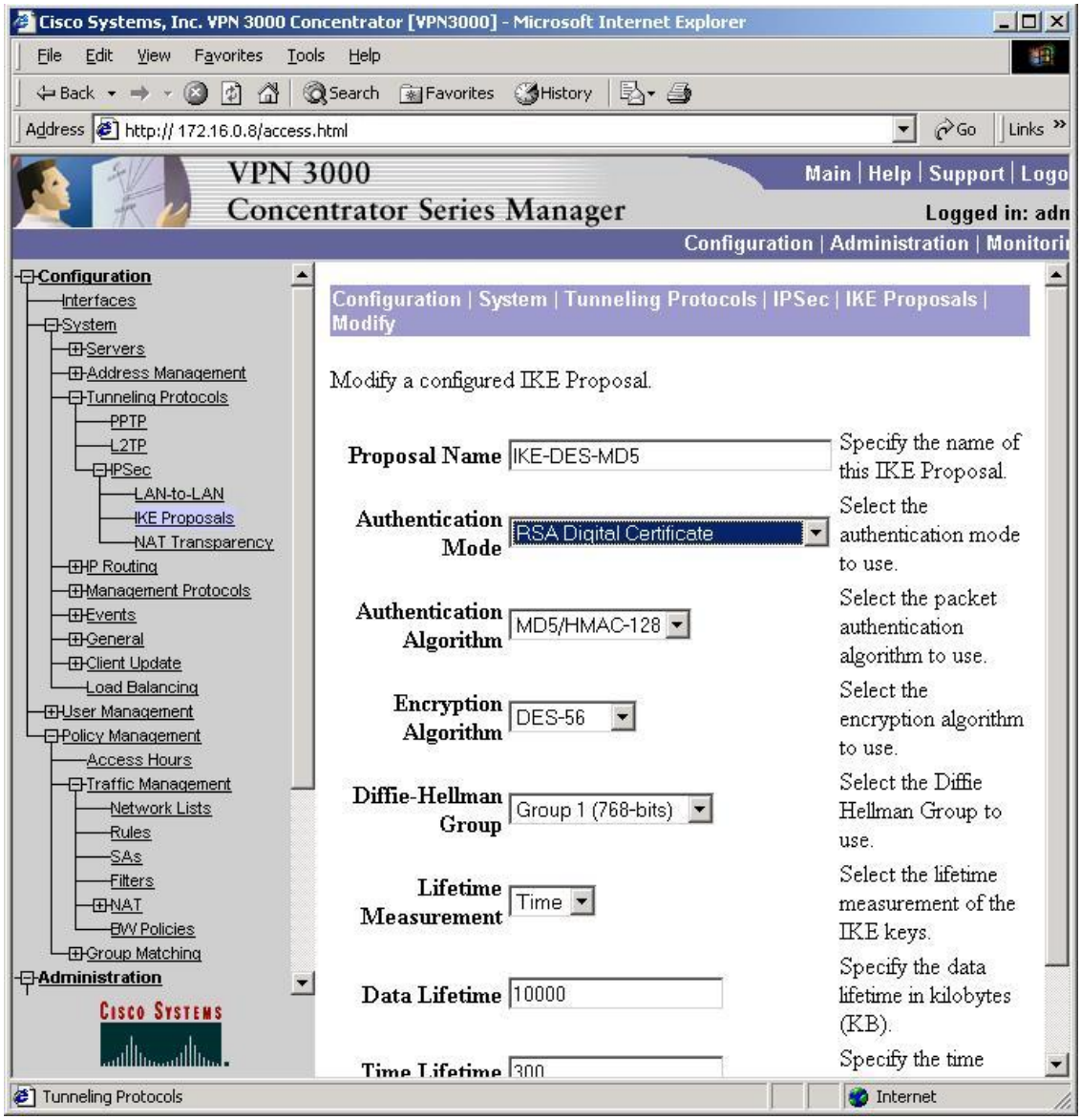

Figure 14: Authentication Mode

VPN3000 LAN-to-LAN / Certificate Selection: (Figure 15)

In the Lan-to-Lan menu, specify:

Digital Certificate: VPN3000w

Certificate Transmission : Entire certificate Chain

clear the pre-shared key entry.

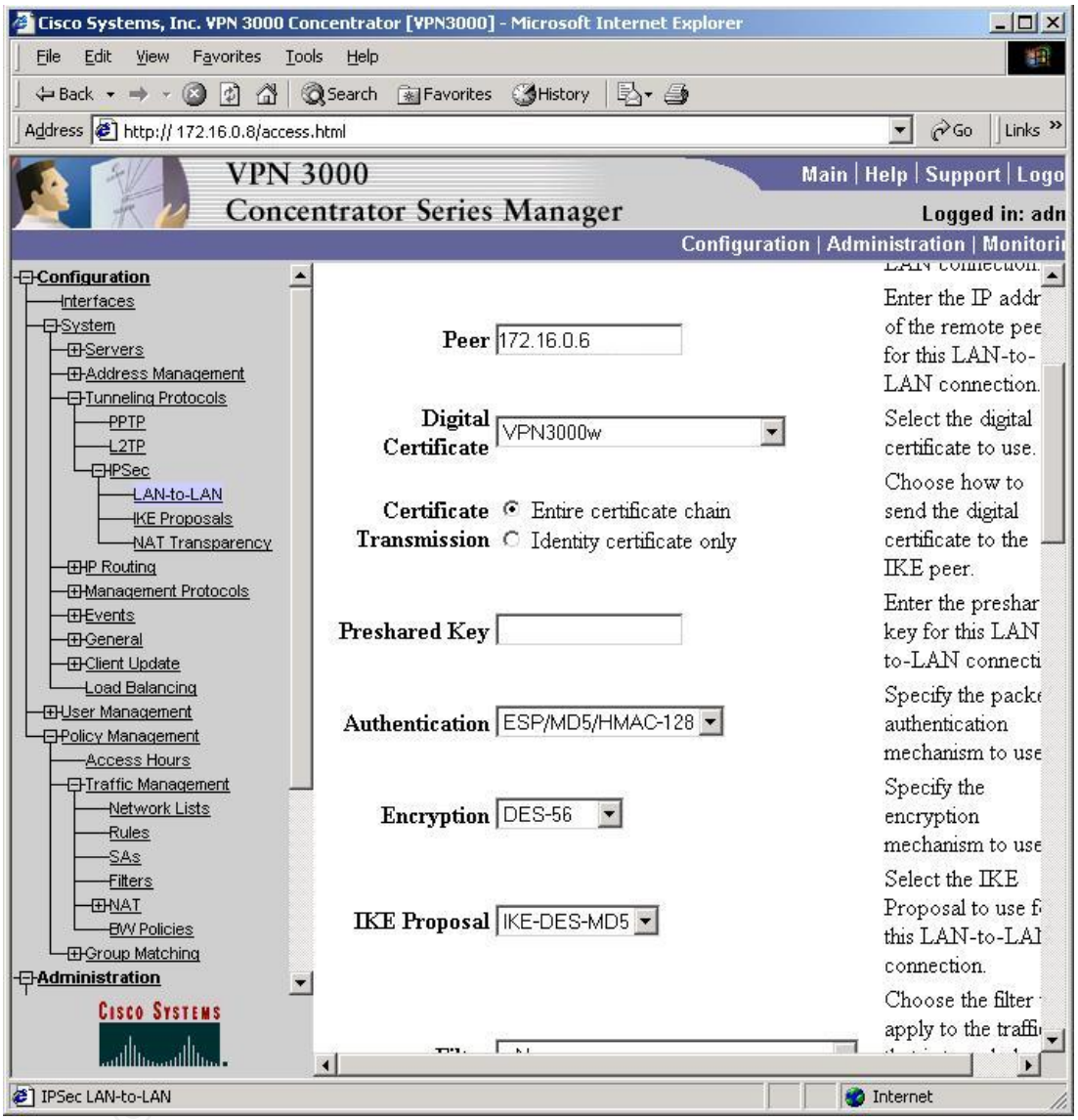

Figure 15: VPN 3000 Certificate Selection

Certificate Contents:
It is frequently a good idea to open certificates and examine the contents with a cert reader to verify that the certificates contain correct information. Shown below is the output produced by reading the Identity certificates of the VPN3000 and the 2621 with a cert reader called Ana_cert. Note the presence of the public keys belonging to the clients, subject information, and digital signature of the CA using the SHA signing algorithm.

Output produced for the 2621 router Self-Certificate:

PKIX Message Analysis Tool
Jul 27 1999
Input filename: A:\routercert.txt

VERSION: 3
SERIAL: 0x009e66b7 212cfc2d c980212f 24bf2b9c dc
SIGNATURE ALGORITHM: SHA1 with RSA Encryption
ISSUER:
Country Name: US
Organization Name: PQ
Organizational Unit Name: IPSECLAB
Common Name: Neptune
VALIDITY:
Not Before: Feb 3, 03 20:45:12 GMT
Not After: Feb 3, 04 20:45:12 GMT
SUBJECT:
Serial Number: 686DB299
Ext Cert Attributes: 172.16.0.6
Unstructured Name: 2621.IPSECLAB.com
PUBLIC KEY:
RSA Encryption: 1024 bits.
MODULUS: 0xf8e8c2cb aa139386 0dda4b31 65406a58
8126d0e7 02f0580a f4d0410a 11d94bfc
1ba2a3e3 35395419 c12bee77 98dd4422
46e2a342 23b81de6 36e46848 fcc9c494
59ddca73 2606d67c a695633a 63b450b3
aa75e6ea e72e2550 bb5a2831 9fd34858
27bbfb19 a70b19d7 14bff6fa cf7b890d
3fef9152 dd4e2edb 41ffc2ae f1cac687
EXPONENT: 0x010001
EXTENSIONS:
Subject Alt Name:
IP Address: 0x3f6c8106
DNS Name: 2621.IPSECLAB.com
CRL Distribution Points:
Distribution point 1:

Uniform Resource ID: http://neptune:447/Neptune%20CA.crl
OUTER SIGNATURE ALGORITHM: SHA1 with RSA Encryption

SIGNATURE: 0xaba90a4f 042f9d2e 5165022a 5de7895a
d820a307 8f3e158b a55f652a 26aa7bc5
c6ecf9dd afae5eac d63eb8e3 817d9e9a
49c98a28 6056010e ef8cefe6 4d6292a3
ec45118a 9c851cc3 8bac64ea b7f2ff35
5f572370 48c9af1c 4aa24c25 58834598
8110c100 83a04833 870a49aa 38a03bc7
e309afc4 c05b3918 1fa41e5f e8144b2e

Output produced for the VPN3000 Self Certificate:

PKIX Message Analysis Tool
Jul 27 1999
Input filename: A:\VPN3000cert.txt

VERSION: 3
SERIAL: 0x23418a70 bf948abc 52c92235 5d8d09c5
SIGNATURE ALGORITHM: SHA1 with RSA Encryption
ISSUER:
Country Name: US
Organization Name: PQ
Organizational Unit Name: IPSECLAB
Common Name: Neptune
VALIDITY:
Not Before: Feb 3, 03 20:34:02 GMT
Not After: Feb 3, 04 20:34:02 GMT
SUBJECT:
Common Name: VPN3000w
Organizational Unit Name: IPSECLAB
Organization Name: PQ
Locality Name: HSV
State Or Province: AL
Country Name: US
PUBLIC KEY:
RSA Encryption: 1024 bits.
MODULUS: 0x86316bb4 efce5418 8a425b5c 28b86a51
9a1f7b75 fbc520eb edbde42f dc8bafbf
b9ac3f75 193a47ba 40068b00 66d1cfdb
beba1dd0 8cff550a cd7b9e2b 10c4d4e7
8c4b46dd d1361ca4 270a72d1 7198fca5
d86e23c3 de9e792c 2073f85e 084bb55b
2fadbde1 8ef3c9ed 23903a74 2e2e6b68

```
            7d701c10 07f94e53 42e16f24 b18317a5
    EXPONENT:    0x07
  EXTENSIONS:
    Subject Alt Name:
      DNS Name:  VPN3000w.com
    CRL Distribution Points:
      Distribution point 1:
        Uniform Resource ID:  http://neptune:447/Neptune%20CA.crl
  OUTER SIGNATURE ALGORITHM:  SHA1 with RSA Encryption
  SIGNATURE:    0x474aab12 46ae7f5b aff65f6b 41db53e8
                f2c1bf73 34da2bd2 d1f8aadd 62db935b
                78942d03 78519c67 110203ac 98fa3931
                71402c2b be1f009c 52f8ef4a 7901babe
                478e11f0 d3a8a6cc a30115f0 6f0817e2
                a586a926 30716125 ab17fcd8 24ac874d
                f1e348d0 e9162975 fc343095 9fd28f0a
                71a4171e a6fc3e88 87cda8fc b875b90f
```

Final Configuration Steps:
Once the certificates are loaded, the authentication RSA-SIG command will change the IKE policy to use the certificates from CA NEPTUNE for VPN authentication.

```
2621#
2621#config t
Enter configuration commands, one per line.  End with CNTL/Z.
2621(config)#crypto isakmp policy 100
2621(config-isakmp)#authentication rsa-sig
2621(config-isakmp)#exit
2621(config)#exit
2621#
1w3d: %SYS-5-CONFIG_I: Configured from console by consol
2621#
2621#wri
Building configuration...
[OK]
```

Now, a ping from Host A to Host B will bring up a VPN Tunnel between the two gateways. The debug output produced by the router provides details of the progress of the phase 1 and phase 2 transactions and the successful creation of the SAs:

```
2621#
1w3d: IPSEC(sa_request): ,
  (key eng. msg.) OUTBOUND local= 172.16.0.6, remote= 172.16.0.8,
    local_proxy= 10.10.10.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-md5-hmac ,
    lifedur= 300s and 4608000kb,
    spi= 0xA778E9E9(2809719273), conn_id= 0, keysize= 0, flags= 0x400D
```

```
1w3d: ISAKMP: received ke message (1/1)
1w3d: ISAKMP: local port 500, remote port 500
1w3d: ISAKMP (0:1): Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
Old State = IKE_READY  New State = IKE_I_MM1

1w3d: ISAKMP (0:1): beginning Main Mode exchange
1w3d: ISAKMP (0:1): sending packet to 172.16.0.8 (I) MM_NO_STATE
1w3d: ISAKMP (0:1): received packet from 172.16.0.8 (I) MM_NO_STATE
1w3d: ISAKMP (0:1): Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Old State = IKE_I_MM1  New State = IKE_I_MM2

1w3d: ISAKMP (0:1): processing SA payload. message ID = 0
1w3d: ISAKMP (0:1): Checking ISAKMP transform 1 against priority 100
policy
1w3d: ISAKMP:      encryption DES-CBC
1w3d: ISAKMP:      hash MD5
1w3d: ISAKMP:      default group 1
1w3d: ISAKMP:      auth RSA sig
1w3d: ISAKMP:      life type in seconds
1w3d: ISAKMP:      life duration (basic) of 300
1w3d: ISAKMP (0:1): atts are acceptable. Next payload is 0
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): vendor ID seems Unity/DPD but bad major
1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Old State = IKE_I_MM2  New State = IKE_I_MM2

1w3d: ISAKMP (0:1): sending packet to 172.16.0.8 (I) MM_SA_SETUP
1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Old State = IKE_I_MM2  New State = IKE_I_MM3

1w3d: ISAKMP (0:1): received packet from 172.16.0.8 (I) MM_SA_SETUP
1w3d: ISAKMP (0:1): Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Old State = IKE_I_MM3  New State = IKE_I_MM4

1w3d: ISAKMP (0:1): processing KE payload. message ID = 0
1w3d: ISAKMP (0:1): processing NONCE payload. message ID = 0
1w3d: ISAKMP (0:1): SKEYID state generated
1w3d: ISAKMP (0:1): processing CERT_REQ payload. message ID = 0
1w3d: ISAKMP (0:1): peer wants a CT_X509_SIGNATURE cert
1w3d: ISAKMP (0:1): peer want cert issued by CN = Neptune, OU =
IPSECLAB, O = PQ, C = US
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): vendor ID is Unity
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): vendor ID seems Unity/DPD but bad major
1w3d: ISAKMP (0:1): vendor ID is XAUTH
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): speaking to another IOS box!
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): vendor ID seems Unity/DPD but bad major
1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Old State = IKE_I_MM4  New State = IKE_I_MM4

1w3d: ISAKMP (0:1): SA is doing RSA signature authentication using id
type ID_IPV4_ADDR
1w3d: ISAKMP (1): ID payload
      next-payload : 6
```

```
      type         : 1
      protocol     : 17
      port         : 500
      length       : 8
1w3d: ISAKMP (1): Total payload length: 12
1w3d: ISAKMP (0:1): sending packet to 172.16.0.8 (I) MM_KEY_EXCH
1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Old State = IKE_I_MM4  New State = IKE_I_MM5

1w3d: ISAKMP (0:1): received packet from 172.16.0.8 (I) MM_KEY_EXCH
1w3d: ISAKMP (0:1): Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Old State = IKE_I_MM5  New State = IKE_I_MM6
1w3d: ISAKMP (0:1): processing ID payload. message ID = 0
1w3d: ISAKMP (0:1): processing CERT payload. message ID = 0
1w3d: ISAKMP (0:1): processing a CT_X509_SIGNATURE cert
1w3d: ISAKMP (0:1): processing CERT payload. message ID = 0
1w3d: ISAKMP (0:1): processing a CT_X509_SIGNATURE cert
1w3d: CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL

1w3d: CRYPTO_PKI: cert revocation status unknown.
1w3d: ISAKMP: cert approved with warning
1w3d: CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL

1w3d: CRYPTO_PKI: cert revocation status unknown.
1w3d: ISAKMP: cert approved with warning
1w3d: ISAKMP (0:1): OU = IPSECLAB
1w3d: ISAKMP (0:1): processing SIG payload. message ID = 0
1w3d: ISAKMP:received payload type 14
1w3d: ISAKMP (0:1): processing vendor id payload
1w3d: ISAKMP (0:1): vendor ID is DPD
1w3d: ISAKMP (0:1): SA has been authenticated with 172.16.0.8
1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Old State = IKE_I_MM6  New State = IKE_I_MM6

1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Old State = IKE_I_MM6  New State = IKE_P1_COMPLETE

1w3d: ISAKMP (0:1): beginning Quick Mode exchange, M-ID of -1118353828
1w3d: ISAKMP (0:1): sending packet to 172.16.0.8 (I) QM_IDLE
1w3d: ISAKMP (0:1): Node -1118353828, Input = IKE_MESG_INTERNAL,
IKE_INIT_QM
Old State = IKE_QM_READY  New State = IKE_QM_I_QM1

1w3d: ISAKMP (0:1): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE

1w3d: ISAKMP (0:1): received packet from 172.16.0.8 (I) QM_IDLE
1w3d: ISAKMP (0:1): processing HASH payload. message ID = -1118353828
1w3d: ISAKMP (0:1): processing SA payload. message ID = -1118353828
1w3d: ISAKMP (0:1): Checking IPSec proposal 1
1w3d: ISAKMP: transform 1, ESP_DES
1w3d: ISAKMP:   attributes in transform:
1w3d: ISAKMP:      SA life type in seconds
1w3d: ISAKMP:      SA life duration (basic) of 300
1w3d: ISAKMP:      SA life type in kilobytes
```

```
1w3d: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
1w3d: ISAKMP:      encaps is 1
1w3d: ISAKMP:      authenticator is HMAC-MD5
1w3d: ISAKMP:      group is 1
1w3d: ISAKMP (0:1): atts are acceptable.
1w3d: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.16.0.6, remote= 172.16.0.8,
    local_proxy= 10.10.10.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-md5-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x14
1w3d: ISAKMP (0:1): processing NONCE payload. message ID = -1118353828
1w3d: ISAKMP (0:1): processing KE payload. message ID = -1118353828
1w3d: ISAKMP (0:1): processing ID payload. message ID = -1118353828
1w3d: ISAKMP (0:1): processing ID payload. message ID = -1118353828
1w3d: ISAKMP (0:1): Creating IPSec SAs
1w3d:         inbound SA from 172.16.0.8 to 172.16.0.6
        (proxy 192.168.1.0 to 10.10.10.0)
1w3d:         has spi 0xA778E9E9 and conn_id 420 and flags 15
1w3d:         lifetime of 300 seconds
1w3d:         lifetime of 4608000 kilobytes
1w3d:         outbound SA from 172.16.0.6    to 172.16.0.8    (proxy
10.10.10.0      to 192.168.1.0    )
1w3d:         has spi 12808073 and conn_id 421 and flags 1D
1w3d:         lifetime of 300 seconds
1w3d:         lifetime of 4608000 kilobytes
1w3d: IPSEC(key_engine): got a queue event...
1w3d: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 172.16.0.6, remote= 172.16.0.8,
    local_proxy= 10.10.10.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-md5-hmac ,
    lifedur= 300s and 4608000kb,
    spi= 0xA778E9E9(2809719273), conn_id= 420, keysize= 0, flags= 0x15
1w3d: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 172.16.0.6, remote= 172.16.0.8,
    local_proxy= 10.10.10.0/255.255.255.0/0/0 (type=4),
    remote_proxy= 192.168.1.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= esp-des esp-md5-hmac ,
    lifedur= 300s and 4608000kb,
    spi= 0xC36F89(12808073), conn_id= 421, keysize= 0, flags= 0x1D
1w3d: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.16.0.6, sa_prot= 50,
    sa_spi= 0xA778E9E9(2809719273),
    sa_trans= esp-des esp-md5-hmac , sa_conn_id= 420
1w3d: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.16.0.8, sa_prot= 50,
    sa_spi= 0xC36F89(12808073),
    sa_trans= esp-des esp-md5-hmac , sa_conn_id= 421
1w3d: ISAKMP (0:1): sending packet to 172.16.0.8 (I) QM_IDLE
1w3d: ISAKMP (0:1): deleting node -1118353828 error FALSE reason ""
1w3d: ISAKMP (0:1): Node -1118353828, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
Old State = IKE_QM_I_QM1  New State = IKE_QM_PHASE2_COMPLETE
```

Conclusion:

Digital certificates are a sophisticated and effective method for providing authentication for a variety of applications. Digital certificate authentication employs multiple security mechanisms that help ensure integrity of network communication. As is frequently the case in the technical world, a lot of things have to be exactly right for it to work. Technological sophistication often brings with it complexity, implementation difficulties and costs which have at times made the future of PKI seem unclear. The standards are fairly well established, but some vendors are still working out problems in their implementations. In spite of the drawbacks, use of certificates is prevalent in network applications and many enterprises have found deployment of a digital certificate system to be a necessity for support of growing networks and remote access.

Endnotes:

1. Sampo Savolainen, "Internet Key Exchange," Helsinki University of Technology. Online. Internet. November 22, 1999. Available http://www.niksula.cs.hut.fi/~sjsavola/SoN/essay.html

2. D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)." The Network Working Group. Online. Internet. November 1998. Available http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2408.html

3. William Stallings, Cryptography and Network Security: Principals and Practice, Second Edition (Upper Saddle River, New Jersey: Prentice Hall, 1999) 341.

4. James S. Tiller, A Technical Guide to IPSec Virtual Private Networks (Washington, D.C.: Auberbach Publications, 2001) 67.

5. Netscape Communications Corporation, "Understanding certificates," Netscape Certificate Server Administrator's Guide for Unix. Online. Internet. April 22, 2002. Available http://developer.netscape.com/docs/manuals/certificate/certagux/overview.htm

6. Ibid.

7. William Stallings, Cryptography and Network Security: Principals and Practice, Second Edition (Upper Saddle River, New Jersey: Prentice Hall, 1999) 343.

8. Ibid. 342.

9. Xiaoyi Lui, Cheryl Madson, David McGrew, and Andrew Nourse, "Cisco Systems Simple Certificate Enrollment Protocol (SCEP)," The Network Working Group. Online. Internet. May 15,2002. Available http://www.ietf.org/internet-drafts/draft-nourse-scep-06.txt

10. Ibid.

11. Ibid.

12. Andrew Nash, PKI: Implementing and Managing E-Security (Berkeley, California: RSA Press McGraw Hill, 2001) 201.

13. Xiaoyi Lui, Cheryl Madson, David McGrew, and Andrew Nourse, "Cisco Systems Simple Certificate Enrollment Protocol (SCEP)," The Network Working Group. Online. Internet. May 15,2002. Available http://www.ietf.org/internet-drafts/draft-nourse-scep-06.txt

14. Andrew Nash, PKI: Implementing and Managing E-Security (Berkeley, California: RSA Press McGraw Hill, 2001) 107.

15. Ibid. 451.

16. Ibid. 51.

17. E. Gerck and MCG, "Overview of Certification Systems: X.509, CA, PGP and SKIP," MCG Internet Open Group on Certification and Security. Online. Internet. August 1, 1998. Available http://www.mcg.org.br/cert.htm

18. D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)." The Network Working Group. Online. Internet. November 1998. Available http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2408.html

19. James S. Tiller, A Technical Guide to IPSec Virtual Private Networks (Washington, D.C.: Auberbach Publications, 2001) 184.

20. D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)." The Network Working Group. Online. Internet. November 1998. Available http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2408.html

21. James S. Tiller, A Technical Guide to IPSec Virtual Private Networks (Washington, D.C.: Auberbach Publications, 2001) 180.

22. Ibid. 182.

23. Ibid. 181.

24. Ibid. 185.

Bibliography:

Gerck E. and MCG. "Overview of Certification Systems: X.509, CA, PGP and SKIP," MCG Internet Open Group on Certification and Security. Online. Internet. August 1, 1998. Available http://www.mcg.org.br/cert.htm

Harkins, D. and D. Carrel. "The Internet Key Exchange (IKE)," The Network Working Group. Online. Internet. November 1998. Available http://www.ietf.org/rfc/rfc2409.txt

Liu, Xiaoyi, Cheryl Madson, David McGrew, and Andrew Nourse. "Cisco Systems Simple Certificate Enrollment Protocol (SCEP)," The Network Working Group. Online. Internet. May 15,2002. Available http://www.ietf.org/internet-drafts/draft-nourse-scep-06.txt

Maughan, D., M. Schertler, M. Schneider, and J. Turner. "Internet Security Association and Key Management Protocol (ISAKMP)." The Network Working Group. Online. Internet. November 1998. Available http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2408.html

Nash, Andrew. PKI: Implementing and Managing E-Security. Berkeley, California: RSA Press McGraw Hill, 2001.

Netscape Communications Corporation. "Understanding certificates." Netscape Certificate Server Administrator's Guide for Unix. Online. Internet. April 22, 2002. Available http://developer.netscape.com/docs/manuals/certificate/certagux/overview.htm

Savolainen, Sampo. "Internet Key Exchange." Helsinki University of Technology. Online. Internet. November 22, 1999. Available http://www.niksula.cs.hut.fi/~sjsavola/SoN/essay.html

Stallings, William. Cryptography and Network Security: Principals and Practice. Second Edition. Upper Saddle River, New Jersey: Prentice Hall, 1999.

Tiller, James S. A Technical Guide to IPSec Virtual Private Networks. Washington, D.C.: Auberbach Publications, 2001.