

Chapter

3

Encryption and VPNs

VPN-1 enables you to create site-to-site Virtual Private Networks (VPNs) that provide secure communication between two defined participants, by encrypting the communication on unsecured public networks, such as the Internet.

Objectives:

- Explain encryption for VPNs.
- Compare and contrast common encryption methods.
- Describe the process for setting up a encrypted VPN tunnels.

Key Terms:

- Diffie-Hellman (DH)
- Certificate Authority (CA)
- Security Association (SA)
- Aggressive mode
- Main mode
- Certificate Revocation List
- Internal Certificate Authority (ICA)

COPY

Securing Communication

When information is sent over a public network, such as the Internet, a message passes through many computers, routers, switches, and similar equipment, before arriving at a destination. Along the way are many opportunities to intercept the message. The original message can be altered or a false message sent. The false message appears to have come from a trusted sender, but does not.

Security Administrators need to ensure:

Confidentiality — No one, other than the intended parties, can understand the communication.

Integrity — The sensitive data passed between the communicating parties is unchanged, and this can be proved with an integrity check.

Authentication — The communicating parties must be sure they are connecting with the intended party.

VPN-1 protects communication on the Internet, and enables an enterprise to build its own easy-to-maintain VPN, using private- and public-network segments. Supported are industry-standard algorithms and protocols, such as DES, 3DES, and IPSec/IKE. Public Key Infrastructure (PKI) support enables the use of digital Certificates.

Privacy

Encryption is the transformation of readable data, or cleartext, into an unreadable form, called ciphertext. You use encryption with VPN-1 through a Virtual Private Network (VPN). A VPN provides secured connections between points where encrypted data may travel through unsecured networks.

Encryption works by encrypting data with encryption software and a secret key that is known only to the sender and recipient. This shared-secret key is used to decrypt the encrypted packet. This figure is an example of encryption:

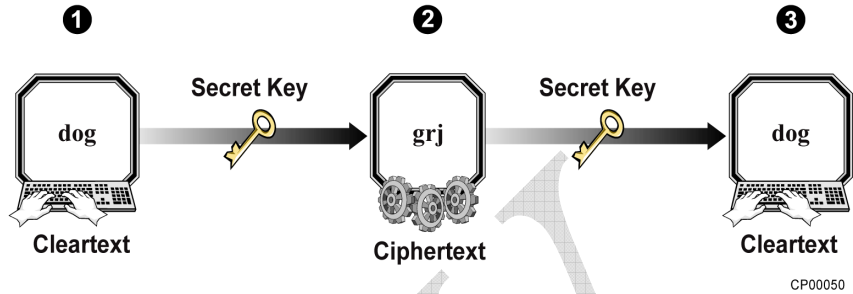


Figure 3-21: Shared-Secret Key

VPN-1 supports the following encryption technologies:

- Symmetric and asymmetric encryption
- Diffie-Hellman key management
- Digital signatures

Symmetric Encryption

Symmetric encryption, in which the same key is used to encrypt and decrypt data, is also called shared-key encryption. Symmetric encryption is primarily used for faster encryption performance.

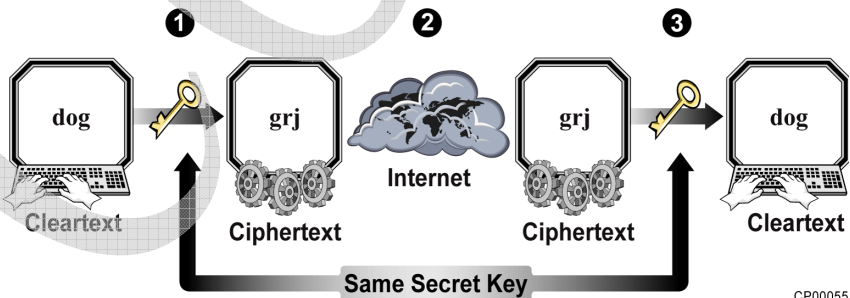


Figure 3-22: Symmetric Encryption

1. The cleartext message is encrypted, using the shared key.
2. The encrypted packet passes through the unsecured network.
3. On the Gateway (on the other side), the same shared key is used to decrypt the ciphertext.

Symmetric Disadvantages

Because symmetric encryption uses one key for both encryption and decryption, you should consider the following issues before implementation:

- Since the secret key is used for both encryption and decryption, anyone who steals the key can then steal all of the data that is currently, or had already been, encrypted.
- Because of this danger, the keys must be delivered in a protected manner, such as a direct face-to-face negotiation, or a telephone call.

This method of transmitting and replacing keys may be acceptable for a small number of keys. But as the number of keys increases, key management becomes impossible.



Do not send shared keys through unsecured networks, such as the Internet. Use sneakernet (hand delivery), mail, fax, or telephone to send the private key to the intended recipient.

Symmetric secret-key encryption is simple and fast, but:

- The number of keys required can quickly become unmanageable, since there must be a different key pair for each pair of possible correspondents. For example, the number of keys to be managed for 10,000 hosts is about 50 million.

Public-key encryption systems solve these problems. With public-key encryption systems, a key pair is composed of two mathematically related keys — a public key known to everyone, and a private key known only to its owner. A message encrypted with one of the keys in a key pair can only be decrypted with the other key in a pair.

Asymmetric Encryption

Asymmetric encryption, which uses one key to encrypt a message and another to decrypt the message, is an encryption technology used for the following:

- Secure-key exchange mechanisms
- Authentication
- Data-integrity checking

Asymmetric encryption is also called public/private-key encryption, because the encryption scheme uses two keys, one private and one public. These keys are created using the Diffie-Hellman encryption scheme, where one Gateway's public key and another Gateway's private key create a shared-secret key. This shared-secret key is used to verify and decrypt the encrypted packet.

Because different keys are used for encryption and decryption, they are called asymmetric keys.

Diffie-Hellman

A **Diffie-Hellman** public/private-key pair is used for calculating a secret key, which is used for encrypting and decrypting messages. No secret information is communicated during the key exchange, so it does not require a secure channel. Only one key pair needs to be managed for each correspondent.

This figure illustrates the Diffie-Hellman encryption scheme:

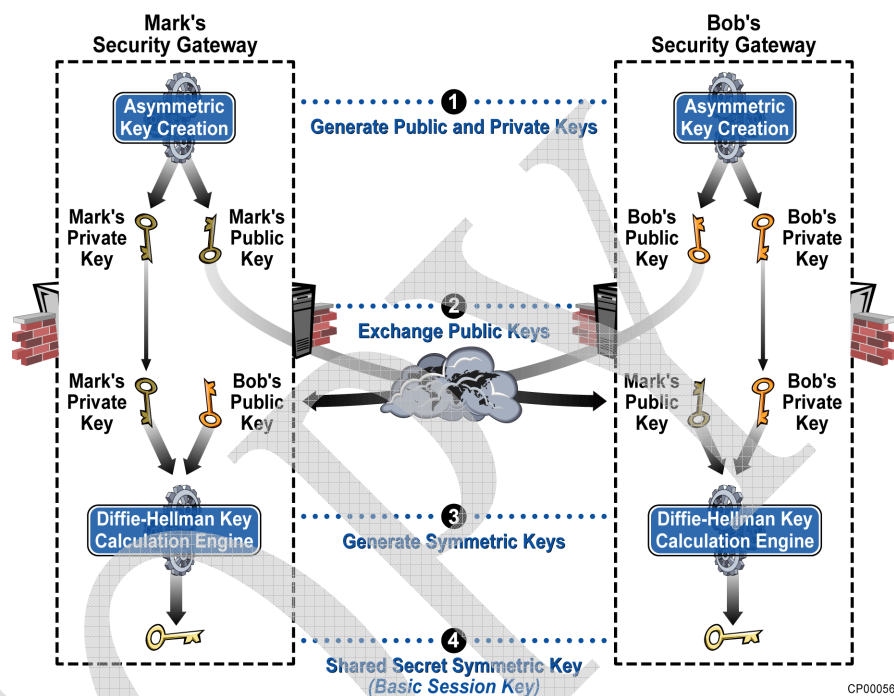


Figure 3-23: Diffie-Hellman Encryption

1. Mark and Bob exchange public keys.
2. Using Diffie-Hellman, Mark combines his private key with Bob's public key to generate the shared-secret key.
3. Since the two parties exchanging their respective public keys are the only parties that can generate the shared secret, a two-way trust model is established, where no other party can create the shared secret.

Due to performance issues, asymmetric cryptography is 1,000 times slower than symmetric cryptography. Asymmetric cryptography is typically used to encrypt small amounts of data, such as keys, for symmetric cryptography.

The Diffie-Hellman scheme is generally considered secure when an appropriate mathematical group is used. The standard Diffie-Hellman protocol includes three predefined groups. Longer groups provide better security, but their computation requires more CPU resources. VPN-1 allows you to select the most suitable mathematical group, and you can also expand the database of mathematical groups by adding customized groups.

Integrity

In addition to keys ensuring a message's integrity, a hash of the message can be computed. A hash function is a one-way mathematical function that maps variable values into smaller values of a fixed length. The size of the message is made smaller, to ensure maximum network performance. The shorter the message, the less computation required, and the better the performance. The result of the hash function, known as the message digest, is much smaller than the original message, but unique to it. If any changes to a message occur, the message digest will be different, indicating the message has been altered.

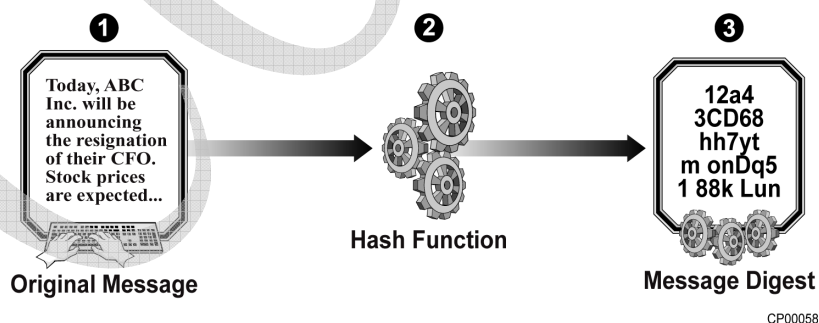


Figure 3-24: Hash Function

1. The original message is created.
2. The original message is processed using a hash function.
3. The hash function creates a new, and much smaller, representation of the original message.

The hash is:

- Simple and fast
- Unique and irreversible.



In practical terms, it is impossible to derive an original message from a resulting message digest, because the only known method would take millions of years — hence the term “one-way hash function”.

The benefits of using a one-way hash function are:

- Easy to use and irreversible
- Encrypted with a sender’s private RSA key, a message digest becomes a digital signature.
- Can be used for authenticating data integrity

Authentication

To verify that a message actually comes from a specific sender and not an imposter, a digital signature is attached to the message. A digital signature acts as proof of the sender’s identity and message’s integrity.

A digital signature is a code that can be attached to an electronically transmitted message. The digital signature then uniquely identifies the sender, and verifies that the message has not been tampered with in transit. Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message is really who he claims to be. Digital signatures use public-key cryptography. To be effective, digital signatures must be unforgeable. Digital signatures are provided by a **Certificate Authority**, where a third party verifies key authenticity.

This figure shows where a digital signature is attached during message transmission:

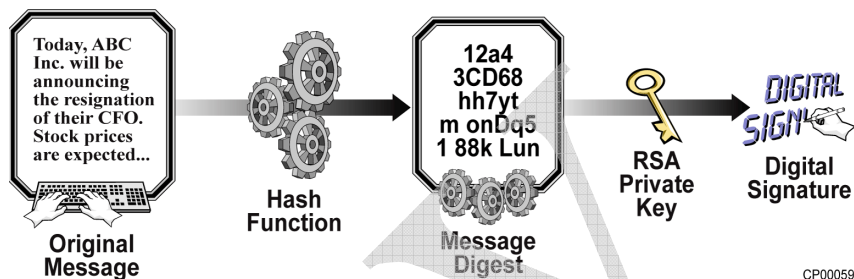


Figure 3-25: Digital Signature

VPN-1 digital signatures use RSA encryption and hash functions.

RSA is the public-key cryptosystem that VPN-1 uses to create and verify digital signatures. In contrast to Diffie-Hellman, RSA key pairs are used for signing and verifying Certificates. Any information encrypted with an RSA public key can only be decrypted with the matching RSA private key, and vice versa. This establishes a one-way trust model.

In the figure above, the hash value confirms the message's integrity, and is also the previously agreed-upon text of the digital signature. It is possible to use some other agreed-upon text. But the hash value is convenient, because it is different for each message, and content does not actually need to be agreed on in advance.



Digital signatures are especially important for e-commerce, and are key components of most authentication schemes.

Two Phases of Encryption

Public-key encryption requires much more computing effort than private-key encryption, so it is much slower. In practice, encrypted communication sessions are usually divided into two phases:

- A preliminary, short key negotiation that is the exchange phase, secured by the slower public-key encryption; a private key is negotiated or exchanged for encrypting the actual message. Internet Key Exchange (IKE) is an example of a commonly used key-exchange method.
- The main message-encryption phase, in which a message is encrypted using the faster private key that was negotiated in the first phase; Data Encryption Standard (DES) and CAST are examples of commonly used encryption algorithms.

VPN-1 supports the IKE encryption scheme. All encryption schemes consist of:

- Key-management protocol — for generating and exchanging keys
- Encryption algorithm — for encrypting messages
- Authentication algorithm — for ensuring integrity

The following table describes elements of IKE:

| Key-Management Protocol | Encryption Algorithm | Authentication Algorithm | Encryption is ... |
|--|--|--------------------------|--|
| IKE: Industry-standard protocol for VPN key management | DES (Triple DES for key encryption), CAST, AES | HMAC-MD5 HMAC-SHA-1 | Encapsulated; traffic encryption is IPSec. |

Table 3-26: Elements of IKE

Encryption Algorithms

The following are the encryption algorithms that can be used by VPN-1 to determine how to encrypt data:

DES — Short for Data Encryption Standard, DES is a symmetric-key encryption method that uses a 56-bit key. DES allows interoperability with other IKE-compliant firewalls, and provides one standard for encryption.

Triple DES — Addresses the security concerns resulting from the relatively short 56-bit key used for DES; Triple DES encrypts under three different DES keys in succession, equivalent to doubling the DES key length to 168 bits.

AES — The Advanced Encryption Standard (AES) is the new Federal Information Processing Standard (FIPS) publication that specifies a cryptographic algorithm for use by U.S. government organizations to protect sensitive (unclassified) information. AES will be widely used on a voluntary basis by organizations, institutions, and individuals outside the U.S. government, and outside the United States in some cases. Rijndael is the AES algorithm. A key length of between 128 and 256 bits is supported. The more bits added, the stronger the encryption.

CAST cipher — CAST cipher is similar to DES. While the VPN-1 implementation of CAST uses a 40-bit key length, the CAST algorithm supports variable key lengths, anywhere from 40 to 128 bits. CAST has a 64-bit block size, which is the same as DES. Even though CAST has been found to be twice as fast as a typical implementation of DES, and six times faster than a Triple DES implementation, CAST is not as strong as DES, using comparable key lengths.

IKE

ISAKMP

The Internet Security Association and Key Management Protocol (ISAKMP), is the encryption standard of the Internet Engineering Task Force (IETF). The ISAKMP protocol provides a consistent framework for transferring keys and authenticating data, independent of the encryption and authentication mechanisms.

Oakley

Oakley is the protocol used to establish strong cryptography-based keys used for encrypting data.

- Oakley defines how users select the prime-number groups for the Diffie-Hellman key exchange.
- Keys can be derived from Diffie-Hellman keys, or from an existing encryption key.
- Oakley allows IPSec to use secret-key and Certificate-based authentication.

ISAKMP/Oakley

ISAKMP/Oakley, also known as IKE, allows VPN servers and clients to share encrypted-key information. Computers must agree on how to exchange and secure information, by forming an SA (Security Association) before that exchange can take place. The computers need to agree on how to encrypt and decrypt the data being sent.

To do this, Security Administrators use a combination of the ISAKMP protocol and the Oakley key-generation protocol. The ISAKMP protocol is the centralized manager of the SA between the server and client, while the Oakley protocol generates and manages the encryption keys used to secure the information.

The ISAKMP/Oakley process is done in two phases. Each phase uses the encryption and authentication agreed on by two computers, during the initial negotiations.

Phase 1

ISAKMP SA Negotiation

In Phase 1, peers negotiate a **Security Association** (SA) that will be used for encrypting and authenticating Phase 2 exchanges. Phase 1 involves long and CPU-intensive computations, so is executed infrequently. A cookie-exchange mechanism precedes the computations, to prevent denial-of-service attacks. The negotiated SA includes the encryption method, authentication method, and keys. This SA is then used in the Phase 2 negotiation. The first phase is used to protect the identity of the two machines to be connected.

The first step in Phase 1 is to negotiate the four parameters of the SA:

- The encryption algorithm: DES, 3DES, AES, CAST
- The hash algorithm: MD5 or SHA1
- The authentication method
- The Diffie-Hellman group

VPN-1 supports two modes for Phase 1:

- **Aggressive mode** (the default), in which three packets are exchanged
- **Main mode**, in which six packets are exchanged

IKE has additional modes for dealing with remote access:

- **Hybrid mode** — provides an alternative to IKE Phase 1, where the Gateway is allowed to authenticate using Certificates and the client via some other means, such as SecurID

- **Office mode** — is an extension to the IKE protocol; Office mode is used to resolve routing issues between remote-access clients and the VPN-1 Domain. During the IKE negotiation, a special mode called config mode is inserted between Phases 1 and 2. During config mode, the remote-access client requests an IP address from the Gateway. After the Gateway assigns the IP address, the client creates a virtual adapter in the operating system. The virtual adapter uses the assigned IP address.

Phase 2

IPSec SA Negotiation

In Phase 2, the Security Association (SA) negotiated in Phase 1 is used by the peers to negotiate an SA for encrypting the IPSec traffic. Keys can be modified as often as required during a connection's lifetime, by performing Phase 2. Phase 2 provides additional protection, by refreshing the keys to ensure the reliability of the SAs, and to prevent a man-in-the-middle attack.

The first step in Phase 2 is the policy negotiation, exchanging:

- The IPSec protocol: ESP, AH, ESP+HA
- The hash algorithm: MD5, SHA1



When two computers are in agreement, two SAs are established — one for inbound communication and one for outbound communication.

- Oakley refreshes the key material, and new shared or secret keys are generated for authentication and encryption.
- The SAs and keys are passed to the IPSec driver, and communication is established.

IKE Example

1. The Oslo HR Server in Net-Oslo contacts the Rome HR Server in Net-Rome, asking to transfer personnel records.
2. The Oslo VPN-1 checks to see if the packets should be encrypted.
3. IPSec is being used, so ISAKMP/Oakley begins the negotiations.
4. The ISAKMP/Oakley service on the Rome VPN-1 receives a request for secure negotiations.
5. The two Gateways establish an ISAKMP SA, and exchange the Diffie-Hellman public-key information (Phase 1).
6. With identities confirmed, the IPSec SAs are established, and Oakley generates or refreshes the session keys (Phase 2).
7. The Oslo Gateway uses the outbound SA to sign and encrypt the HR data packets.
8. The data packets are routed to Net-Rome.
9. The Rome Gateway receives the packets, and uses the inbound SA to check the integrity signature and decrypt the packets.
10. The decrypted packets are then passed to the Rome HR Server.

Tunneling-Mode Encryption

IKE uses tunneling-mode encryption, which works by encapsulating an entire packet, and then adding its own encryption protocol header to the encrypted packet, as shown in this figure:

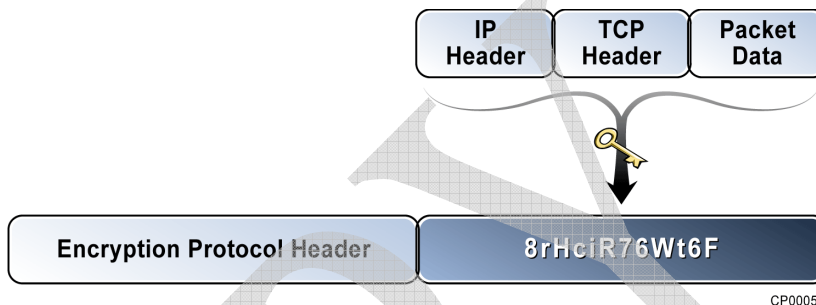


Figure 3-27: Encrypted Packet

You can envision tunneling-mode encryption as follows:

1. You write a message (data) and place it in an envelope.
2. You address the outside of the envelope with the destination address and the return address (header).
3. You then place the addressed envelope (packet) into another envelope containing a different destination and return address (encryption-protocol header).
4. The envelope is then mailed.

A drawback to using tunneling-mode encryption is that packet size is increased. This increase in packet size degrades network performance; however, the security of the packet is increased.

VPN-1 uses tunneling-mode encrypting, IKE:

| Feature | IKE |
|---|---|
| Portability | Standard |
| Key Management | Automatic, external PKI |
| Session Keys | Keys change at configured times during connection's live time. |
| Number of keys is proportional to the ... | Square of the number of correspondents (with pre-shared secret) Number of correspondents (with digital Certificates) |
| Packet Size | Increased |
| PFS (Perfect Forward Secrecy) | Yes |
| Replay Protection | Yes |
| Certificate Authority | Specified in the workstation properties |
| Supported Schemes | All |
| Packet Headers | Encrypts original IP and TCP headers |
| Encryption Mode | Tunneling adds new IP (first) header and IPSec (second) header in a packet. |
| VPN Capabilities | Can be used in VPNs that use reserved IP addresses, without needing address translation or proxying |

Table 3-28: IKE Features

Certificate Authorities

A Certificate Authority (CA) issues Certificates to entities (users or hosts), used to identify and provide verifiable information about themselves. A Certificate might include a Distinguished Name (DN), public key, and IP address. After two entities exchange and validate each other's Certificates, they can encrypt communication between them, using the public keys contained in the Certificates.

Two kinds of entities can identify themselves using Certificates:

- Encrypting Gateways, when encrypting with other encrypting Gateways, or with VPN-1 SecureClient
- VPN-1 SecureClient, and the site confirming each others' identities with Certificates

Certificates are used in Gateway-to-Gateway encryption, when both gateways have public-key signatures enabled in their IKE properties. Certificates are used in client-to-site encryption, when a VPN-1 Gateway has public-key signatures enabled (in its IKE properties), and a user has public key enabled in the IKE properties (of its user properties).

Certificates

For VPN-1 to use Certificates:

1. Determine which Certificate Authorities (CAs) to use, contact them, and load any specific software they require.
2. Define the CA to VPN-1.
3. Generate the certificates, using the steps that apply to the specific type of CA.

VPN-1 IKE implementation supports X.509 digital Certificates provided by these Public Key Infrastructure (PKI) implementations:

- Check Point VPN-1 Certificate Manager
- OPSEC PKI vendors

- Entrust Technologies
- Internal CA on a Check Point SmartCenter Server

Multiple Certificate Authorities

For small networks, a single Certificate Authority (CA) may be all that is needed. Larger enterprise networks that need to authenticate and encrypt their communication with different branches, vendors, and customers that use different CAs may need to use multiple CAs.

Enterprise-network VPNs must be able to:

1. Acquire and recognize different Certificates, such as Entrust and OPSEC PKI.
2. Trust more than one CA.
3. Acquire more than one Certificate for an entity.



An entity can only have one Certificate from each CA.

Certificate Authority Hierarchy

In a Certificate Authority hierarchy, where a CA's Certificate is issued by another CA, only the highest-level trusted CA needs to be defined. Certificates issued by a CA subordinate to a trusted CA are also trusted.

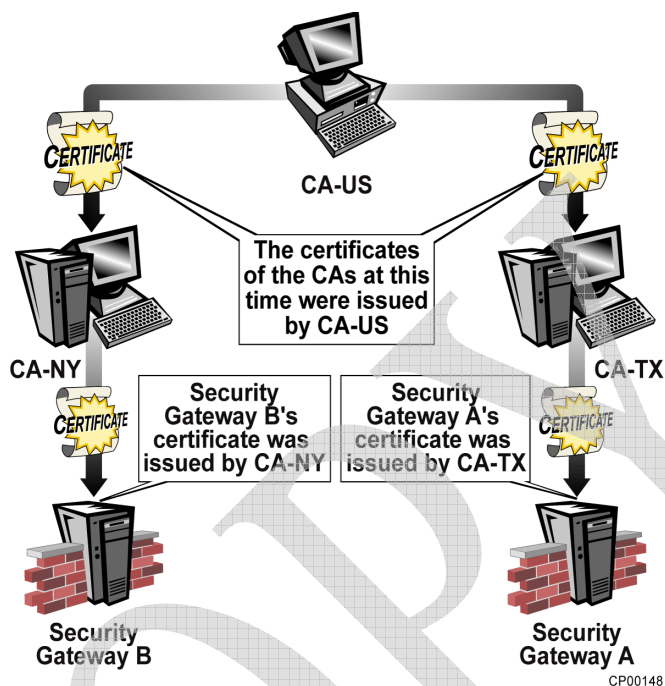


Figure 3-29: CA Hierarchy

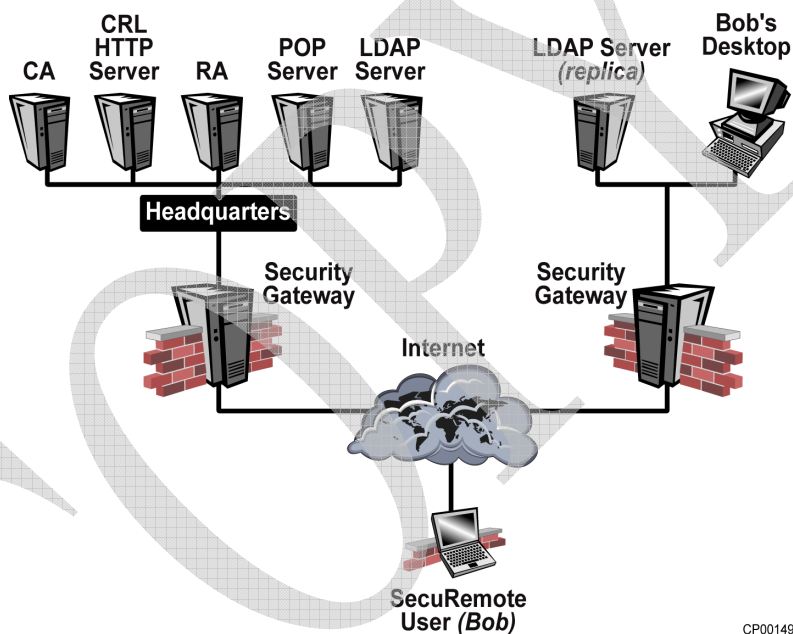
In this figure, Gateway A and Gateway B trust CA-US, and the CA-NY and CA-TX Certificates were both issued by CA-US. Therefore, Gateway A and Gateway B will also trust both CA-NY and Ca-TX. If Gateway A has a Certificate issued by CA-TX, and Gateway B has a Certificate issued by CA-NY, Gateway A and Gateway B will accept each others' Certificates.

VPN-1 SecureClient CAs with Users

When a VPN-1 SecureClient user and a site authenticate using a Certificate, VPN-1 SecureClient trusts only the CA that signed the user's Certificate. If the site's Certificate is signed by a different CA and the Gateway's CA is in the same hierarchy as the user's CA, the authentication will fail.

Local Certificate Authority

In this configuration, the Certificate Authority and **Certificate Revocation List (CRL)** repository are local servers managed by the Security Administrator in Headquarters. VPN-1 SecuRemote users do not have access to the LDAP servers and cannot download CRLs. The VPN-1 SmartCenter Server manages keys and Certificates for the VPN-1 Gateways, involving interaction between the VPN-1 SmartCenter Server and the CA.



CP00149

Figure 3-30: Local Certificate Authority

In the figure, the two Gateways create a VPN between them, using Certificates to authenticate one another. VPN-1 SecuRemote user Bob generates a key pair on his own, contacting the CA to receive his Certificate. He then uses his key and Certificate to establish a VPN between his remote PC, and any of the offices. The VPN-1 SecuRemote

software mandates that the CRL be sent to it. IKE negotiation will fail, if a valid CRL is not sent to it as part of the negotiation.

CA Service via the Internet

In this configuration, the Certificate Authority and CRL HTTP server are accessed through the Internet.

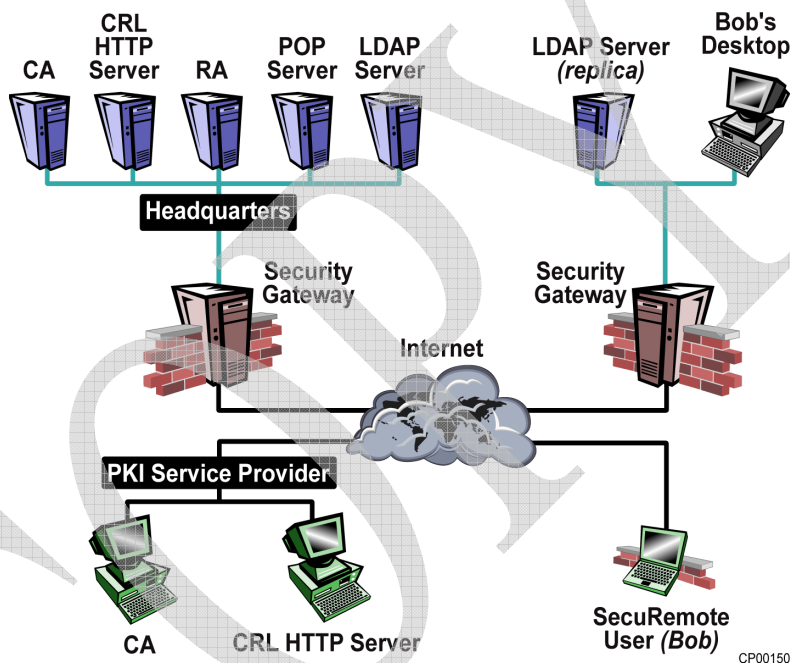


Figure 3-31: CS Services via Internet

The figure shows a PKI where the Certificate Authority and the CRL repository are active servers, accessible through the Internet. The VPN-1 SmartCenter Server manages keys and Certificates for the Gateways, with the certification process itself involving interaction between the VPN-1 SmartCenter Server and the CA.

In the figure above, the two Gateways create a VPN between them, using Certificates to authenticate one another. VPN-1 SecuRemote user Bob generates a key pair on his own, contacting the CA directly and receiving a Certificate. Bob then can use his key and Certificate to establish a VPN between his remote PC, and any of the offices.

Internal Certificate Authority

The **Internal Certificate Authority (ICA)** is a fully featured, internal authentication server that is installed on a Check Point SmartCenter Server. The ICA allows Security Administrators to configure a complete security solution with VPN-1, without the need for third-party software. The ICA can be used in the following situations:

- Establishing Secure Internal Communications between Check Point components, including Open Platform for Security (OPSEC) applications
- Providing certificates for users and Security Administrators
- Authenticating VPN-1 SecuRemote and VPN-1 SecureClient traffic to Gateways for VPN capabilities
- Using Hybrid Mode RAS VPN for authenticating Gateways to VPN-1 SecuRemote or VPN-1 SecureClient users
- Establishing site-to-site VPNs between Gateways

Created during installation, the ICA is responsible for overseeing the generation, signing, and revocation of Certificates.

CA Public Keys

Public keys are the basis for secure encryption, so there must be a reliable and secure way to obtain public keys. A CA certifies a public key, by generating a Certificate. A digital signature acts as proof of a sender's identity. A Certificate's security is based on the difficulty of obtaining and reading the physical device on which the Certificate is stored, and the secrecy of the access password.

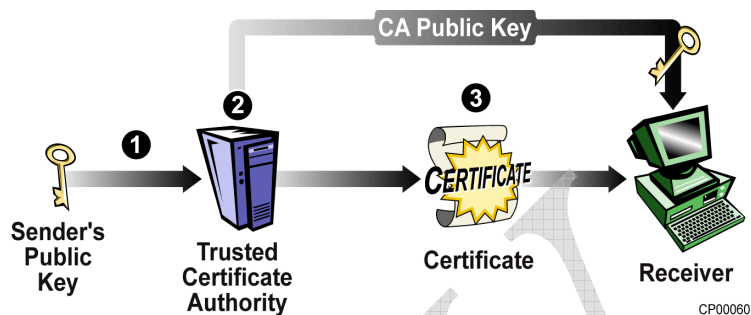


Figure 3-32: CA Action

The actions taken by a CA are as follows:

1. A sender sends his public key to a CA in a secure manner.
2. The CA signs the public key with its own private key, creating a CA public key.
3. The CA creates a Certificate with its public and private keys.

The receiver then authenticates the sender's public key, by matching the CA public key to the CA private key on the Certificate.

A Certificate is issued by a trusted CA and identifies and contains information about the bearer. Information from the CA might be:

- A person or host's unique identifier, such as his LDAP Distinguished Name.
- A person's public key: proof that the Certificate belongs to the bearer, as the signature can be verified.
- The CA's unique identifier, so anyone examining the Certificate knows who issued it.
- An expiration date.

- A digital signature, signed with the CA's private key: proof that the Certificate has not been tampered with

The Certificate, including its hash, is signed by the CA, proving that the Certificate could only have been created by the CA.

A Certificate is often embedded in a token, which is either an encrypted file or a hardware device, such as a smart card. The token has a password. Only someone who has a token, file, or device in his possession and knows the password can use the Certificate.

A Certificate can also be provided by the sender of a message directly. A receiver can verify a Certificate, using the steps listed earlier to verify an encrypted message. A sender proves his identity by sending a message consisting of a digital signature encrypted with his private key. The message also contains the sender's Certificate, and includes a unique identifier (such as the sender's LDAP DN and IP address).

After identities have been proved, the receiver and sender can use the other's public keys with confidence, as they are certified by Certificates from a trusted CA. Usually public keys are used to negotiate a secret key for encrypting actual messages. In VPN's, Certificates are also used by encrypting entities (such as Gateways), to identify themselves and supply their public keys to their peers.

Creating Certificates

There are several different ways in which a user can acquire his Certificate:

- A file or profile is created by the user or by the CA:
 - The user can create a profile on his own computer using client software, such as VPN-1 SecureClient. The profile data is then stored on a disk or hardware token, to minimize the possibility of unauthorized copying and misuse. The profile data is further protected by an access password.

- The CA can create the profile data and give it to the user. This method centralizes the creation of profiles, but may be impractical in a geographically dispersed organization.
- The user registers with the CA using a Web browser, and then exports the Certificate and private key for use with other applications.
- The user creates a Certificate registration-request file, then transfers the file by mail, FTP, and so on to the CA. The CA approves the request and generates the Certificate as a file, transferring it back to the user.

When a user leaves an organization or when a key is compromised (if a token is lost or stolen, for example), the user's Certificate must be revoked. The CA does this by issuing and distributing a Certification Revocation List (CRL). Before accepting a Certificate, the CRL should be checked to confirm that the Certificate has not been revoked. The CRL's distribution point is usually a Web server specified in the Certificate.

COPY