Cisco VPN

Chapter 2 - Technologies

Computer Networks - Winter 2024

The Complete Cisco VPN Configuration Guide

By Richard Deal

Meet Our Team

Seyed Mohsen Razavi
Key Exchanges &
Authentication Methods - Part 1

Shahin Kohzadpour
Keys, Encryption &
Packet Authentication

Seyed Mahdi Mahdavi
Authentication
Methods - Part 2

Keys, Encryption & Packet Authentication

Part 1 - Shahin Kohzadpour

VPN Implementations (Review)

- Secure Socket Layer **(SSL)**
- Layer 2 Tunnel Protocol **(L2TP)**
- Internet Protocol Security **(IPsec)**
- Point-to-Point Tunneling Protocol **(PPTP)**

Keys :

- key = a tool to ***open*** a ***locked*** door, vice versa
- we use keys to **protect information**
- **a data key == a user account password == PIN**

Key Usage :

- In **network security**, keys serve a **multi-functional process**

For example, keys are used for all of these three critical VPN functions:

- Encryption
- Packet integrity checking
- Authentication

Types of keying implementations :

- Symmetric
- Asymmetric

Symmetric Keys:

- use the same **single** key to **encrypt** and **decrypt** information
- So it’s fairly simple and thus **very efficient**.
- They tend to work **very quickly**

Symmetric Keys:

- use the same **single** key to **encrypt** and **decrypt** information
- So it’s fairly simple and thus **very efficient**.
- They tend to work **very quickly**
- Used in **encryption** and **packet integrity checking**

encryption algorithms and standards With symmetric keying

- **DES**
- 3DES
- CAST
- IDEA
- RC-4
- RC-6
- Skipjack
- AES
- MD5 (Hashing function)
- SHA (Hashing function)

One Problem :

- **RouterA** and **RouterB**, are performing **DES encryption**
- **RouterA** generates the **symmetric key** for DES,
- **RouterB** also **will need** this **same** key to decrypt information that **RouterA** sends it.

two basic ways to accomplish :

- Pre-sharing keys : ⟶ doesn't scale very well
  Pre-share the keys, out-of-band between the two devices.

- Using a secure connection:
  1. Use an existing secure, protected connection to send keys across
  2. Create a new protected connection to send keys across (catch-22)

Asymmetric Keys:

- uses two keys:

  Private keys :

  Kept secret by the source and is never shared with any other device.

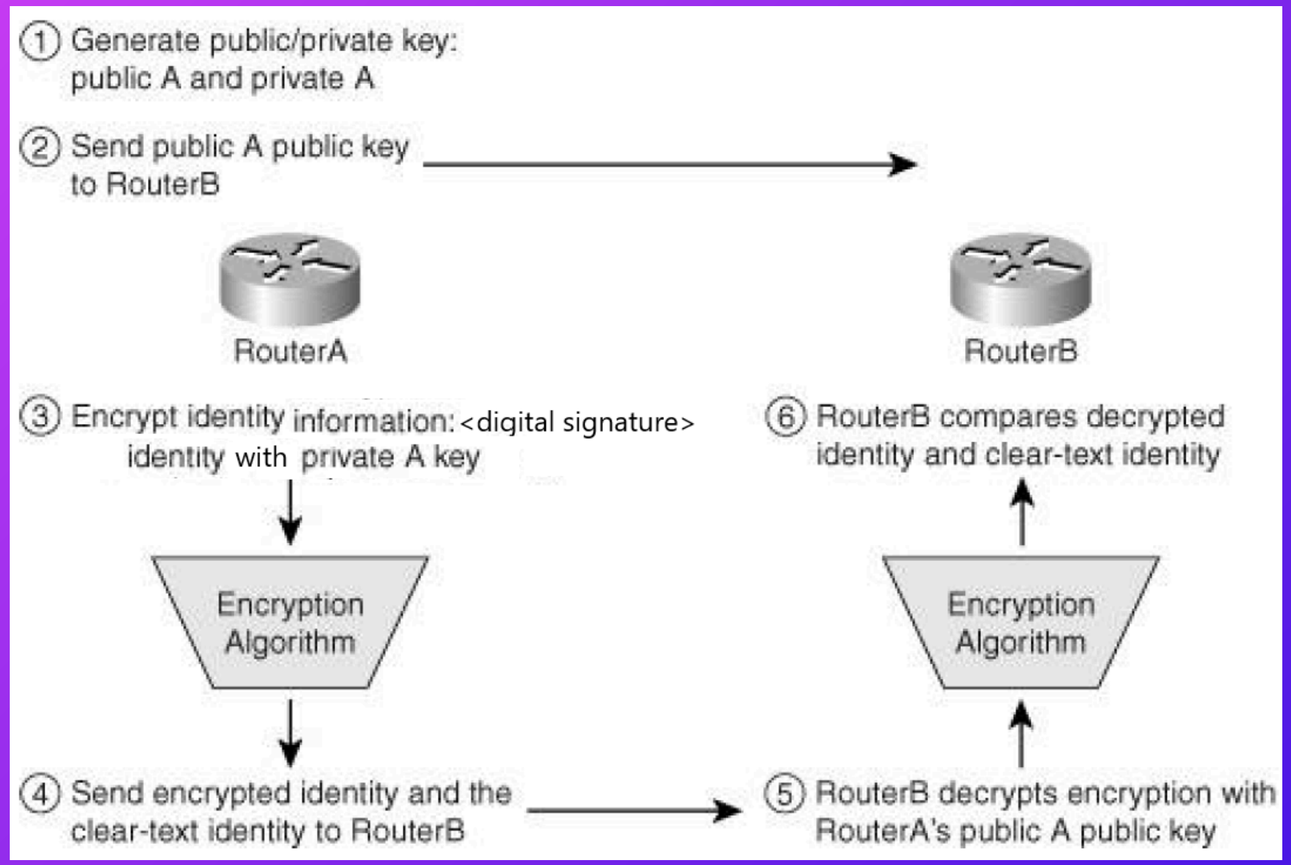
  Public keys :

  Is given out to other devices.

- We **can't** randomly choose any value for the two keys.
- A **special algorithm** is used to create the keys
- They need to have a symbiotic relationship with each other to provide protection

Asymmetric Keying and Authentication :

RouterA needs to authenticate to **RouterB**.



Packet Authentication :

- Packet authentication is used for two purposes:
  a. To provide **data origin** authentication
  b. To **detect** packets that have been **tampered** with

Packet Authentication Implementation :

- **Hashing functions** are used to **create a digital signature**
- **Input :**

  a **variable-length input**, such as user **data** or a **packet**

  along with a **key** and feeding it into a hashing function
- **Output :**
- The output is a **fixed-length** result
- fixed output == digital signature == fingerprint

Hashing Message Authentication Codes :

- are a **subset** of **hashing functions**
- use a shared secret **symmetric** key to create the fingerprint

Examples:

- SHA
- MD5

Why Perform Packet Authentication Alongside Encryption?

*Challenges with Encryption **Alone**:*

1. Verification Issue:

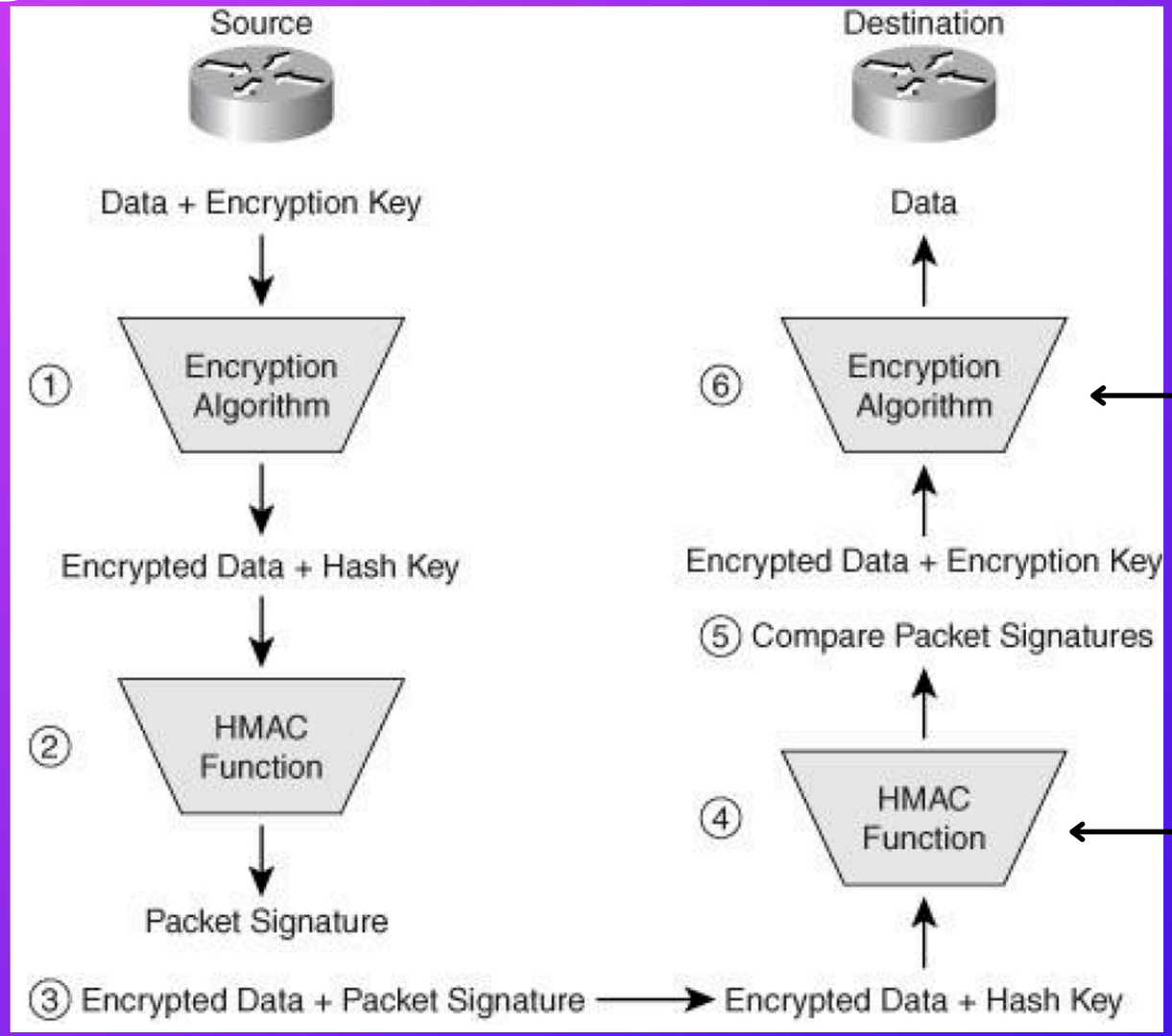
Decryption doesn't confirm authenticity or integrity of **data**.

2. CPU Overhead:

Attackers can **spoof** packets, forcing the device to **waste CPU cycles** decrypting invalid traffic.

Solution: Hashing Functions

HMAC Signature Creation and Verification :



Key Exchanges & Authentication Methods

Part 2 - Seyed Mohsen Razavi Zadegan

Key Exchange

- **Key Sharing Dilemma**

  Pre-Share the key
  Encrypt key with asymmetric keying algorithm
  Use an already encrypted connection

- **Diffie-Hellman Algorithm**

  The public/private key cryptography process was originally credited to Whitfield Diffie, Martin Hellman, and Ralph Merkle in 1976.

- **Limitations of Key Exchange Methods**

Key Exchange Dilemma

- **Use Encrypted Connections**

It is possible for an attacker to eavesdrop the key if we use unencrymented enviroments like telnet. It is prefered to use encrypted programs like SSH.

- **Pre-Share the Key**

Send the key to other peer out of bound, meaning to write the key on a peace of paper or a flash disk then take it to the peer. It is not scalable and hard to change the key priodically.

- **Use Asymmetric Algorithm**

Both peers share their public key with each other, The sender encrypts the data with its private key and reciever decrypts it with senders public key. It takes 15-20 minute to encrypt, send and decrypt 100 MB of data.

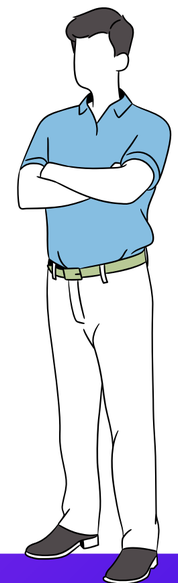
Key Exchange

Diffie-Hellman Algorithm

DH is typically not used to encrypt user data, but is, in most cases, used by VPN implementations to share keying information securely, such as DES, 3DES, AES, SHA, MD5 and other symmetric keys, across an insecure public network, like the Internet.

Key Exchange

Diffie-Hellman Algorithm

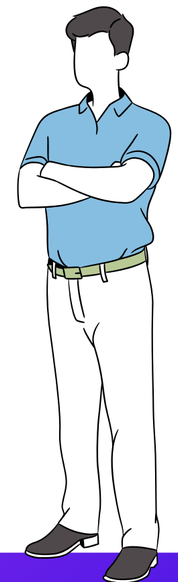


Key Exchange

Diffie-Hellman Algorithm

Base = 2

Modulos = 19



Key Exchange

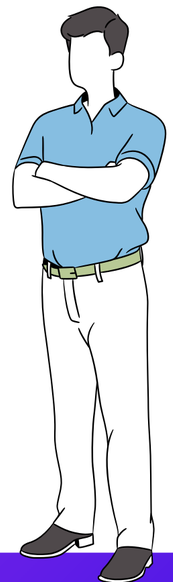
Diffie-Hellman Algorithm

Private Key: 8

Base = 2

Modulos = 19

Private Key: 15



Key Exchange

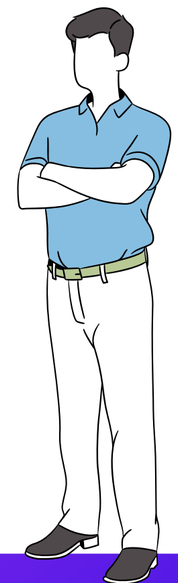
Diffie-Hellman Algorithm

$2^{8} \bmod 19$

$2^{15} \bmod 19$

Base = 2

Modulos = 19



Key Exchange

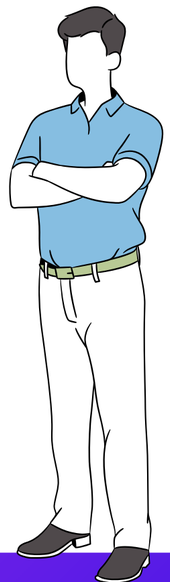
Diffie-Hellman Algorithm

$2^{8} \bmod 19 = 9$

$2^{15} \bmod 19 = 12$

Base = 2

Modulos = 19



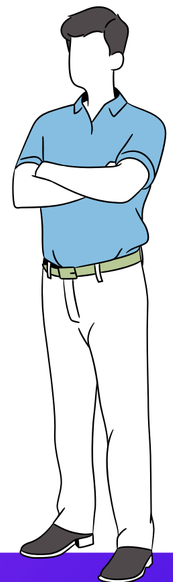
Key Exchange

Diffie-Hellman Algorithm

Public Key: 9

8

Public Key: 12



15

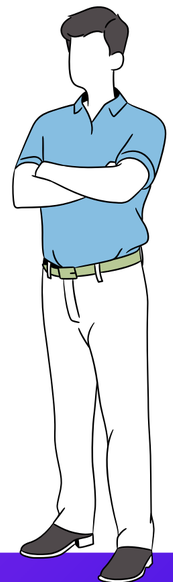
Key Exchange

Diffie-Hellman Algorithm

Public Key: 12

8

Public Key: 9



15

Key Exchange

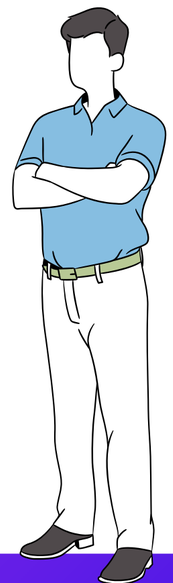
Diffie-Hellman Algorithm

$12^{8} \bmod 19$

$9^{15} \bmod 19$

Base = 2

Modulos = 19



Key Exchange

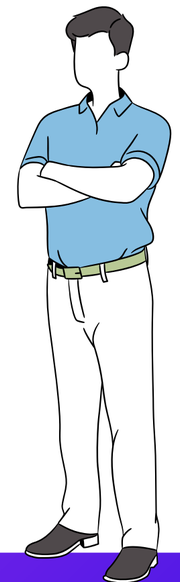
Diffie-Hellman Algorithm

$12^{8} \bmod 19 = 11$

$9^{15} \bmod 19 = 11$

Base = 2

Modulos = 19



Key Exchange

Diffie-Hellman Algorithm

Symmetric Key: 11

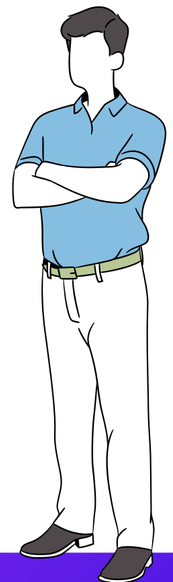
Public Key: 9

Private Key: 8

Symmetric Key: 11

Public Key: 12

Private Key: 15



Key Exchange

Limitations of Key Exchange Methods

- **Asymmetric Keying**

One strength of asymmetric keying algorithms is that the private key, used to decrypt information, is never sent across the network. In a simple asymmetric algorithm implementation, the eavesdropping attacker would have to know the private key to decrypt information

Key Exchange

Limitations of Key Exchange Methods

- **Asymmetric Keying**

One strength of asymmetric keying algorithms is that the private key, used to decrypt information, is never sent across the network. In a simple asymmetric algorithm implementation, the eavesdropping attacker would have to know the private key to decrypt information

- **Diffie-Hellman Algorithm**

An attacker, even if he is eavesdropping on the public key exchange process and sees the exchanged public key or keys, wouldn't be able to use this to decrypt any transmitted information. The attacker would have to know of one of the two private keys to decrypt information. DH does have one main weakness: it is susceptible to a man-in-the-middle attack.

Authentication Methods - Part 1

- Authentication Solutions
- Digital Certificate

Authentication Methods

Authentication Solutions

- **Device Authentication**
  - This method is used in site-to-site and remote access VPNs.
  - Either keying information is pre-shared to assist with the identification process, or it is acquired and verified when the devices need to communicate with each other via digital certificates.

Authentication Methods

Authentication Solutions

- **Device Authentication**
  - This method is used in site-to-site and remote access VPNs.
  - Either keying information is pre-shared to assist with the identification process, or it is acquired and verified when the devices need to communicate with each other via digital certificates.

- **User Authentication**
  - This method is used only in remote access VPNs.
  - With device authentication, the keying information typically is stored on the device.
  - This can be a concern if the device is broken into or stolen, as could easily happen with a laptop or PC.

Authentication Solutions

Device Authentications

- **Digital Certificates**
  - Definition of digital certificates
  - Certificate authority
  - Standards and components
  - Acquiring certificates
  - Using certificates

- **Pre-Shared Symmetric Key**

The sender sends it’s informations like IP address, hostname and digital signature, and the other peer validates the received data.

- **Pre-Shared Asymmetric Key**

Each peer makes it’s own public/private key. public keys are shared out-of-bound. Then sender sends it’s information and result of encryption of the information with other peers public key. The receiver validates data with it’s private key.

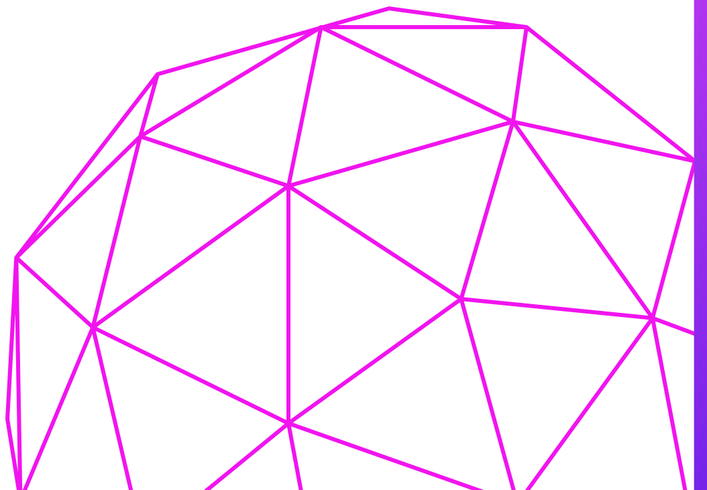
Digital Certificates

Definition Of Digital Certificates

Certificates contain information to assist in the authentication process. Unlike pre-shared key authentication methods, certificates are not pre-shared. Instead, only when devices need to make connections with each other are certificates shared. Digital certificates are based on the use of asymmetric (public/private) keys. You'll actually find many things on a digital certificate.

3 main things found on a digital certificate

- **device's identity information**
- **its public key**
- **its signature, created with its corresponding private key**



Digital Certificates

Certificate Authority (CA)

- **Certificate Authority Role**

A CA performs a similar function to a notary. A notary verifies and validates a person's identity when that person signs a document.

The CA is the most trusted device and is the repository of certificates. All devices that want to communicate with each other in a secure fashion must obtain their certificates from the same trusted source.

Digital Certificates

PKCS #10

- **Public Key Cryptography Standards**

PKCS #10 defines the actual information and format that a device needs to include and use when creating and requesting its personal certificate. The CA will then use this information to create the device's personal certificate, commonly called an identity certificate.

- **The identity information that is sent in the PKCS #10 certificate request can include the following:**

Common Name or Distinguished Name (CN or DN),
Organizational Unit (OU) or Department, Organization (O), Locality (L)
State Province (SP), Country, Subject Alternative Name (FQDN),
Subject Alternative Name (E-mail Address), Key Size, **Public key,** Challenge Password

Digital Certificates

Certificate Authority (CA)

- **Hirarchical Certificate Authority**

In a large-scale deployment of certificates in an enterprise (probably global) network, the two devices that want to establish a secure connection might be using different CAs. How do they trust on them selves ?

A higher level CA with some low level CAs is needed to handle this problem. The ability to create and validate certificates is delegated through a hierarchical chain of CAs. At the top of the hierarchy is the root CA, which is the most trusted device. The root CA creates certificates for the subordinate CAs.

Authentication Methods - Part 2

Part 3 - Seyed Mahdi Mahdavi

X.509v3 Certificates

1 Certificate Issuance Process

- Device submits a certificate request in PKCS#10 format.
- Certificate Authority (CA) reviews the request and issues a digital certificate.
- Standard used: X.509.
- More explanation: RFC 3280

X.509v3 Certificates

2 Types of Certificates

- Root Certificate: Owned by the CA itself.
- Identity Certificate: Issued for devices requiring authentication.

3 Certificate Information

- Details about the CA (name, serial number, validity period, public key, etc.).
- Information from the requesting device (name, optional PKCS#10 details).

X.509v3 Certificates

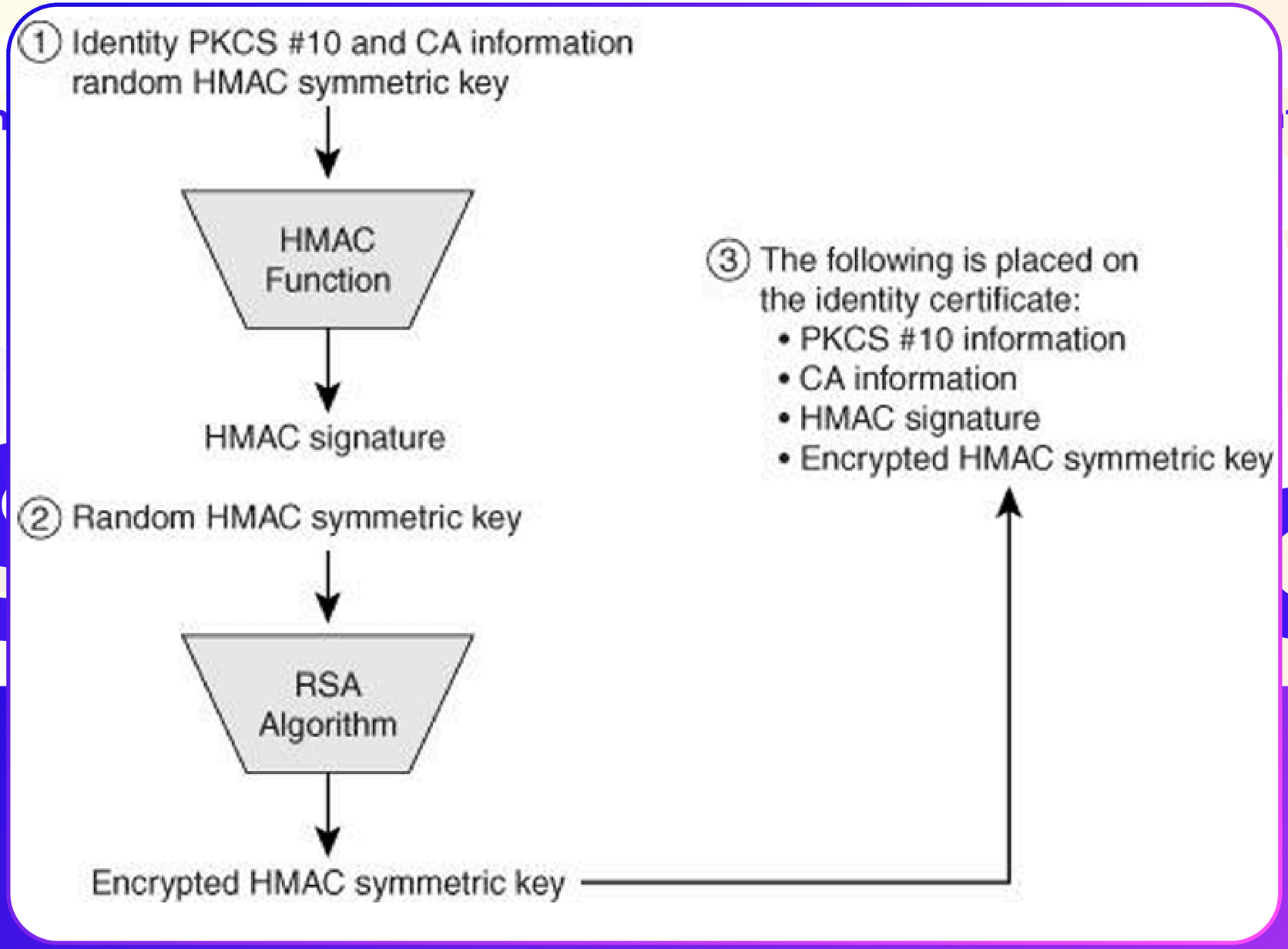
3 Certificate issuance

- Digital Signature:
  a. CA generates a random symmetric key.
  b. Certificate data and symmetric key are processed using HMAC function.
  c. HMAC output (signature) is encrypted with CA's private key.
  d. Encrypted signature and encrypted random HMAC symmetric key are embedded in the certificate.

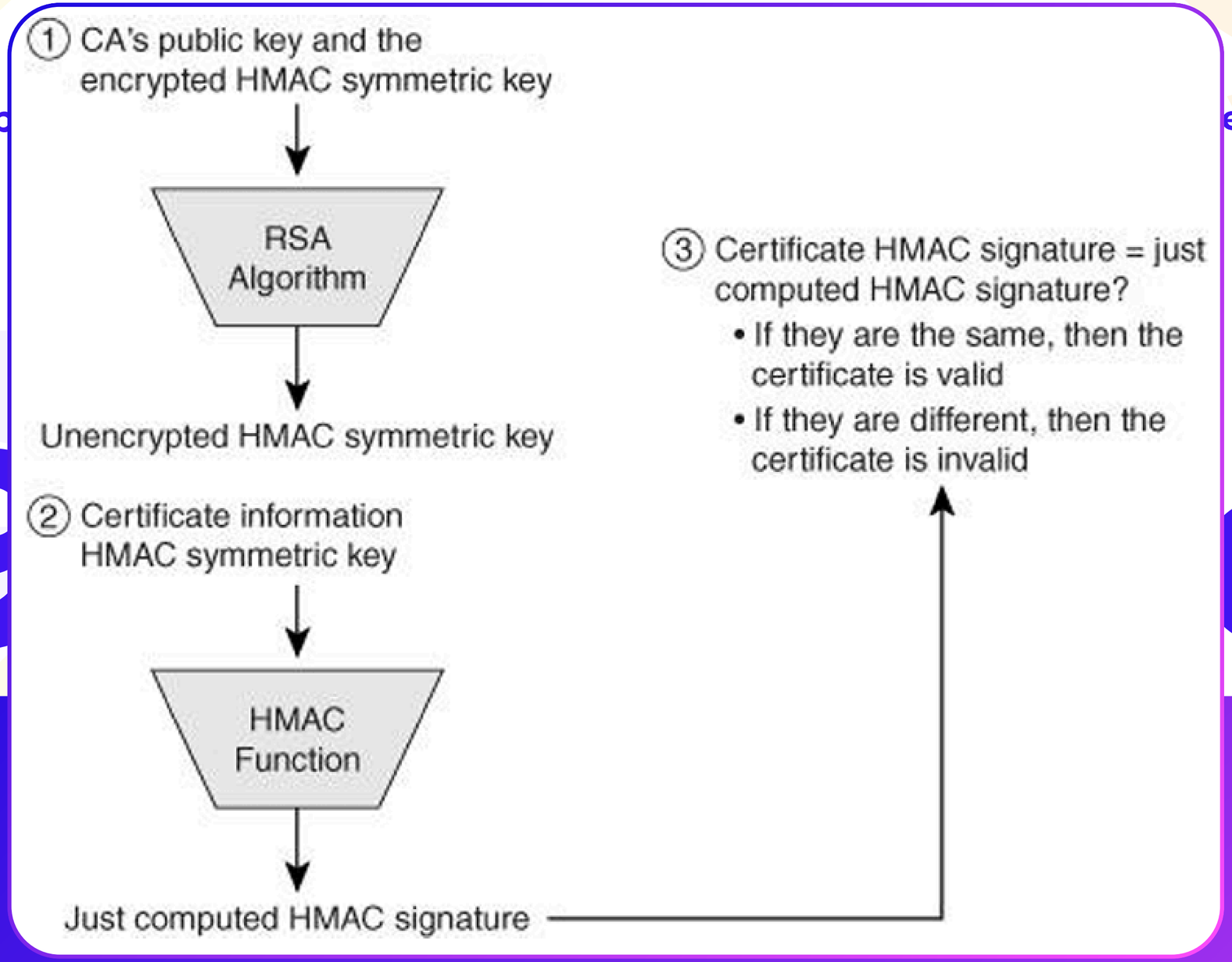
X.509v3 Certificates

4 Certificate Validation Process

- Device decrypts the symmetric key using the CA's public key.
- Certificate data and symmetric key are processed with the HMAC function.
- The output is compared with the signature in the certificate.
- If matched, the certificate is considered valid.



3 Certificate issuance



5 Certificate Validation Process

PKCS #7 Standard

A Standard for Secure Certificate Transmission

1 Overview

- PKCS #7 is a standard for securely transmitting digital certificates (e.g., X.509v3).
- Ensures encrypted transfer and prevents tampering during transit.

PKCS #7 Standard

2 Key Features

1. Certificate Encryption

- Certificates are encrypted with the recipient device's public key.
- Only the intended recipient can decrypt the certificate using their private key.

2. Tamper Detection

- Any alteration during transit is detected by the receiving device, invalidating the certificate.

3. Batch Certificate Transmission

- Supports simultaneous transmission of multiple certificates (e.g., root and device certificates).

PKCS #7 Standard

3 File Format

- **.DER / .CER:** Binary format for certificates.
- **.PEM:** Text-based format for readability.
- **.P7B / .P7C:** Format for signed or encrypted certificates using PKCS #7.
- **.PFX/ .P12:** Secure format including certificates and private keys.

SCEP

Simple Certificate Enrollment Protocol

1 In-band Certificate Acquisition

- SCEP provides a method for requesting and obtaining digital certificates within the network (in-band) using HTTP.
- Eliminates the need for manual methods like email.

SCEP

2 Standards Used

- **PKCS#10:** For certificate requests.
- **PKCS#7:** For encrypted certificate delivery.

3 Advantages

- **Speed & Scalability:** Faster and more scalable than offline methods.
- **Automation:** Supports automated certificate enrollment (not manual).
- **HTTP Protocol:** Widely supported in most networks.

SCEP

4 How it works?

- The device prepares a certificate request in PKCS#10 format.
- Sends the **request** to the CA using **HTTP**.
- The CA processes the request and generates the certificate.
- The CA encrypts the certificate in PKCS#7 format and sends it back (**response**) via **HTTP**.

CRLs

Certificate Revocation Lists

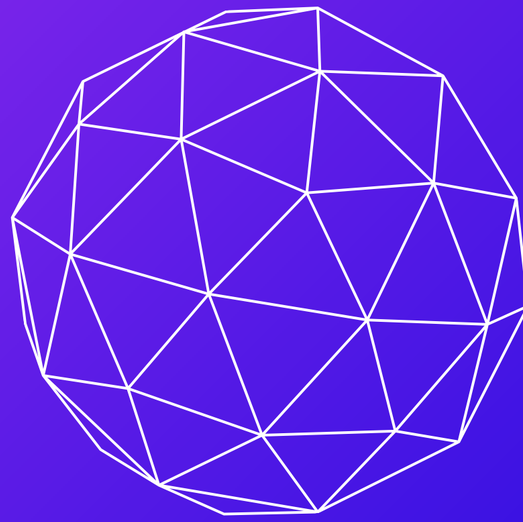
1 Definition

- A Certificate Revocation List (CRL) is a list of digital certificates that have been revoked by the CA.

CRLs

2 Reasons for Revocation

- **Compromised Private Key:** When the private key related to a certificate is exposed to unauthorized parties.
- **Compromised Certificate:** When the certificate itself is exposed to unauthorized parties.
- **Device Out of Service:** When the device related to the certificate is no longer in use.
- **Certificate Expiration:** When the certificate reaches its expiration date.
- **Changes in Security Policy:** When security policies change, requiring new key lengths or reissuance of certificates.



CRLs

4 Importance of CRLs

- **Certificate Validity Check:** CRLs help devices determine if a certificate is still valid or has been revoked, ensuring secure communications.
- **Preventing Use of Compromised Certificates:** Devices can prevent the use of certificates that have been compromised by checking the CRL.

CRLs

5 CRL Limitations

- **Large File Size:** In large environments, CRLs can become very large, leading to:
  - **Long Download Times:** Downloading large CRLs can take significant time.
  - **Storage Issues:** Storing large CRLs can consume considerable storage space on devices.

CRLs

6 OCSP (Online Certificate Status Protocol)

- **Alternative to CRLs:** OCSP provides a more efficient method for checking certificate status.
- **Real-time Status Check:** OCSP allows devices to check the certificate status directly from the CA in real-time.

Acquiring Certificates

1 Importance of Certificates for Authentication

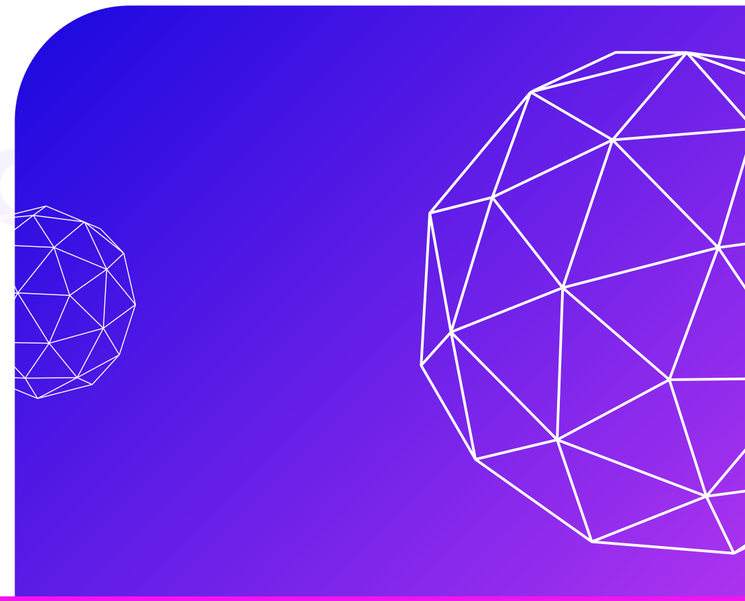
- **Essential for Authentication:**
  - Devices communicating in secure environments require identity certificates.
- **Two Types of Acquiring:**
  - **File-based Enrolled**
  - **Network-based Enrollment**

Acquiring Certificates

2 Two Methods for Acquiring Certificates

A. File-Based Enrollment (Manual/Offline):

1. **Process:**
   - Device creates a PKCS#10 request.
   - Sent manually to CA (e.g., via email).
   - CA signs and returns the certificate manually.
2. **Disadvantages:**
   - Time-consuming and error-prone.
   - Unsuitable for large-scale deployments.



Acquiring Certificates

2 Two Methods for Acquiring Certificates

B. Network-Based Enrollment (Automated/Online):

1. **Process:**
   - Device sends a PKCS#10 request to CA via SCEP.
   - CA processes and signs the certificate automatically.
   - Certificate is returned via the network.
2. **Advantages:**
   - Faster, efficient, and scalable.

Acquiring Certificates

4 Network-Based Enrollment: SCEP Implementation

A. Overview of SCEP Configuration:

- Devices must be configured with the following CA details:
  - Fully Qualified Domain Name (FQDN) of the CA.
  - SCEP URL, e.g.,:
    http://fqdn_or_ip_address_of_ca_server/certsrv/mscep/mscep.dll

Acquiring Certificates

4 Network-Based Enrollment: SCEP Implementation

http://fqdn_or_ip_address_of_ca_server/certsrv/mscep/mscep.dll

- **http://FQDN_or_IP_address_of_CA_server/:** The address of the CA server on which the SCEP service is configured.
- **certsrv/:** The path to the certificate issuance service.
- **mscep/mscep.dll:** The program file responsible for handling SCEP requests on the server.

Using Certificates

1 Overview

- When a device has both root and identity certificates, it can use them in the authentication process when establishing a VPN session with a remote peer. Fortunately, using certificates is much simpler than obtaining them.
- There are **three key checks** that a device performs during certificate-based authentication with a remote peer: ...

Using Certificates

2 Key Authentication Checks

1. Is the Peer’s Identity Certificate **Signed** by a **Trusted CA**?
    - Can the signature be verified using the public key stored locally in the root certificate of the CA?
2. Is the Certificate Still **Within** Its **Validity Period**?
    - Does the certificate fall within its start and end date range?
3. Is the Certificate **Revoked**?
    - Is the certificate listed in the Certificate Revocation List (CRL)?

Using Certificates

3 Authentication Process

- If the device can verify all three points and answer "yes" to each question, it can confidently authenticate the remote peer, completing the authentication process successfully.
- The CRL check is optional in many device implementations.

4 Authentication Steps

1. **Initial Contact:**
    - Peers exchange their identity certificates.

Using Certificates

4 Authentication Steps

2. **Optional CA Query**:
   - Peers can contact the CA to retrieve the other peer's certificate to ensure it hasn't been revoked (using CRL if the CA supports it).
3. **Signature Verification**:
   - The device verifies the certificate signature using the CA's public key & random HMAC symmetric key (explained in X.509v3 Certificate part).

Using Certificates

4 Authentication Steps

4. **Time Validation:**
   - The device compares its time with the certificate's validity period. If the device's time is within the certificate's validity, the check is successful. If not, authentication fails.
5. **Checking CRLs:**
   - If enabled, the device will check the CRL for the certificate's serial number. If found, the certificate is considered invalid, and authentication fails. If not found, authentication proceeds successfully.

User Authentication

1 Remote Access and User Authentication

We learned about with the problems of remote access and PSKs.

- **Solution? Using certificates**

1. **Combining Device and User Authentication:**
   - VPN requires both device certificate and user credentials (username/password).
2. **One-Time Passwords (OTP):**
   - Generates a unique password for every session.
   - Reduces the risk of password guessing.

User Authentication

2 Remote Access and User Authentication

3. **Group Mutual Authentication:**
   - Asymmetric authentication where VPN gateway and user authenticate each other.
   - Two steps:
     i. VPN gateway authenticates itself using a certificate.
     ii. VPN gateway then authenticates the user.
   - Advantage:
     - More secure than PSK-based group authentication.

User Authentication

2 Remote Access and User Authentication

4. Split-Tunneling Considerations:

- Security Requirements:
  - Deploy software firewalls, antivirus, and anti-spyware on user devices.
  - Optionally use tools like Cisco Security Agent (CSA - an endpoint intrusion prevention system software) for advanced intrusion detection and access control.

We Appreciate Your Attention!