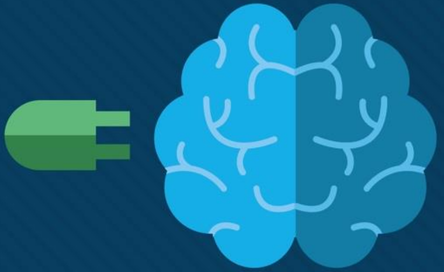
# Module 21: Cryptography

Instructor Materials

CyberOps Associate v1.0



# Module 21: Cryptography

CyberOps Associate v1.0

# Module Objectives

**Module Title:** Public Key Cryptography

**Module Objective**: Explain how the public key infrastructure (PKI) supports network security.

.

| Topic Title | Topic Objective |
|---|---|
| **Integrity and Authenticity** | Explain the role of cryptography in ensuring the integrity and authenticity of data. |
| **Confidentiality** | Explain how cryptographic approaches enhance data confidentiality. |
| **Public Key Cryptography** | Explain public key cryptography. |
| **Authorities and the PKI Trust System** | Explain how the public key infrastructure functions. |
| **Applications and Impacts of Cryptography** | Explain how the use of cryptography affects cybersecurity operations. |

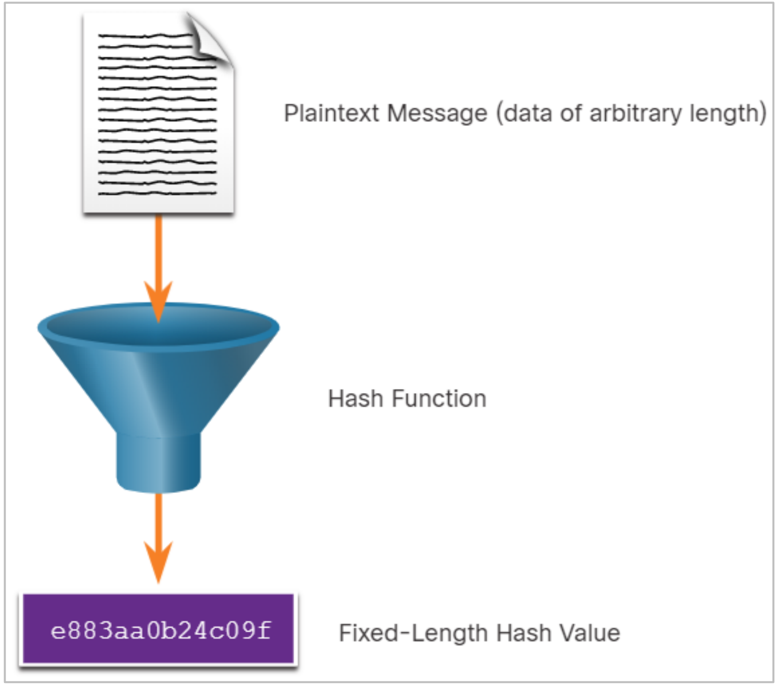
# 21.1 Integrity and Authenticity

# Securing Communications

- Organizations must provide support to secure the data internally as well as externally.
- The four elements of securing communications are:
  - **Data Integrity** - Guarantees that the message was not altered.
  - **Origin Authentication** - Guarantees that the message is not a forgery and it actually comes from whom it states.
  - **Data Confidentiality** - Guarantees that only authorized users can read the message.
  - **Data Non-Repudiation** - Guarantees that the sender cannot repudiate, or refute, the validity of a message sent.

# Cryptographic Hash Functions

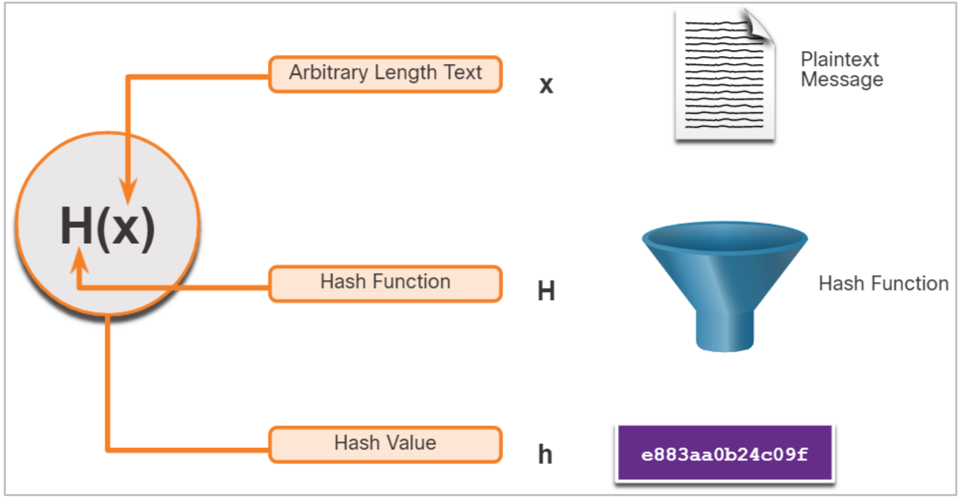
- Hashes are used to verify and ensure data integrity.
- Hashing is based on a one-way mathematical function that is relatively easy to compute, but significantly harder to reverse.
- A hash function takes a variable block of binary data, called the message, and produces a fixed-length, condensed representation, called the hash.
- The resulting hash is also sometimes called the message digest, digest, or digital fingerprint.
- With hash functions, it is computationally infeasible for two different sets of data to come up with the same hash output.
- Every time the data is changed or altered, the hash value also changes.



# Cryptographic Hash Operation

- Mathematically, the equation ***h= H(x)*** is used to explain how a hash algorithm operates.
- As shown in the figure, a hash function H takes an input x and returns a fixed-size string hash value h.

- A cryptographic hash function should have the following properties:
  - The input can be any length.
  - The output has a fixed length.
  - H(x) is relatively easy to compute for given x.
  - H(x) is one way and not reversible.
  - H(x) is collision free, meaning that two different input values will result in different hash values.



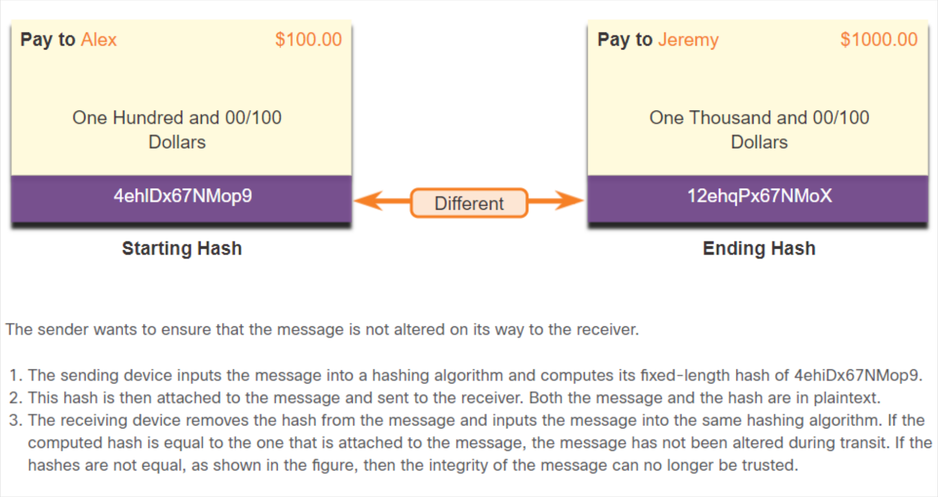
## Cryptography

# MD5 and SHA

- Hash functions are used to ensure the integrity of a message either accidentally or intentionally.
- In the figure, the sender is sending a $100 money transfer to Alex. The sender wants to ensure that the message is not altered on its way to the receiver.

There are four well-known hash functions:

- **MD5 with 128-bit digest** - A one-way function that produces a 128-bit hashed message. MD5 is a legacy algorithm.
- **SHA-1 -** Very similar to the MD5 hash functions. SHA-1 creates a 160-bit hashed message and is slightly slower than MD5.
- **SHA-2 -** If you are using SHA-2, then SHA-256, SHA-384, and SHA-512 algorithms should be used.



The sender wants to ensure that the message is not altered on its way to the receiver.

1. The sending device inputs the message into a hashing algorithm and computes its fixed-length hash of 4ehiDx67NMop9.
2. This hash is then attached to the message and sent to the receiver. Both the message and the hash are in plaintext.
3. The receiving device removes the hash from the message and inputs the message into the same hashing algorithm. If the computed hash is equal to the one that is attached to the message, the message has not been altered during transit. If the hashes are not equal, as shown in the figure, then the integrity of the message can no longer be trusted.

- **SHA-3 -** Next-generation algorithms and should be used whenever possible.

# MD5 and SHA (Contd.)

- While hashing can be used to detect accidental changes, it cannot be used to guard against deliberate changes that are made by a threat actor.
- There is no unique identifying information from the sender in the hashing procedure.
- This means that anyone can compute a hash for any data, as long as they have the correct hash function.
- Therefore, hashing is vulnerable to man-in-the-middle attacks and does not provide security to transmitted data. To provide integrity and origin authentication, something more is required.

***Note:*** *Hashing algorithms only protect against accidental changes and does not protect the data from changes deliberately made by a threat actor.*

# Origin Authentication

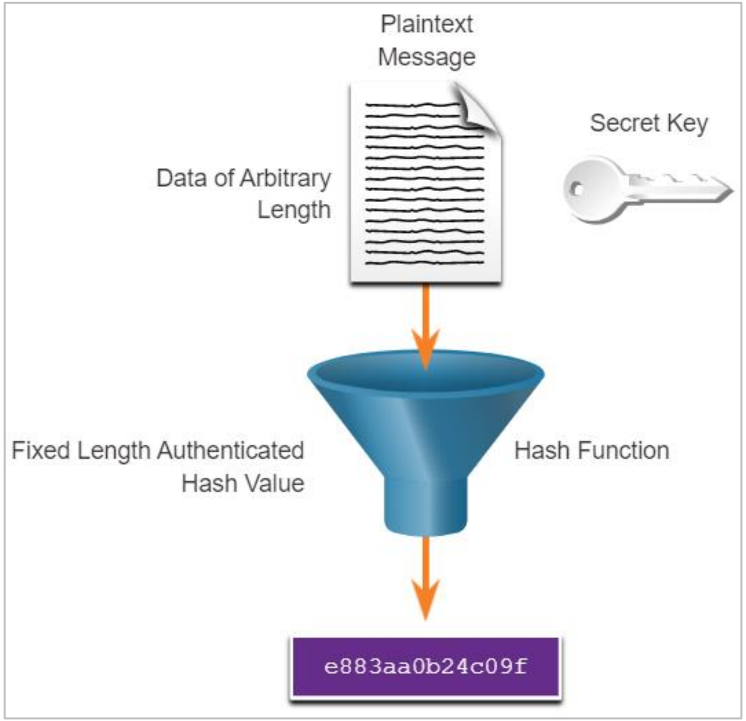
- To add origin authentication and integrity assurance, use a keyed-hash message authentication code (HMAC).
- HMAC uses an additional secret key as input to the hash function.

***Note:*** *Other Message Authentication Code (MAC) methods are also used. However, HMAC is used in many systems including SSL, IPsec, and SSH.*

# Origin Authentication (Contd.)

## HMAC Hashing Algorithm

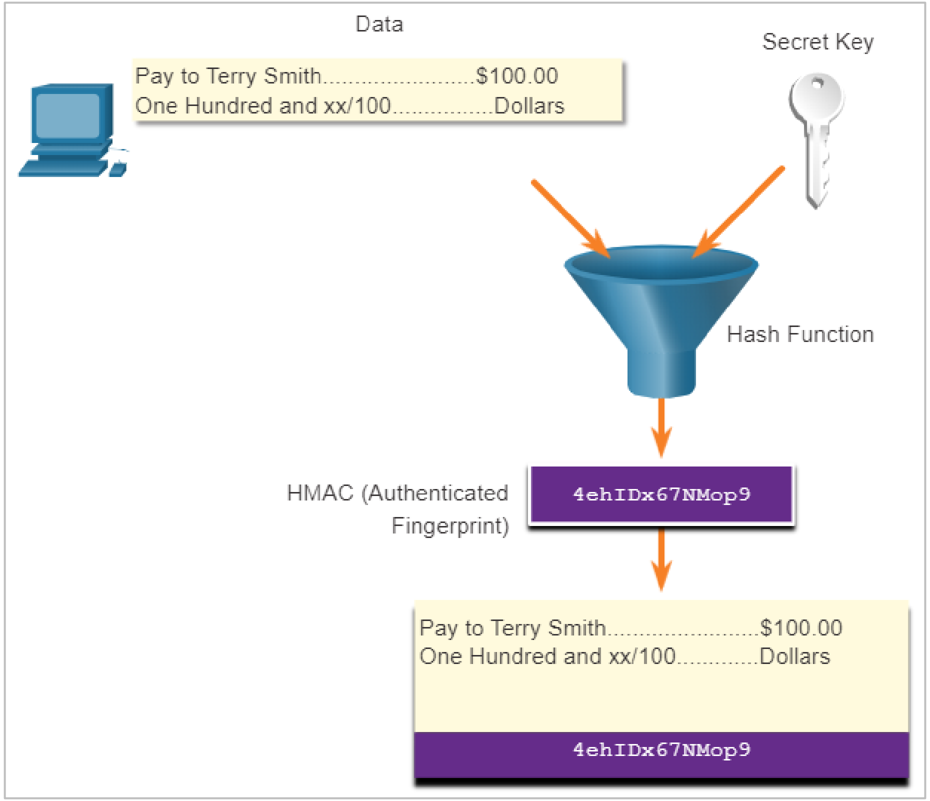
- An HMAC is calculated using any cryptographic algorithm that combines a cryptographic hash function with a secret key.
- Only the sender and the receiver know the secret key, and the output of the hash function depends on the input data and the secret key.
- Only parties who have access to that secret key can compute the digest of an HMAC function.
- If two parties share a secret key and use HMAC functions for authentication, a properly constructed HMAC digest of a message that a party has received indicates that the other party was the originator of the message.



# Origin Authentication (Contd.)

## Creating the HMAC Value

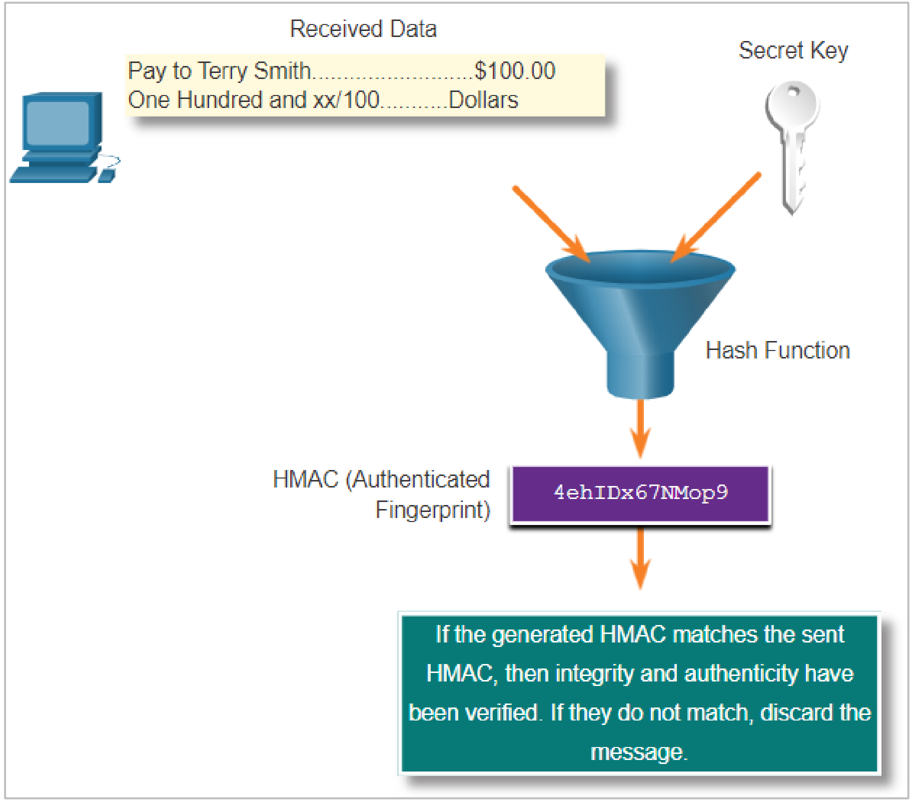
- As shown in the figure, the sending device inputs data into the hashing algorithm and calculates the fixed-length HMAC digest.
- This authenticated digest is then attached to the message and sent to the receiver.


Data
Pay to Terry Smith.......................$100.00
One Hundred and xx/100...............Dollars
Secret Key
Hash Function
HMAC (Authenticated Fingerprint)
4ehIDx67NMop9
Pay to Terry Smith.......................$100.00
One Hundred and xx/100............Dollars
4ehIDx67NMop9

# Origin Authentication (Contd.)

## Verifying the HMAC Value

- In the figure, the receiving device removes the digest from the message and uses the plaintext message with its secret key as input into the same hashing function.
- If the digest that is calculated by the receiving device is equal to the digest that was sent, the message has not been altered.
- Additionally, the origin of the message is authenticated because only the sender possesses a copy of the shared secret key. The HMAC function has ensured the authenticity of the message.


Received Data
Pay to Terry Smith.........................$100.00
One Hundred and xx/100...........Dollars
Secret Key
Hash Function
HMAC (Authenticated Fingerprint)
4ehIDx67NMop9

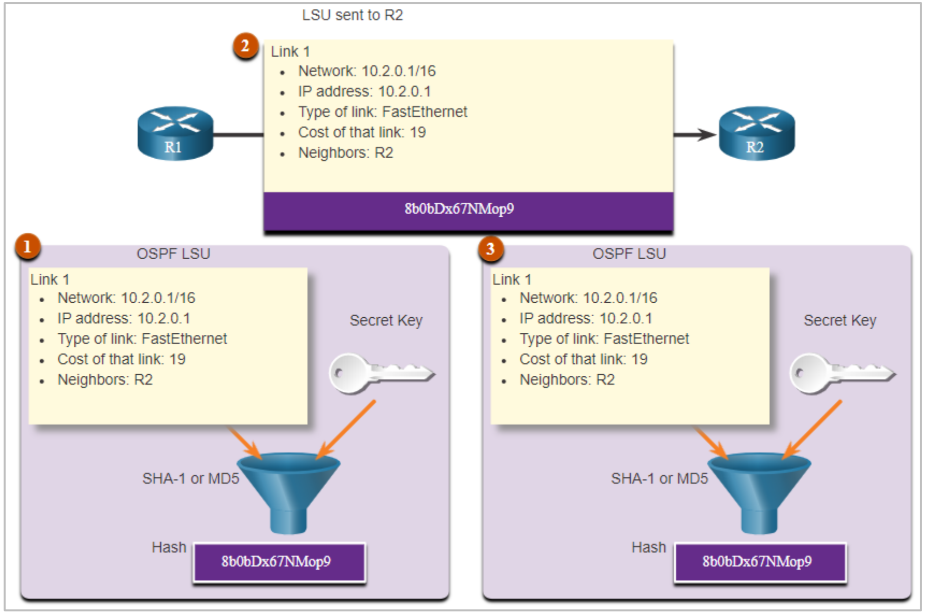
If the generated HMAC matches the sent HMAC, then integrity and authenticity have been verified. If they do not match, discard the message.

Cryptography

# Origin Authentication (Contd.)

## Cisco Router HMAC Example

- In the figure, HMACs are used by Cisco routers that are configured to use Open Shortest Path First (OSPF) routing authentication.
- R1 is sending a link state update (LSU) regarding a route to network 10.2.0.0/16:
  - R1 calculates the hash value using the LSU message and the secret key.
  - The resulting hash value is sent with the LSU to R2.
  - R2 calculates the hash value using the LSU and its secret key. R2 accepts the update if the hash values match. If they do not match, R2 discards the update.



# Lab – Hashing Things Out

In this lab, you will complete the following objectives:

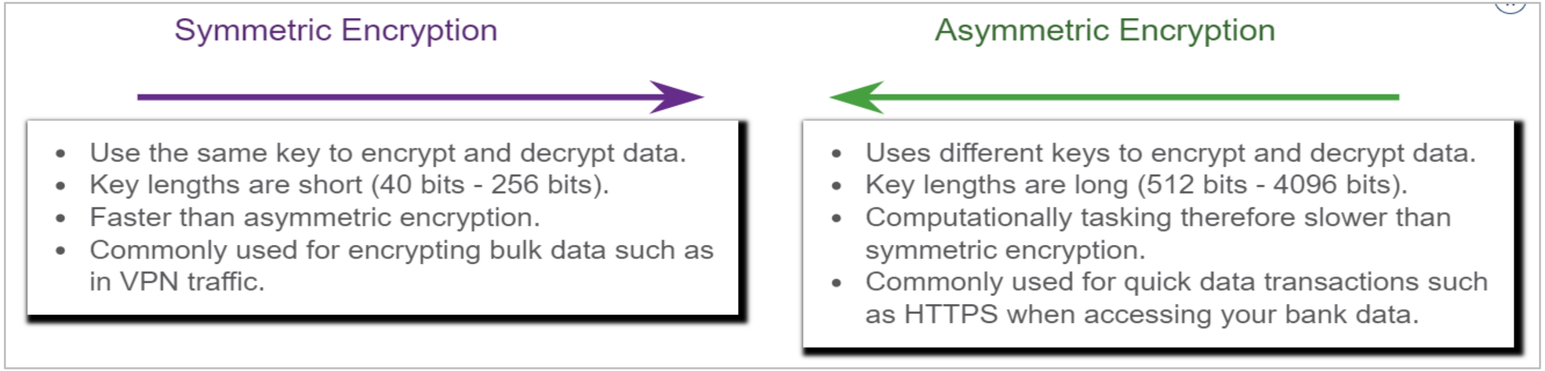
- Creating Hashes with OpenSSL
- Verifying Hashes

# 21.2 Confidentiality

Confidentiality

# Data Confidentiality

- There are two classes of encryption used to provide data confidentiality; asymmetric and symmetric. These two classes differ in how they use keys.
- Symmetric encryption algorithms such as Data Encryption Standard (DES), 3DES, and Advanced Encryption Standard (AES) are based on the premise that each communicating party knows the pre-shared key.
- Data confidentiality can also be ensured using asymmetric algorithms, including Rivest, Shamir, and Adleman (RSA) and the public key infrastructure (PKI).
- The figure highlights some differences between symmetric and asymmetric encryption.



Symmetric Encryption

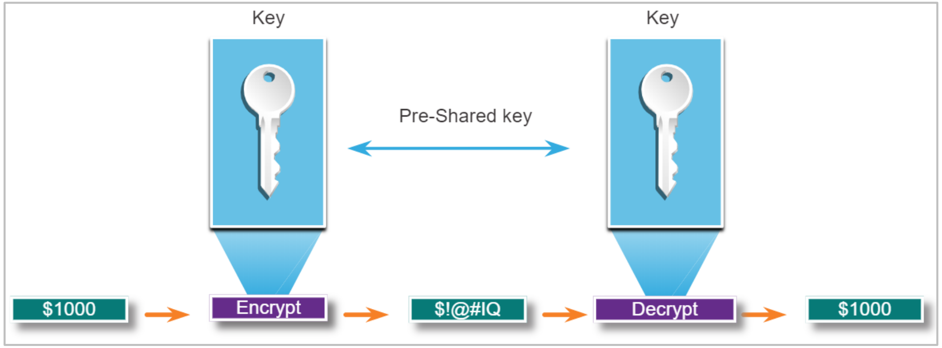
- Use the same key to encrypt and decrypt data.
- Key lengths are short (40 bits - 256 bits).
- Faster than asymmetric encryption.
- Commonly used for encrypting bulk data such as in VPN traffic.

Asymmetric Encryption

- Uses different keys to encrypt and decrypt data.
- Key lengths are long (512 bits - 4096 bits).
- Computationally tasking therefore slower than symmetric encryption.
- Commonly used for quick data transactions such as HTTPS when accessing your bank data.

# Symmetric Encryption

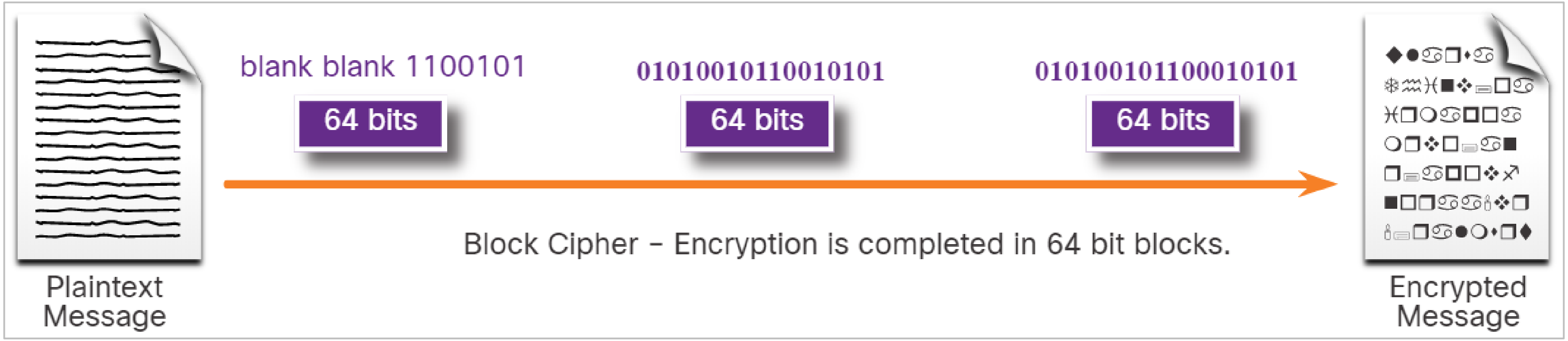
- Symmetric algorithms use the same pre-shared key (secret key) to encrypt and decrypt data.
- Symmetric encryption algorithms are commonly used with VPN traffic because they use less CPU resources than asymmetric encryption algorithms.
- When using these algorithms, the longer the key, the longer it will take for someone to discover the key.
- Most encryption keys are between 112 and 256 bits. Use a longer key for more secure communications.
- Symmetric encryption algorithms are sometimes classified as a block cipher or a stream cipher.



# Symmetric Encryption (Contd.)

## Block Ciphers

- Block ciphers transform a fixed-length block of plaintext into a common block of ciphertext of 64 or 128 bits.
- Common block ciphers include DES with a 64-bit block size and AES with a 128-bit block size.



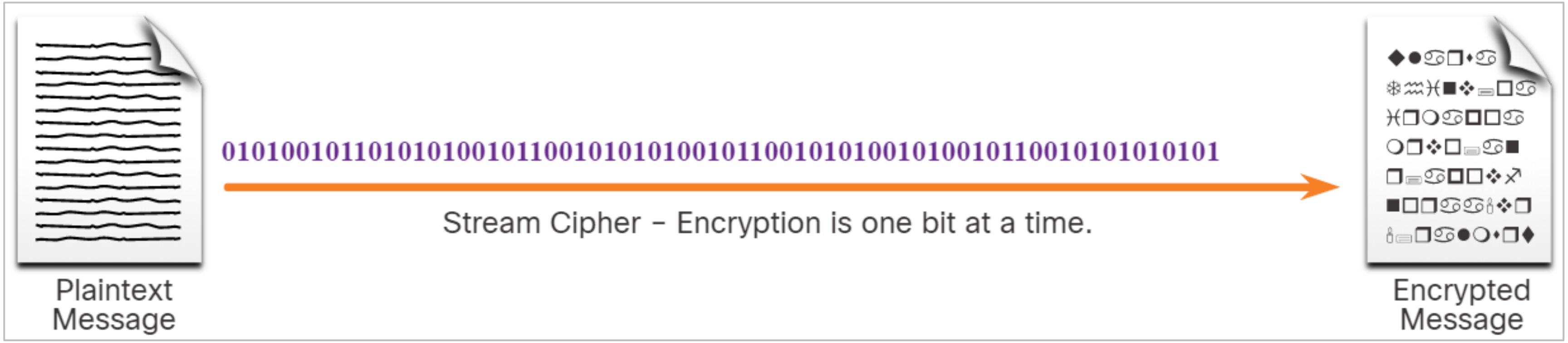
Block Cipher – Encryption is completed in 64 bit blocks.

Confidentiality

# Symmetric Encryption (Contd.)

## Stream Ciphers

- Stream ciphers encrypt plaintext one byte or one bit at a time.
- Stream ciphers are basically a block cipher with a block size of one byte or bit.
- Stream ciphers are typically faster than block ciphers because data is continuously encrypted.
- Examples include RC4 and A5 which is used to encrypt GSM cell phone communications.



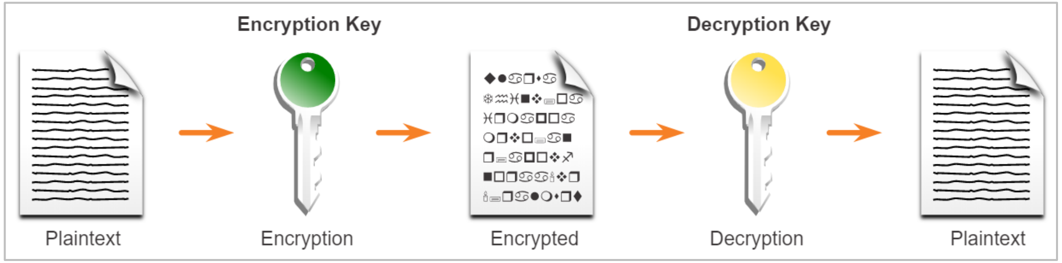
# Symmetric Encryption (Contd.)

Well-known symmetric encryption algorithms are described in the table.

| Symmetric Encryption Algorithms | Description |
|---|---|
| Data Encryption Standard (DES) | This is a legacy algorithm. It uses a short key length that makes it insecure. |
| 3DES (Triple DES) | This is the replacement for DES and repeats the DES algorithm three times. It should be avoided as it is scheduled to be retired in 2023. If implemented, use very short key lifetimes. |
| Advanced Encryption Standard (AES) | It offers combinations of 128-, 192-, or 256-bit keys to encrypt 128, 192, or 256 bit-long data blocks. |
| Software-Optimized Encryption Algorithm (SEAL) | It is a stream cipher that uses a 160-bit encryption key and has a lower impact on the CPU compared to other software-based algorithms. |
| Rivest ciphers (RC) series algorithms | RC4 is a stream cipher that was used to secure web traffic. It has been found to have multiple vulnerabilities which have made it insecure. RC4 should not be used. |

# Asymmetric Encryption

- Asymmetric algorithms, also called public-key algorithms, are designed in a way that the encryption and the decryption keys are different.
- Asymmetric algorithms use a public key and a private key. Both keys are capable of the encryption process, but the complementary paired key is required for decryption.
- The process is also reversible. Data that is encrypted with the public key requires the private key to decrypt.
- Asymmetric algorithms achieve confidentiality and authenticity by using this process.
- Asymmetric encryption can use key lengths between 512 to 4,096 bits.
- Asymmetric algorithms are substantially slower than symmetric algorithms.



# Asymmetric Encryption (Contd.)

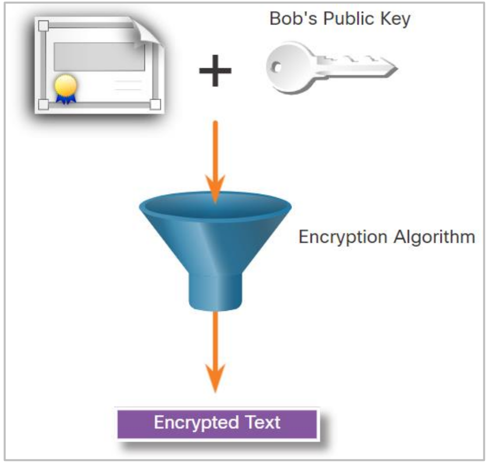
Common examples of asymmetric encryption algorithms are described in the table.

| Asymmetric Encryption Algorithms | Key Length | Description |
|---|---|---|
| Diffie-Hellman (DH) | 512, 1024, 2048, 3072, 4096 | This algorithm allows two parties to agree on a key that they can use to encrypt messages they want to send to each other. The security depends on the assumption that it is easy to raise a number to a certain power, but difficult to compute which power was used, given the number and the outcome. |
| Digital Signature Standard (DSS) and Digital Signature Algorithm (DSA) | 512 – 1024 | It specifies DSA as the algorithm for digital signatures. DSA is a public key algorithm based on the ElGamal signature scheme. Signature creation speed is similar to RSA, but is 10 to 40 times slower for verification. |
| Elliptic curve techniques | 224 or higher | Elliptic curve cryptography can be used to adapt many cryptographic algorithms, such as Diffie-Hellman or ElGamal. The main advantage of elliptic curve cryptography is that the keys can be much smaller. |

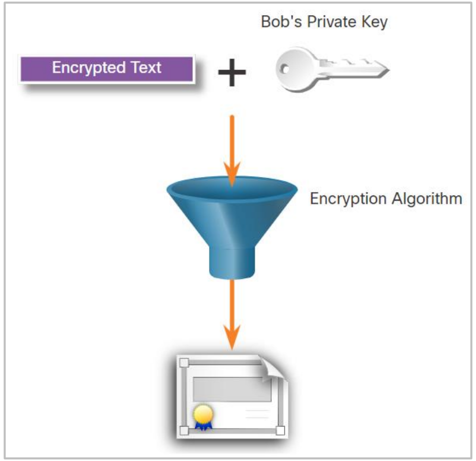
# Asymmetric Encryption - Confidentiality

- Asymmetric algorithms are used to provide confidentiality without pre-sharing a password.
- The confidentiality objective of asymmetric algorithms is initiated when the encryption process is started with the public key.
- The process can be summarized using the formula: **Public Key (Encrypt) + Private Key (Decrypt) = Confidentiality**
- When the public key is used to encrypt data, the private key must be used to decrypt data.
- Only one host has the private key; therefore, confidentiality is achieved.

**Example**: **Data exchange between Bob and Alice**



Alice acquires and uses Bob's public key to encrypt a message and then send it to Bob.



Bob decrypts the message with the private key and as he is the only one with the private key, confidentiality is achieved.

Confidentiality

# Asymmetric Encryption - Authentication

- The authentication objective of asymmetric algorithms is initiated with the private key encryption process.
- The process can be summarized using the formula: **Private Key (Encrypt) + Public Key (Decrypt) = Authentication**
- When the private key is used to encrypt the data, the corresponding public key must be used to decrypt the data.
- Because only one host has the private key, only that host could have encrypted the message, providing authentication of the sender.
- When a host successfully decrypts a message using a public key, it is trusted that the private key encrypted the message, which verifies who the sender is. This is a form of authentication.

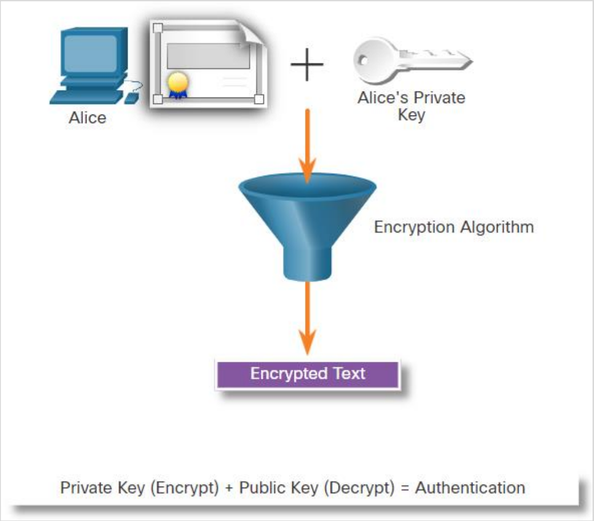
Confidentiality

# Asymmetric Encryption - Authentication (Contd.)

- Let's see how the private and public keys can be used to provide authentication to the data exchange between Bob and Alice.

### Alice uses her private key

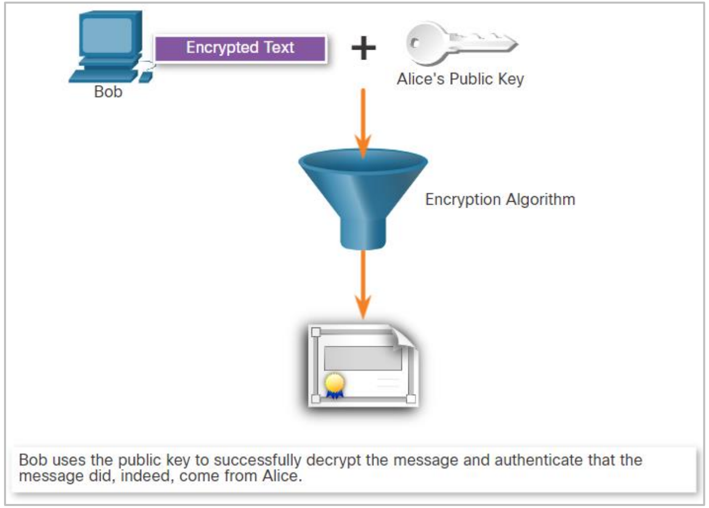
Alice encrypts a message using her private key and sends it to Bob.



Private Key (Encrypt) + Public Key (Decrypt) = Authentication

### Bob decrypts using the public key

After Bob obtains Alice’s public key, he uses it to decrypt the message and to authenticate that the message has been received from Alice.

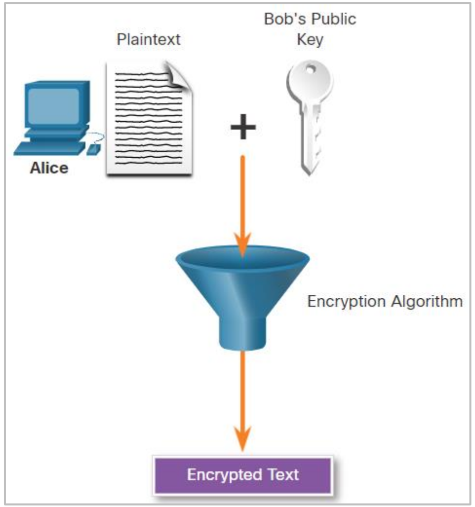


Bob uses the public key to successfully decrypt the message and authenticate that the message did, indeed, come from Alice.

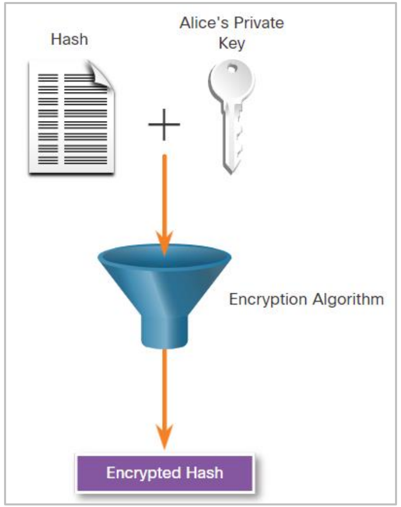
Confidentiality

# Asymmetric Encryption - Integrity

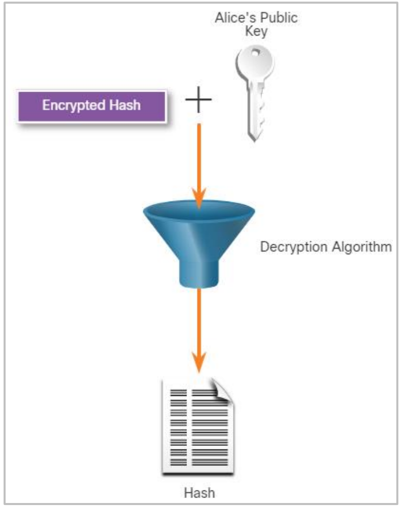
- Combining the two asymmetric encryption processes provides message confidentiality, authentication, and integrity. In this example, a message will be ciphered using Bob’s public key and a ciphered hash will be encrypted using Alice’s private key.



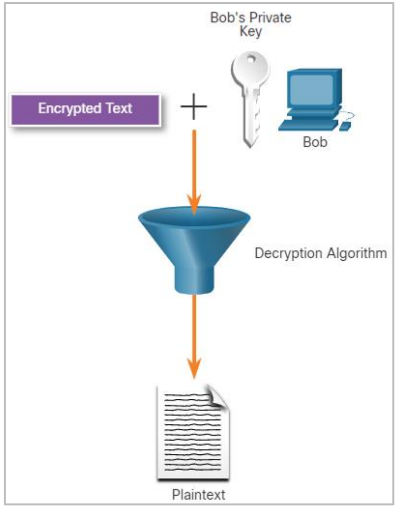
Alice uses Bob’s Public Key



Alice encrypts a hash using her private key



Bob uses Alice’s public key to decrypt the hash



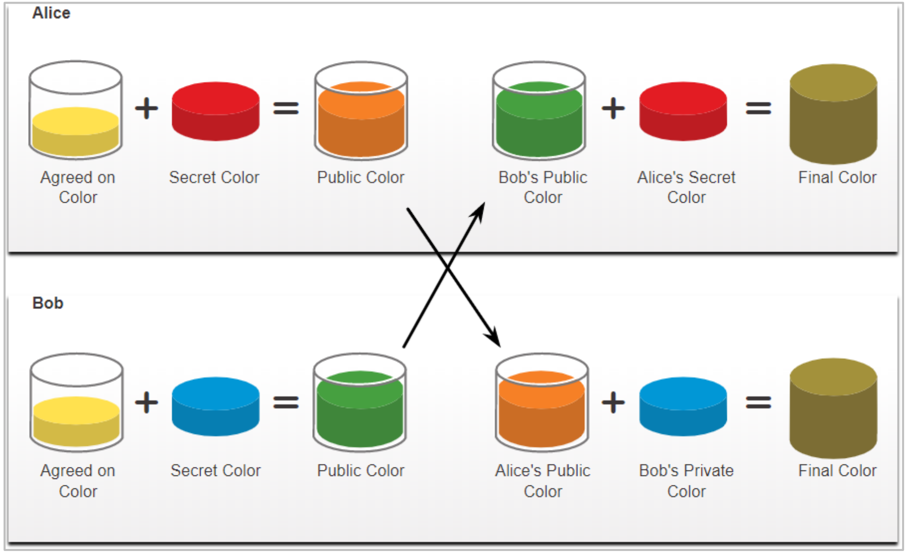
Bob uses his private key to decrypt the message

Confidentiality

# Diffie-Hellman

- Diffie-Hellman (DH) is an asymmetric mathematical algorithm that allows two computers to generate an identical shared secret without having communicated before.

- The new shared key is never actually exchanged between the sender and receiver.
- The key can be used by an encryption algorithm to encrypt traffic between the two systems as both parties know it.
- Following are two examples of instances when DH is commonly used:
  - Data is exchanged using an IPsec VPN
  - SSH data is exchanged
- The security of DH is based on the fact that it uses very large numbers in its calculations.



DH operation

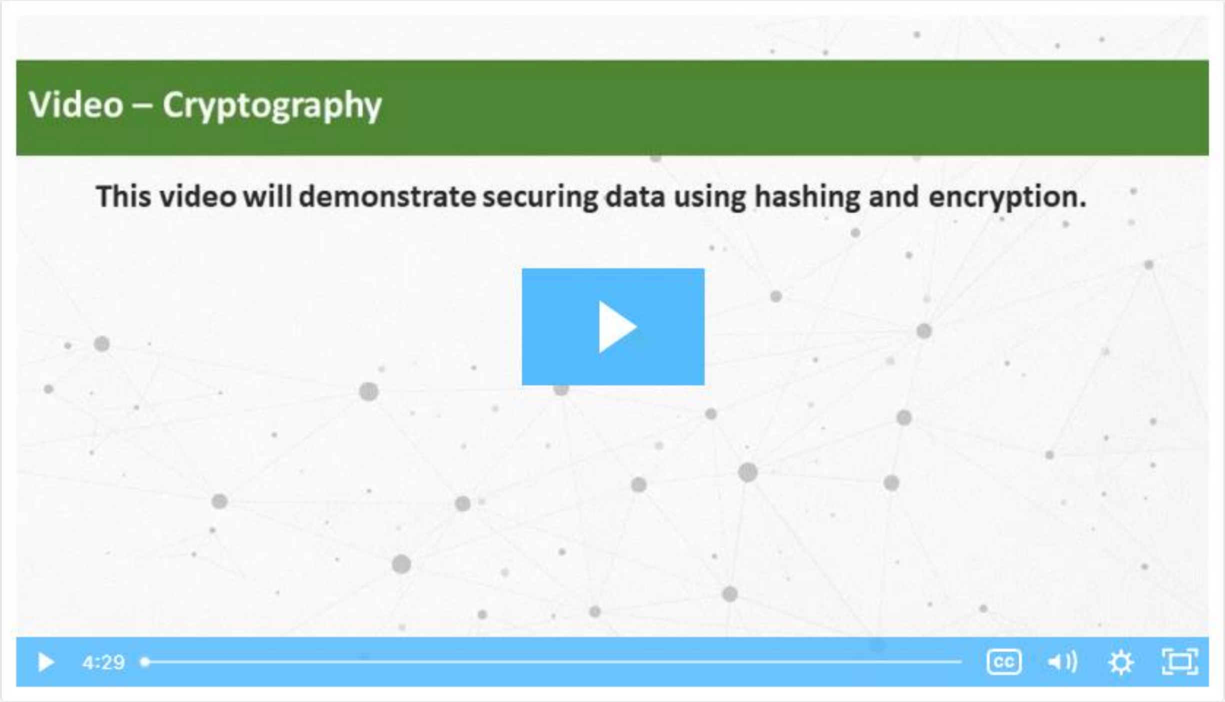
# Diffie-Hellman (Contd.)

- Diffie-Hellman uses different DH groups to determine the strength of the key that is used in the key agreement process. The higher group numbers are more secure, but require additional time to compute the key.
- The following identifies the DH groups supported by Cisco IOS Software and their associated prime number value:
  - DH Group 1: 768 bits
  - DH Group 2: 1024 bits
  - DH Group 5: 1536 bits
  - DH Group 14: 2048 bits
  - DH Group 15: 3072 bits
  - DH Group 16: 4096 bits

***Note**: A DH key agreement can also be based on elliptic curve cryptography. DH groups 19, 20, and 24, are supported by Cisco IOS Software.*

# Video - Cryptography

Watch the video to learn more about Cryptography.



Video – Cryptography

This video will demonstrate securing data using hashing and encryption.

# Lab - Encrypting and Decrypting Data Using OpenSSL

In this lab, you will complete the following objectives:

- Encrypting Messages with OpenSSL
- Decrypting Messages with OpenSSL

# Lab - Encrypting and Decrypting Data Using a Hacker Tool

In this lab, you will complete the following objectives:

- Setup Scenario
- Create and Encrypt Files
- Recover Encrypted Zip File Passwords

Confidentiality

# Lab - Examining Telnet and SSH in Wireshark

In this lab, you will complete the following objectives:

- Examine a Telnet Session with Wireshark
- Examine an SSH Session with Wireshark

# 21.3 Public Key Cryptography

# Using Digital Signatures

- Digital signatures are a mathematical technique used to provide authenticity, integrity, and nonrepudiation.
- Digital signatures use asymmetric cryptography.
- Digital signatures are commonly used in the following two situations:
  - **Code signing -** Code signing is used to verify the integrity of executable files downloaded from a vendor website. It also uses signed digital certificates to authenticate and verify the identity of the site that is the source of the files.
  - **Digital certificates -** These are used to authenticate the identity of a system with a vendor website and establish an encrypted connection to exchange confidential data.
- The Digital Signature Standard (DSS) algorithms used for generating and verifying digital signatures are:
  - **Digital Signature Algorithm (DSA)**
  - **Rivest-Shamir Adelman Algorithm (RSA)**
  - **Elliptic Curve Digital Signature Algorithm (ECDSA)**

# Digital Signatures for Code Signing

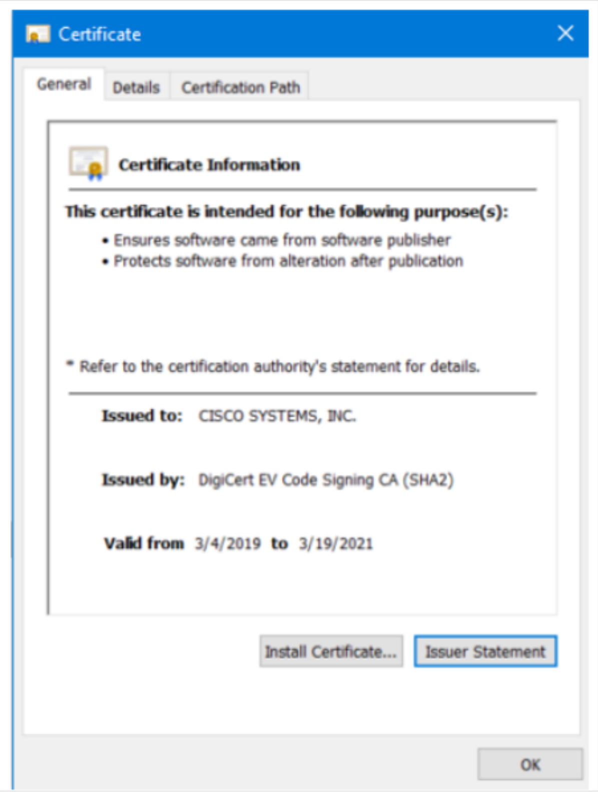
- Digital signatures are commonly used to provide assurance of the authenticity and integrity of software code.
- Executable files are wrapped in a digitally signed envelope, which allows the end user to verify the signature before installing the software.
- Digitally signing code provides several assurances about the code:
  - The code is authentic and is actually sourced by the publisher.
  - The code has not been modified since it left the software publisher.
  - The publisher undeniably published the code. This provides nonrepudiation of the act of publishing.
- The purpose of digitally signed software is to ensure that the software has not been tampered with, and that it originated from the trusted source as claimed.

## Public Key Cryptography

# Digital Signatures for Code Signing (Contd.)

The properties of a file that has a digitally signed certificate are as follows:

| Properties | Description |
|---|---|
| File Properties | This executable file was downloaded from the internet and it contains a software tool from Cisco Systems. |
| Digital Signatures | This tab reveals that the file is from a trusted organization, Cisco Systems Inc. |
| Digital Signatures Details | This window reveals that the file was signed by Cisco Systems, Inc mentioning the given year, month and time. |
| Certificate Information | The **General** tab provides information such as who the certificate was issued to, and who issued the certificate. It also displays the  period for which the certificate is valid. |
| Certificate Path | In this tab, you can see the file was signed by Cisco Systems, as verified to DigiCert. |



**Certificate Information**

**This certificate is intended for the following purpose(s):**

- Ensures software came from software publisher
- Protects software from alteration after publication

* Refer to the certification authority's statement for details.

**Issued to:** CISCO SYSTEMS, INC.

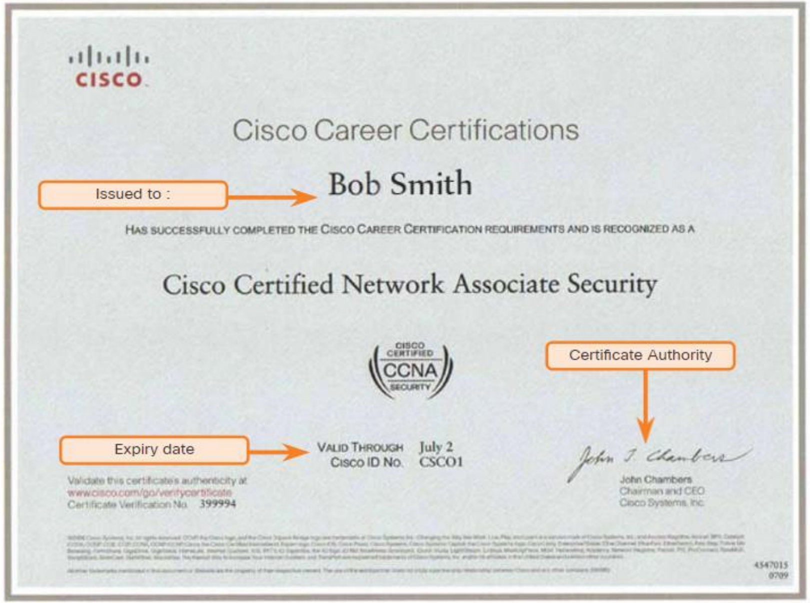
**Issued by:** DigiCert EV Code Signing CA (SHA2)

**Valid from** 3/4/2019 **to** 3/19/2021

Public Key Cryptography

# Digital Signatures for Digital Certificates

- A digital certificate enables users, hosts, and organizations to securely exchange information over the Internet.
- It is used to authenticate and verify that a user who is sending a message is who they claim to be.
- Digital certificates can also be used to provide confidentiality for the receiver with the means to encrypt a reply.
- Digital certificates are similar to physical certificates.
- Digital certificate independently verifies an identity.
- In other words, a certificate verifies an identity, a signature verifies information coming from an identity.


Issued to :
Expiry date
Certificate Authority

cisco

Cisco Career Certifications

Bob Smith

HAS SUCCESSFULLY COMPLETED THE CISCO CAREER CERTIFICATION REQUIREMENTS AND IS RECOGNIZED AS A

Cisco Certified Network Associate Security

CISCO CERTIFIED CCNA SECURITY

VALID THROUGH July 2
CISCO ID NO. CSCO1

Validate this certificate's authenticity at
www.cisco.com/go/verifycertificate
Certificate Verification No. 399994

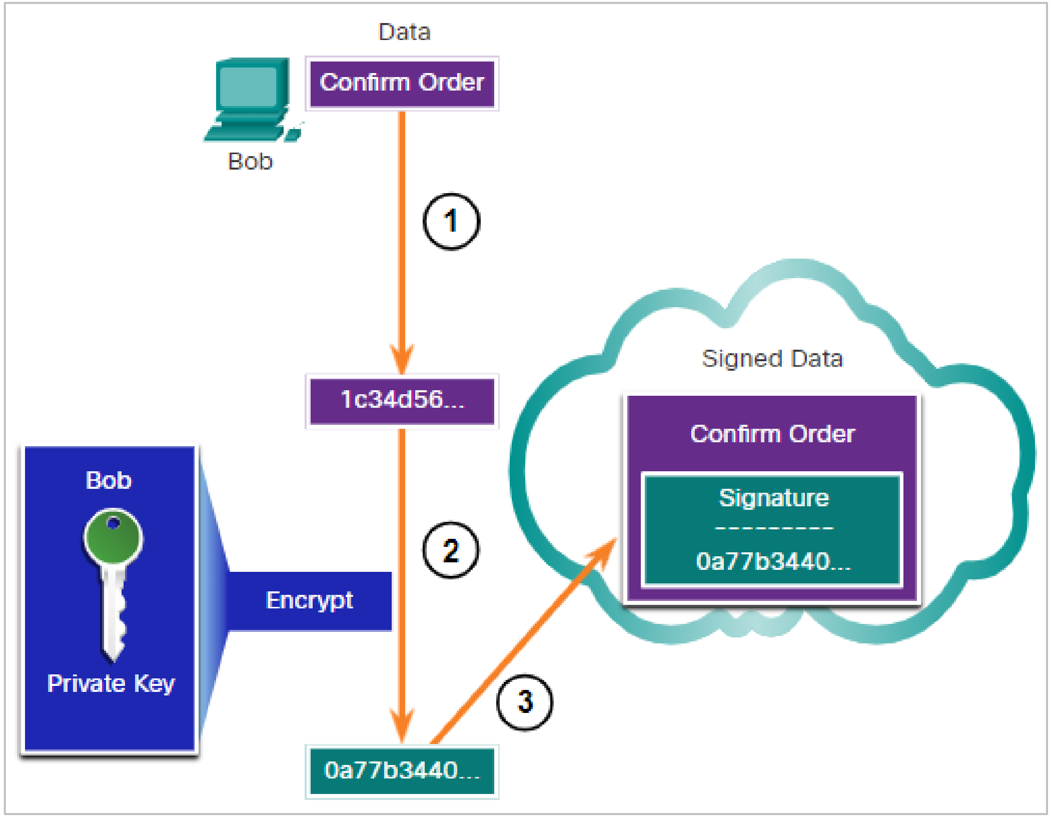
John T. Chambers
John Chambers
Chairman and CEO
Cisco Systems, Inc.

4547015
0709

# Digital Signatures for Digital Certificates (Contd.)

This scenario will help you understand how a digital signature is used.

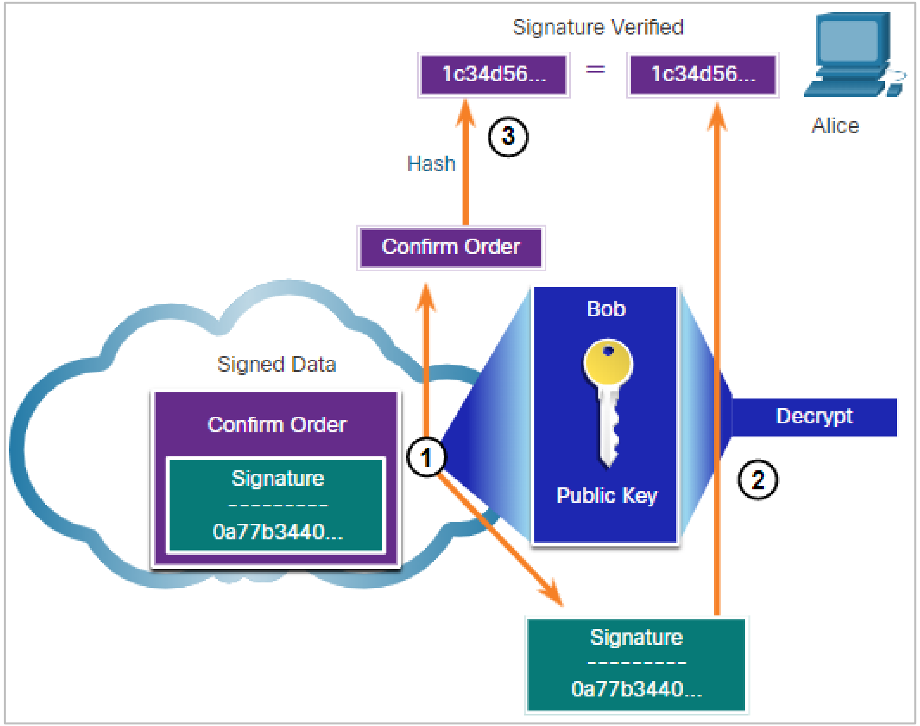
- Bob is confirming an order with Alice, which she is ordering from Bob's website.
- Bob confirms the order and his computer creates a hash of the confirmation.
- The computer encrypts the hash with Bob's private key.
- The encrypted hash, which is the digital signature, is added to the document.
- The order confirmation is then sent to Alice over the internet.



# Digital Signatures for Digital Certificates (Contd.)

When Alice receives the digital signature, the following process occurs:

- Alice's receiver accepts the order confirmation with the digital signature and obtains Bob's public key.
- Alice's computer then decrypts the signature using Bob's public key which reveals the assumed hash value of the sending device.
- Alice's computer creates a hash of the received document, without its signature, and compares this hash to the decrypted hash.
- If the hashes match, the document is authentic. This means the confirmation was sent by Bob and has not changed since signed.

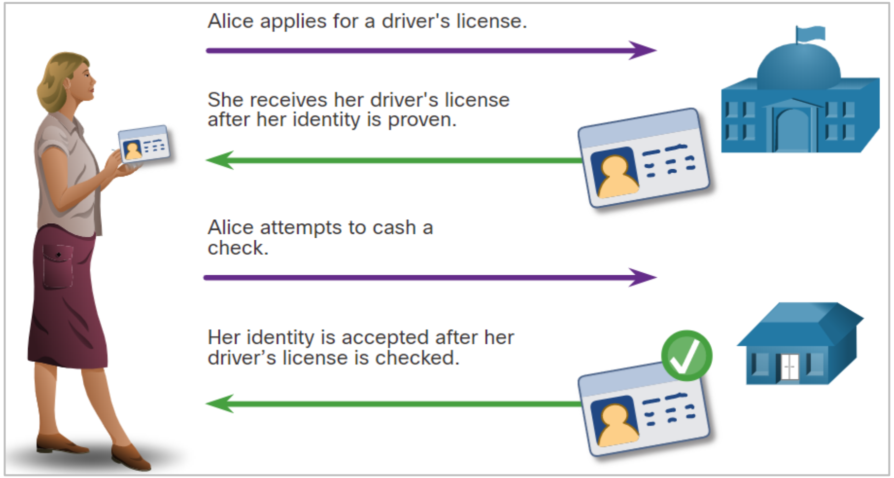


# 21.4 Authorities and the PKI Trust System

Authorities and the PKI Trust System

# Public Key Management

- When establishing an asymmetric connection between two hosts, the hosts will exchange their public key information.
- Trusted third parties on the Internet validate the authenticity of these public keys using digital certificates. The third-party issues credentials that are difficult to forge.
- From that point forward, all individuals who trust the third party simply accept the credentials that the third-party issues.
- The Public Key Infrastructure (PKI) consists of specifications, systems, and tools that are used to create, manage, distribute, use, store, and revoke digital certificates.


Alice applies for a driver's license.
She receives her driver's license after her identity is proven.
Alice attempts to cash a check.
Her identity is accepted after her driver's license is checked.

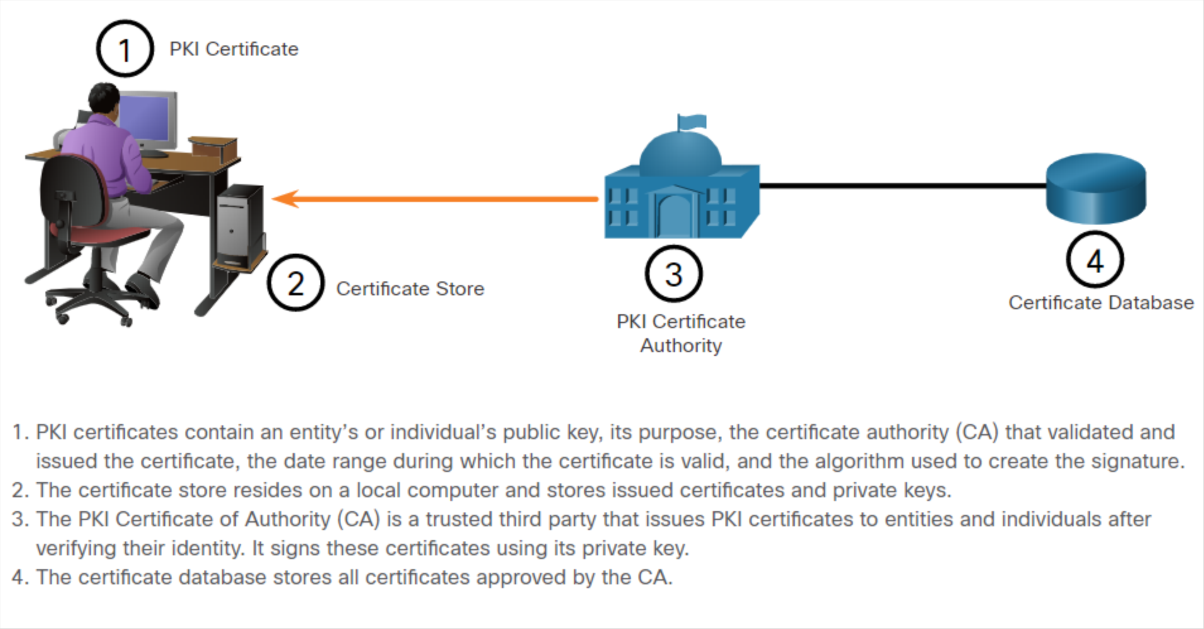
Illustrates how a driver’s license is analogous to a digital certificate

- The Certificate Authority (CA) creates digital certificates by tying a public key to a confirmed identify, such as a website or individual.

## Authorities and the PKI Trust System

# The Public Key Infrastructure

- PKI is needed to support large-scale distribution and identification of public encryption keys.
- The PKI framework facilitates a highly scalable trust relationship.
- It consists of the hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates.
- The figure shows the main elements of the PKI.

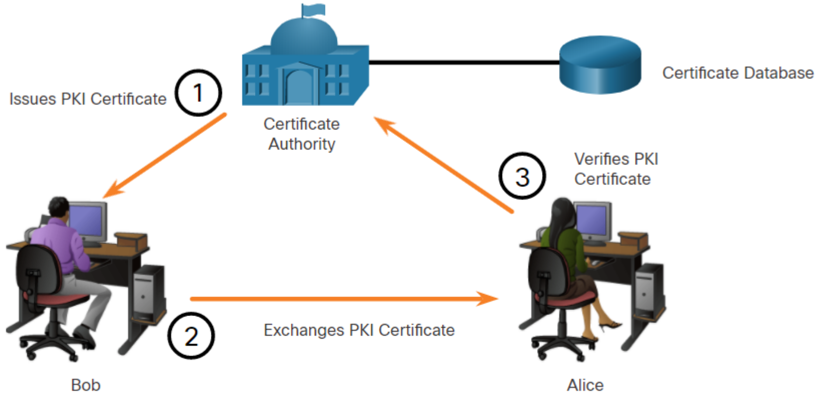


1. PKI certificates contain an entity's or individual's public key, its purpose, the certificate authority (CA) that validated and issued the certificate, the date range during which the certificate is valid, and the algorithm used to create the signature.
2. The certificate store resides on a local computer and stores issued certificates and private keys.
3. The PKI Certificate of Authority (CA) is a trusted third party that issues PKI certificates to entities and individuals after verifying their identity. It signs these certificates using its private key.
4. The certificate database stores all certificates approved by the CA.

## Authorities and the PKI Trust System

# The Public Key Infrastructure (Contd.)

- The below figure shows how the elements of the PKI interoperate:



1. **Issues PKI Certificate**. Bob initially requests a certificate from the CA. The CA authenticates Bob and stores Bob's PKI certificate in the certificate database.
2. **Exchanges PKI Certificate**. Bob communicates with Alice using his PKI certificate.
3. **Verifies PKI Certificate**. Alice communicates with the trusted CA using the CA's public key. The CA refers to the certificate database to validate Bob's PKI certificate.

***Note****: Not all PKI certificates are directly received from a CA. A Registration Authority (RA) is a subordinate CA and is certified by a root CA to issue certificates for specific uses.*

Authorities and the PKI Trust System

# The PKI Authorities System

- Many vendors provide CA servers as a managed service or as an end-user product.
- Organizations may also implement private PKIs using Microsoft Server or Open SSL.
- CAs issue certificates based on classes which determine how trusted a certificate is.
- The class number is determined by how rigorous the procedure was that verified the identity of the holder when the certificate was issued.
- The higher the class number, the more trusted the certificate.
- Some CA public keys are preloaded, such as those listed in web browsers.

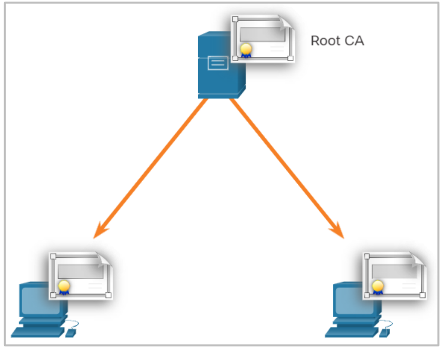
| Class | Description |
|---|---|
| 0 | Used for testing in situations in which no checks have been performed. |
| 1 | Used by individuals who require verification of email. |
| 2 | Used by organizations for which proof of identity is required. |
| 3 | Used for servers and software signing. |
| 4 | Used for online business transactions between companies. |
| 5 | Used for private organizations or government security. |

***Note***: *An enterprise can also implement PKI for internal use. PKI can be used to authenticate employees who are accessing the network. In this case, the enterprise is its own CA.*

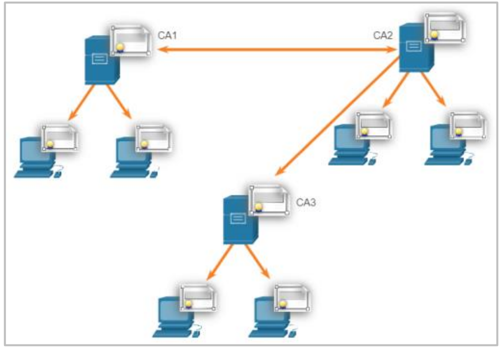
Authorities and the PKI Trust System

# The PKI Trust System

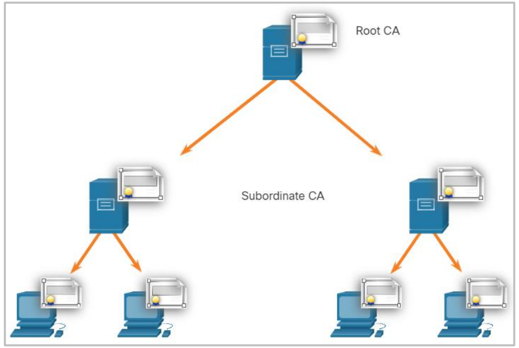
- PKIs can form different topologies of trust which are as follows:
  - **Single-Root PKI Topology:** The simplest is the single-root PKI topology. The root CA issues all the certificates to the end users within the same organization. On larger networks, PKI CAs may be linked using two basic architectures:
    - **Cross-certified CA topologies**: A peer-to-peer model in which individual CAs establish trust relationships with other CAs by cross-certifying CA certificates.
    - **Hierarchical CA topologies**: The root CA (highest level CA), can issue certificates to end users and to a subordinate CA.



Single-Root PKI Topology



Cross-certified CA Topologies



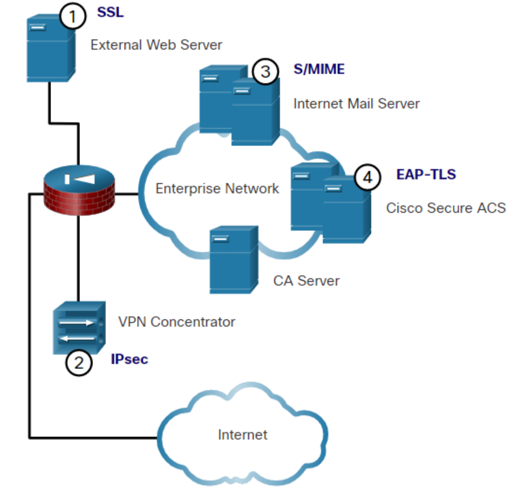
Hierarchical CA Topologies

## Authorities and the PKI Trust System

# Interoperability of Different PKI Vendors

- Interoperability between a PKI and its supporting services is a concern because many CA vendors have proposed and implemented proprietary solutions.
- To address this interoperability concern, the IETF published the Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework (RFC 2527).
- The X.509 version 3 (X.509 v3) standard defines the format of a digital certificate.

***Note**: LDAP and X.500 are protocols that are used to query a directory service, such as Microsoft Active Directory, to verify a username and password.*



1. **SSL** - Secure web servers user X.509.v3 for website authentication in the SSL and TLS protocols, while web browsers use X.509v3 to implement HTTPS client certificates. SSL is the most widely used certificate-based authentication.
2. **IPsec** - IPsec VPNs use X.509 when certificates can be used as a public key distribution mechanism for internet key exchange (IKE) RSA-based authentication.
3. **S/MIME** - User mail agents that support mail protection with the Secure/Multipurpose Internet Mail Extensions (S/MIME) protocol use X.509 certificates.
4. **EAP-TLS** - Cisco switches can use certificates to authenticate end devices that connect to LAN ports using 802.1.x between the adjacent devices. The authentication can be proxied to a central ACS via the Extensible Authentication Protocol with TLS (EAP-TLS).

X.509v3 Applications

# Certificate Enrollment, Authentication and Revocation

- All systems that leverage the PKI must have the CA’s public key, which is called the self-signed certificate.
- The CA public key verifies all the certificates issued by the CA and is vital for the proper operation of the PKI.
- The certificate enrollment process is used by a host system to enroll with a PKI. To do so, CA certificates are retrieved in-band over a network, and the authentication is done out-of-band (OOB) using the telephone.
- The system enrolling with the PKI contacts a CA to request and obtain a digital identity certificate for itself and to get the CA’s self-signed certificate.
- The final stage verifies that the CA certificate was authentic and is performed using an out-of-band method such as the POTS to obtain the fingerprint of the valid CA identity certificate.
- A digital certificate can be revoked if key is compromised or if it is no longer needed.

***Note**: Only a root CA can issue a self-signed certificate that is recognized or verified by other CAs within the PKI.*

# Lab – Certificate Authority Stores

In this lab, you will complete the following objectives:

- Certificates Trusted by Your Browser
- Checking for Man-In-Middle

# 21.5 Applications and Impacts of Cryptography

# PKI Applications

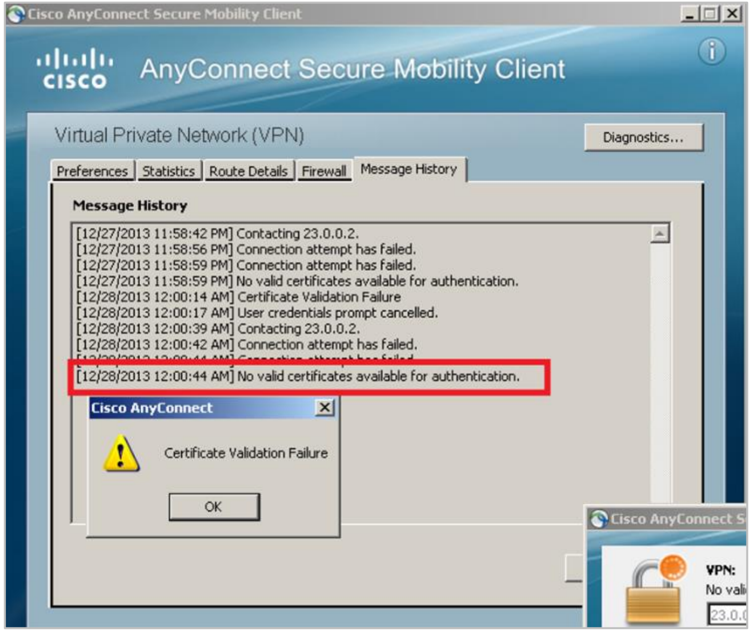
The following provides a short list of common uses of PKIs:

- SSL/TLS certificate-based peer authentication
- Secure network traffic using IPsec VPNs
- HTTPS Web traffic
- Control access to the network using 802.1x authentication
- Secure email using the S/MIME protocol
- Secure instant messaging
- Approve and authorize applications with Code Signing
- Protect user data with the Encryption File System (EFS)
- Implement two-factor authentication with smart cards
- Securing USB storage devices

## Applications and Impacts of Cryptography

# Encrypted Network Transactions

- Threat actors can use SSL/TLS to introduce regulatory compliance violations, viruses, malware, data loss, and intrusion attempts in a network.
- Other SSL/TLS-related issues may be associated with validating the certificate of a web server. When this occurs, the web browsers will display a security warning. PKI-related issues associated with security warnings include:
  - **Validity date range** - The X.509v3 certificates specify "not before" and "not after" dates. If the current date is outside the range, the web browser displays a message.
  - **Signature validation error** - If a browser cannot validate the signature on the certificate, there is no assurance that the public key in the certificate is authentic.



Message History

```
[12/27/2013 11:58:42 PM] Contacting 23.0.0.2.
[12/27/2013 11:58:56 PM] Connection attempt has failed.
[12/27/2013 11:58:59 PM] Connection attempt has failed.
[12/27/2013 11:58:59 PM] No valid certificates available for authentication.
[12/28/2013 12:00:14 AM] Certificate Validation Failure
[12/28/2013 12:00:17 AM] User credentials prompt cancelled.
[12/28/2013 12:00:39 AM] Contacting 23.0.0.2.
[12/28/2013 12:00:42 AM] Connection attempt has failed.
[12/28/2013 12:00:44 AM] Connection attempt has failed.
[12/28/2013 12:00:44 AM] No valid certificates available for authentication.
```

Certificate Validation Failure

# Encryption and Security Monitoring

- Network monitoring becomes more challenging when packets are encrypted.
- As HTTPS introduces end-to-end encrypted HTTP traffic (via TLS/SSL), it is not as easy to peek into user traffic.
- Security analysts must know how to circumvent and solve these issues. Here is a list of some of the things that a security analyst could do:
  - Configure rules to distinguish between SSL and non-SSL traffic, HTTPS and non-HTTPS SSL traffic.
  - Enhance security through server certificate validation using CRLs and OCSP.
  - Implement antimalware protection and URL filtering of HTTPS content.
  - Deploy a Cisco SSL Appliance to decrypt SSL traffic and send it to intrusion prevention system (IPS) appliances to identify risks normally hidden by SSL.

Applications and Impacts of Cryptography

# Encryption and Security Monitoring (Contd.)

- Cryptography is dynamic and always changing. A security analyst must maintain a good understanding of cryptographic algorithms and operations to be able to investigate cryptography-related security incidents.
- There are two main ways in which cryptography impacts security investigations.
- First, attacks can be directed to specifically target the encryption algorithms themselves.
- After the algorithm has been cracked and the attacker has obtained the keys, any encrypted data that has been captured can be decrypted by the attacker and read, thus exposing private data.
- Secondly, the security investigation is also affected because data can be hidden in plain sight by encrypting it.

# 21.6 Public Key Cryptography Summary

Public Key Cryptography Summary

# What Did I Learn in this Module?

- The four elements of secure communications: data integrity, origin authentication, data confidentiality, and data non-repudiation.
- A hash function takes a variable block of binary data, called the message, and produces a fixed-length, condensed representation, called the hash.
- There are two classes of encryption that are used to provide data confidentiality: asymmetric and symmetric.
- Symmetric encryption algorithms, such as DES, 3 DES, and AES are based on the premise that each communicating party knows the pre-shared key.
- Asymmetric algorithms (public key algorithms) are designed so that the key that is used for encryption is different from the key used for encryption.
- Data confidentiality can also be ensured using asymmetric algorithms, including Rivest, Shamir, and Aldeman (RSA) and PKI. The process is summarized using this formula: Public key (Encrypt) + Private Key (Decrypt) = Confidentiality.

# What Did I Learn in this Module? (Contd.)

- The authentication objective of an asymmetric algorithm is initiated when the encryption process is started with the private key. The process can be summarized with this formula: Private Key (Encrypt) + Public Key (Decrypt) = Authentication.
- Diffie-Hellman (DH) is an asymmetric mathematical equation algorithm that allows two computers to generate an identical shared secret key without having communicate before.
- Digital signatures are a mathematical technique used to provide three basic security services: authenticity, integrity, and non-repudiation. Digital signatures are commonly used in code signing and digital certificates.
- The Public Key Infrastructure (PKI) consists of specifications, systems, and tools that are used to create, manage, distribute, use, store, and revoke digital certificates.
- There are many common uses of PKIs including a few listed here: SSL/TLS certificate-based peer authentication, HTTPS Web traffic, secure instant message, and securing USB storage devices.
- A security analyst must be able to recognize and solve potential problems related to permitting PHI-related solutions on the enterprise network.

cisco