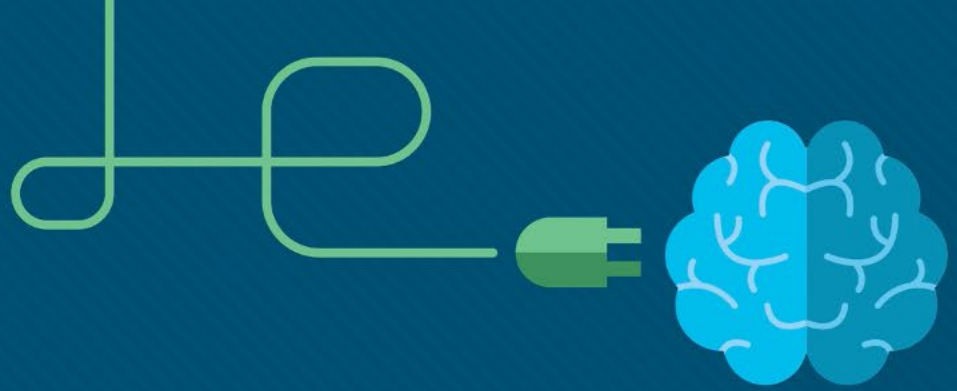


Module 18: Cryptography

Cybersecurity Essentials 3.0

Module Objectives

Module Title: Cryptography

Module Objective: Explain how to use hashing tools.

Topic Title	Topic Objective
Confidentiality	Determine the encryption algorithm to use according to requirements.
Obscuring Data	Use a technique to obscure data.
Integrity and Authenticity	Explain the role of cryptography in ensuring the integrity and authenticity of data.
Using Hashes	Use hashing tools to hash a text file and to verify the integrity of a different file.
Public Key Cryptography	Use a digital signature.
Authorities and the PKI Trust System	Use hashing to detect network interception.
Applications and Impacts of Cryptography	Explain how the use of cryptography affects cybersecurity operations.

18.1 Confidentiality

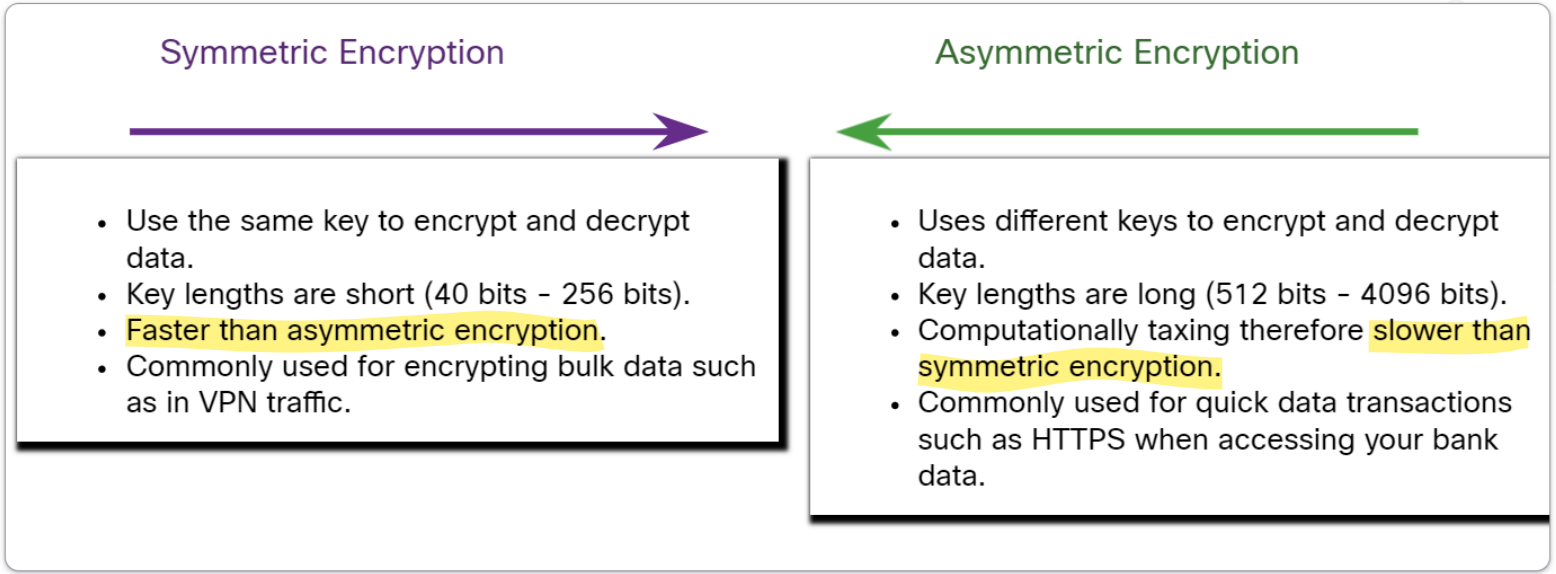
Data Confidentiality

- 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).

Note: DES is a legacy algorithm and should not be used. 3DES should be avoided if possible.

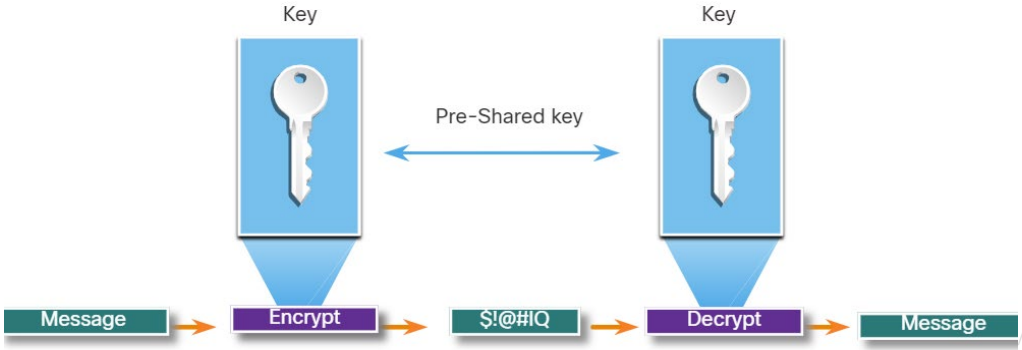
Data Confidentiality (Cont.)

- The figure highlights some differences between symmetric and asymmetric encryption.



Symmetric Encryption

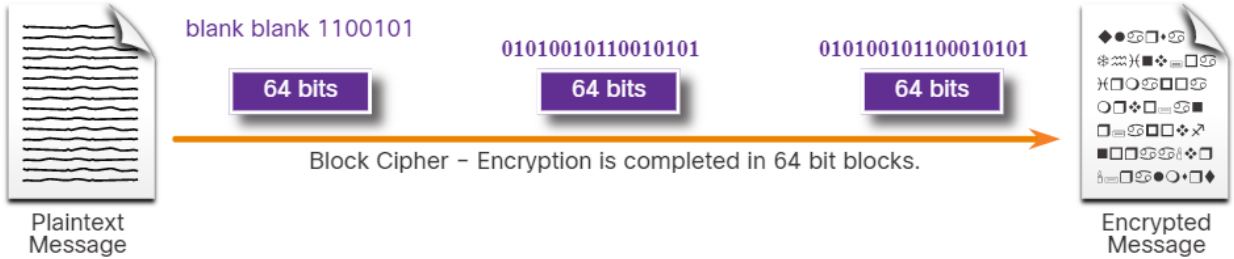
- Symmetric algorithms use the same pre-shared key to encrypt and decrypt data.
- A pre-shared key (secret key), is known by the sender and receiver before any encrypted communications can take place.
- Symmetric encryption algorithms are commonly used with VPN traffic, because they use less CPU resources than asymmetric encryption algorithms, allowing the encryption and decryption of data to be fast when using a VPN.
- When using symmetric encryption algorithms, like any other type of encryption, the longer the key, the longer it will take for someone to discover the key.
- Most encryption keys are between 112 and 256 bits but to ensure that the encryption is safe, a minimum key length of 128 bits should be used.



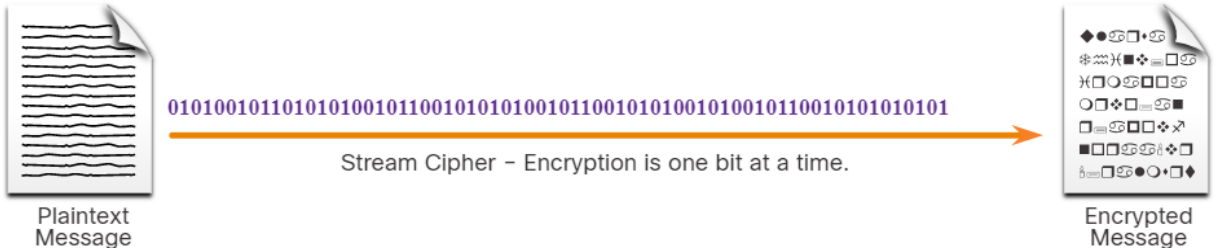
Symmetric Encryption (Cont.)

Symmetric encryption algorithms are sometimes classified as either a block cipher or a stream cipher:

- **Block ciphers** transform a fixed-length block of plaintext into a common block of ciphertext of 64 or 128 bits.



- **Stream ciphers** encrypt plaintext one byte or one bit at a time and are typically faster than block ciphers because data is continuously encrypted.



Symmetric Encryption (Cont.)

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

Symmetric Encryption Algorithms	Description
Data Encryption Standard (DES)	A legacy symmetric encryption algorithm. It uses a short key length that makes it insecure for most current uses.
3DES (Triple DES)	The replacement for DES and repeats the DES algorithm process three times. It should be avoided if possible as it is scheduled to be retired in 2023. If implemented, use very short key lifetimes.
Advanced Encryption Standard (AES)	A popular and recommended symmetric encryption algorithm. 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)	A faster alternative symmetric encryption algorithm to AES. It is a stream cypher 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	Several variations have been developed, but RC4 was the most prevalent in use. 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 (public-key algorithms), are designed so that the key used for encryption is different from the key used for decryption.
- The decryption key cannot, in any reasonable amount of time, be calculated from the encryption key and vice versa.

- Asymmetric algorithms use a public key and a private key, and both keys are capable of the encryption process, but the complementary paired key is required for decryption.
- The process is also reversible, so the 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 (Cont.)

- Asymmetric encryption can use key lengths between 512 to 4,096 bits.

- Key lengths greater than or equal to 2,048 bits can be trusted, while key lengths of 1,024 or shorter are considered insufficient.

- Examples of protocols that use asymmetric key algorithms include:
  - **Internet Key Exchange (IKE)** - A fundamental component of IPsec VPNs.
  - **Secure Socket Layer (SSL)** - It is implemented now as IETF standard Transport Layer Security (TLS).
  - **Secure Shell (SSH)** - It provides a secure remote access connection to network devices.
  - **Pretty Good Privacy (PGP)** - It provides cryptographic privacy and authentication.

- Asymmetric algorithms are substantially slower than symmetric algorithms, so they are typically used in low-volume cryptographic mechanisms, such as digital signatures and key exchange.

- However, the key management of asymmetric algorithms tends to be simpler than symmetric algorithms, because usually one of the two encryption or decryption keys can be made public.

Asymmetric Encryption (Cont.)

Asymmetric Encryption Algorithm	Key Length	Description
Diffie-Hellman (DH)	512, 1024, 2048, 3072, 4096	It allows two parties to agree on a key that they can use to encrypt messages they want to send to each other. The security of this algorithm 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	DSS specifies DSA as the algorithm for digital signatures. DSA is a public key algorithm based on the ElGamal signature scheme. Signature creation speed is like RSA but is 10 to 40 times slower for verification.
Rivest, Shamir, and Adleman encryption algorithms (RSA)	512 to 2048	RSA is for public-key cryptography based on the current difficulty of factoring very large numbers. It is believed to be secure given sufficiently long keys and the use of up-to-date implementations.
ElGamal	512 - 1024	Used for public-key cryptography based on the DH key agreement. Its disadvantage is that the encrypted message becomes very big, about twice the size of the original message and for this reason it is only used for small messages such as secret keys.
Elliptic curve techniques	224 or higher	It can be used to adapt many cryptographic algorithms (DH or ElGamal).

Asymmetric Encryption - Confidentiality

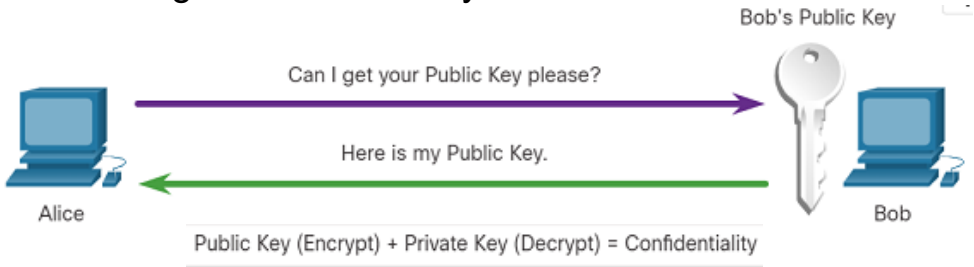
- Asymmetric algorithms are used to provide confidentiality without pre-sharing a password.
- The confidentiality objective is initiated when the encryption process is started with the public key.
- The process can be summarized using the formula:

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

- When the public key is used to encrypt the data, the private key must be used to decrypt the data.
- Only one host has the private key; therefore, confidentiality is achieved.
- If the private key is compromised, another key pair must be generated to replace the compromised key.

Example: Bob and Alice Data Exchange - Confidentiality

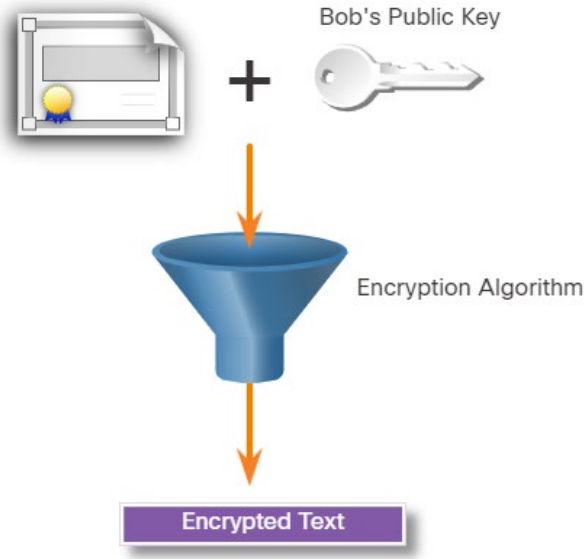
- Alice requests and obtains Bob's public key.



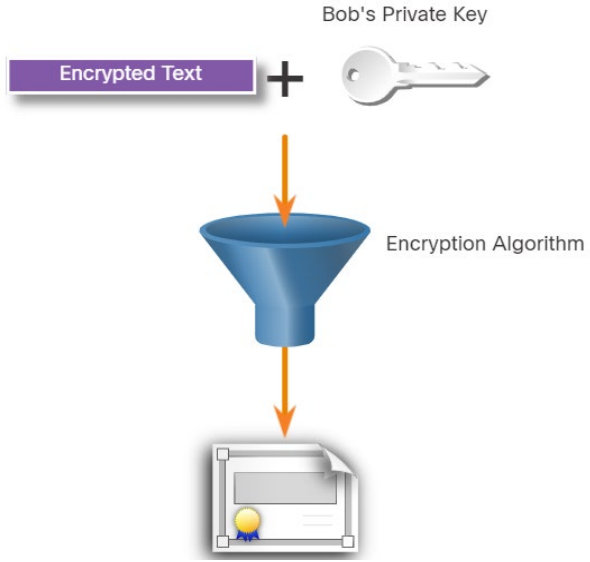
Asymmetric Encryption – Confidentiality (Cont.)

Example: Bob and Alice Data Exchange - Confidentiality

- Alice uses Bob's public key to encrypt a message using an agreed-upon algorithm.
- Alice sends the encrypted message to Bob.



- Bob then uses his private key to decrypt the message.
- Bob is the only one with the private key, so Alice's message can only be decrypted by Bob and thus confidentiality is achieved.



Asymmetric Encryption - Authentication

- The authentication objective of asymmetric algorithms is initiated when the encryption process is started with the private key.
- The process can be summarized using the formula:

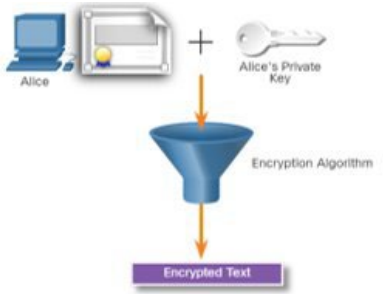
Private Key (Encrypt) + Public Key (Decrypt) = Authentication

- When the private key is used to encrypt the data, the corresponding public key must be used to decrypt it.
- Only one host has the private key, so 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.

Asymmetric Encryption – Authentication (Cont.)

Example: Bob and Alice Data Exchange - Authentication

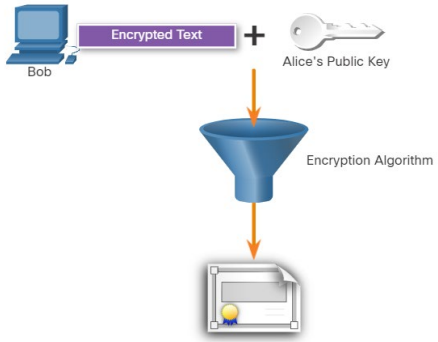
- Alice encrypts a message using her private key.
- Alice sends the encrypted message to Bob.
- Bob needs to authenticate that the message came from Alice.



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

- Bob requests Alice's public key to authenticate the message.

Bob needs to verify that the message actually came from Alice. He requests and acquires Alice's public key.



- Bob uses Alice's public key to decrypt the message.

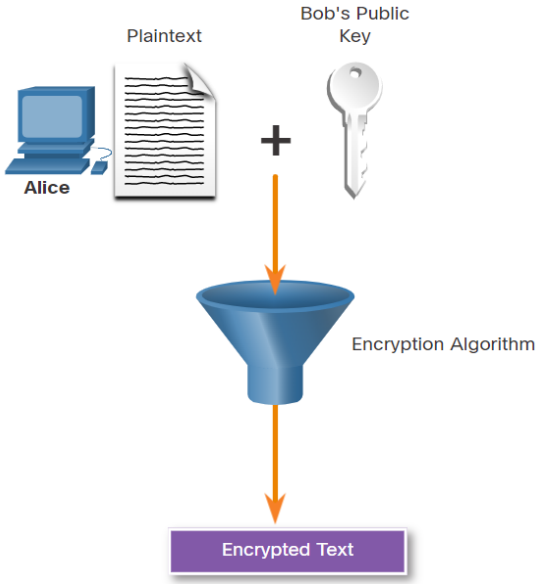
Bob uses the public key to successfully decrypt the message and authenticate that the message did, indeed, come from Alice.

Asymmetric Encryption - Integrity

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

Example: Bob and Alice Data Exchange - Integrity

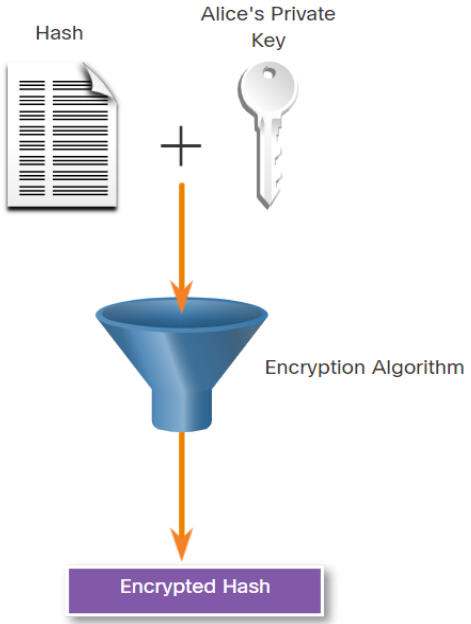
- Alice wants to send a message to Bob ensuring that only Bob can read the document.
- Alice wants to ensure message confidentiality.
- Alice uses the public key of Bob to cipher the message.
- Only Bob will be able to decipher it using his private key.



Asymmetric Encryption – Integrity (Cont.)

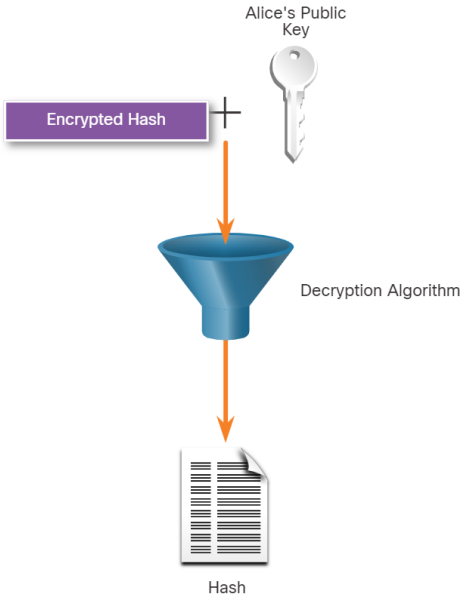
Example: Bob and Alice Data Exchange - Integrity

- Alice also wants to ensure message authentication and integrity.
- Authentication ensures Bob that the document was sent by Alice, and integrity ensures that it was not modified.
- Alice uses her private key to cipher a hash of the message.
- Alice sends the encrypted message with its encrypted hash to Bob.



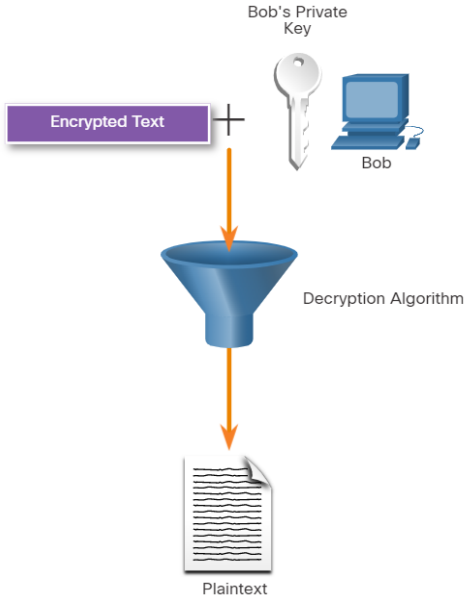
Asymmetric Encryption – Integrity (Cont.)

Example: Bob and Alice Data Exchange - Integrity



- Bob uses Alice's public key to verify that the message was not modified.
- The received hash is equal to the locally determined hash based on Alice's public key.
- This verifies that Alice is the sender of the message because nobody else has Alice's private key.

- Bob uses his private key to decipher the message.

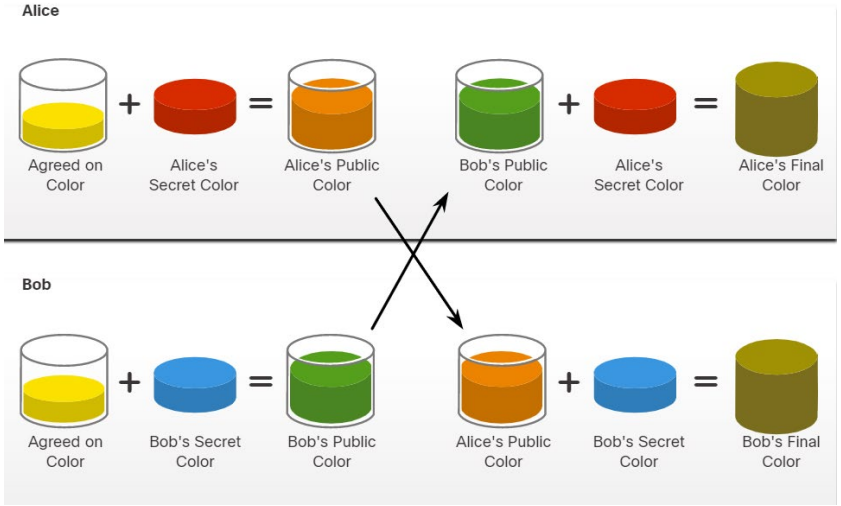


Diffie-Hellman

- 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.
- However, because both parties know it, the key can be used by an encryption algorithm to encrypt traffic between the two systems.
- Here are two examples of instances when DH is commonly used:
  - Data is exchanged using an IPsec VPN
  - SSH data is exchanged

Diffie-Hellman (Cont.)

The colors in the figure will be used instead of complex long numbers to simplify the DH key agreement process.



- The DH key exchange begins with Alice and Bob agreeing on an arbitrary common color that does not need to be kept secret (yellow).
- The chosen secret colors never to be shared (Alice-red, Bob-blue) represent the private key of each party.
- Alice will mix the shared common color of yellow with her red color to produce a public color of orange. Bob will mix the yellow and the blue to produce a public color of green.
- Alice sends her public color (orange) to Bob and Bob sends his public color (green) to Alice.
- Alice and Bob each mix the color they received with their own, original secret color (red for Alice and blue for Bob.). The result is a final brown color mixture that is identical to the partner's final color mixture.
- The brown color represents the resulting shared secret key between Bob and Alice.

Diffie-Hellman (Cont.)

- The security of DH uses very large numbers in its calculations.
- 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

Video - Cryptography

- Watch the video to learn more about Cryptography.

Lab - Use Classic and Modern Encryption Algorithms

- In this lab, you will complete the following objectives:
  - Part 1: Use a Classic Encryption Algorithm.
  - Part 2: Use a Modern Symmetrical Encryption Algorithm.
  - Part 3: Use a Modern Asymmetrical Encryption Algorithm.

Lab - Encrypting and Decrypting Data Using OpenSSL

- In this lab, you will complete the following objectives:
  - Part 1: Encrypting Messages with OpenSSL
  - Part 2: Decrypting Messages with OpenSSL

Lab - Encrypting and Decrypting Data Using a Hacker Tool

- In this lab, you will complete the following objectives:
  - Part 1: Create and Encrypt Files
  - Part 2: Recover Encrypted Zip File Passwords

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

Lab - Determine the Encryption Algorithm to Use

- In this lab, you will complete the following objectives:
  - Part 1: Secure Personal Data
  - Part 2: Secure Wireless Data
  - Part 3: Secure Corporate Data Between Sites

18.2 Obscuring Data

Data Masking Techniques

- Data masking technology secures data by replacing sensitive information with non-sensitive versions of it.
- The non-sensitive version looks and acts like the original so that an organizational process can use non-sensitive data with no change needed to the supporting applications or data storage facilities.
- Masking most commonly limits the propagation of sensitive data within IT systems by distributing surrogate data sets for testing and analysis.
- Information can be dynamically masked on the spot if the system or application detects a risky user request for sensitive information.
- Data masking can replace sensitive data in non-production environments to protect the underlying information. Several data masking techniques can be used.

Data Masking Techniques (Cont.)

The below methods ensure that data remains meaningful but changed enough to protect it.

- **Substitution** replaces data with authentic-looking values to apply anonymity to the data records.
- **Shuffling** derives a substitution set from the same column of data that a user wants to mask.
  - This technique works well for financial information in a test database, for example.
- **Nulling out** applies a null value to a particular field, which completely prevents visibility of the data.

Steganography

- Steganography conceals data (a message) in another file such as a graphic, audio, or video file.
- The advantage of steganography over cryptography is that the secret message does not attract any special attention.
- No one would ever know that a picture contained a secret message if they just viewed the file either electronically or in hard copy form.
- There are several components involved in hiding data:
  - First, there is the embedded data, which is the secret message.
  - The cover-text (or cover-image or cover-audio) hides the embedded data producing the stego text (or stego image or stego audio).
  - A stego key controls the hiding process.

Steganography (Cont.)

Steganography techniques	Used to embed data in a cover image is using least significant bits (LSB). This method uses bits of each pixel in the image. A pixel is the basic unit of programmable color in a computer image. The specific color of each pixel is a blend of three colors (RGB). Three bytes of data specify a pixel's color (one byte for each color). Eight bits make up a byte so a 24-bit color system uses all three bytes. LSB uses a bit of each of the red, green, and blue color components. Each pixel can store three bits.
Social steganography	It hides information in plain sight by creating output that can be read a certain way by some to get the secret message, based on previously set rules and/or definitions. As a result, those who view it in a normal way will not see the message. Teens on social media use this tactic to communicate with their closest friends while keeping others, like their parents, unaware of what the message means. Individuals in countries that censor media also use it to get their messages out by misspelling words on purpose or making obscure references that mean something to those in the know. In effect, they communicate to different audiences at the same time, sending out two different messages: the apparent message and the secret message.
Detection	Steganalysis follows the discovery that hidden information exists. The goal of steganalysis is to discover this hidden information. Patterns in the stego image create suspicion. Disk analysis utilities can report on hidden information in unused clusters of storage devices. Filters can capture data packets that contain hidden information in packet headers. Both methods use steganography signatures. By comparing an original image with the stego image, an analyst may pick up repetitive patterns visually.

Lab - Use Steganography to Hide Data

- In this lab, you will use Steg hide, an open-source steganography program, to hide a data file within an image file.

18.3 Integrity and Authenticity

Securing Communications

- Organizations must provide support to secure internal and external traffic.

- The four elements of secure communications are:
  - **Data Integrity** - Guarantees that the message was not altered, and any changes to data in transit will be detected. Integrity is ensured by implementing either of the SHA-2 or SHA-3. The MD5 message digest algorithm is still in use, but it is insecure and should be avoided.

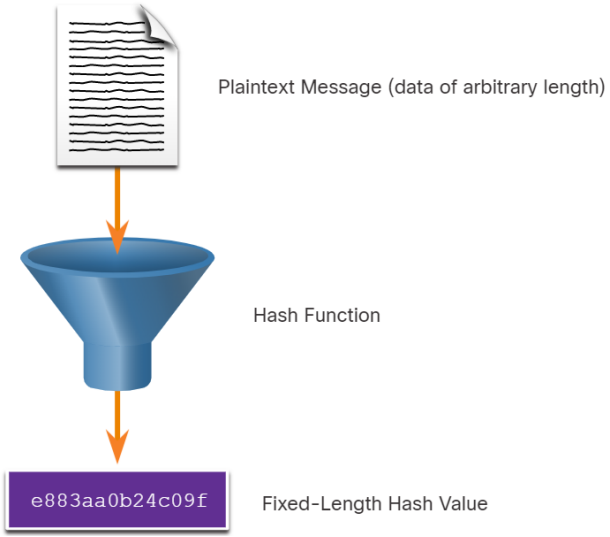
  - **Origin Authentication** - Guarantees that the message is not a forgery and actually comes from whom it states. Many modern networks ensure authentication with algorithms such as hash-based message authentication code (HMAC).

  - **Data Confidentiality** - Guarantees that only authorized users can read the message. If the message is intercepted, it cannot be deciphered within a reasonable amount of time. Data confidentiality is implemented using symmetric and asymmetric encryption algorithms.

  - **Data Non-Repudiation** - Guarantees that the sender cannot repudiate, or refute, the validity of a message sent. Nonrepudiation relies on the fact that only the sender has the unique characteristics or signature for how that message is treated.

Cryptographic Hash Functions

- Hashes are used to verify and ensure data integrity and is based on a one-way mathematical function relatively easy to compute, but significantly harder to reverse.
- The cryptographic hashing function can also be used to verify authentication.
- A hash function takes a variable block of binary data (message) and produces a fixed-length, condensed representation (hash).
- With hash functions, it is computationally infeasible for two different sets of data to produce the same hash output.
- Every time the data is changed or altered; the hash value also changes.
- The cryptographic hash function is applied in many different situations for entity authentication, data integrity, and data authenticity purposes.

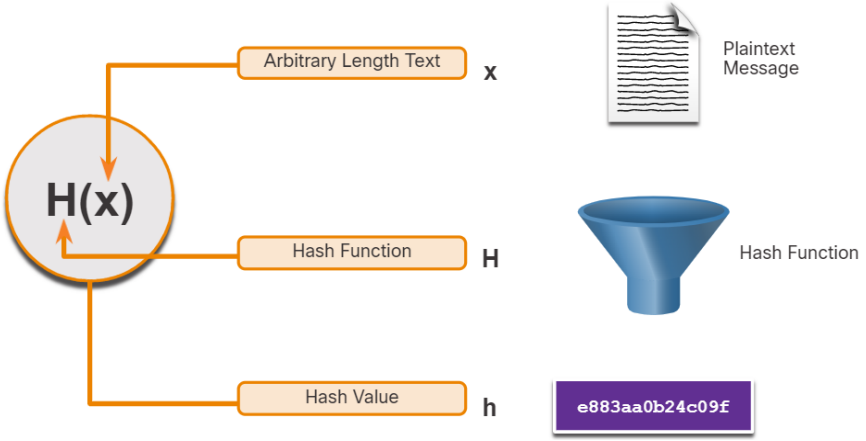


Cryptographic Hash Operation

- 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 any 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.

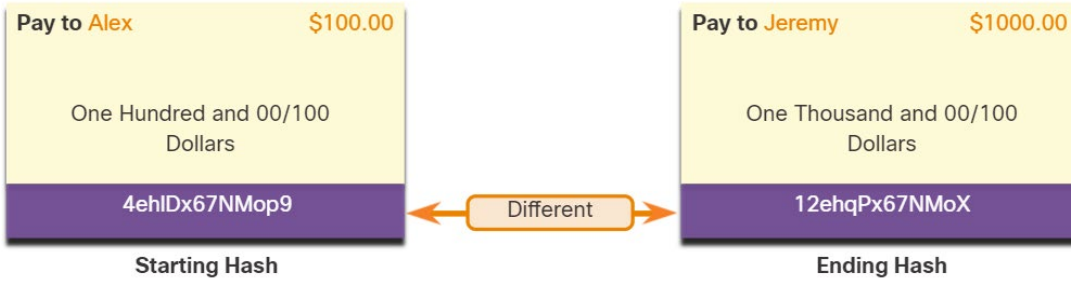
- Hard to invert (one-way hash) means that given a hash value of h, it is computationally infeasible to find an input for x such that h=H(x).



MD5 and SHA

- Hash functions are used to ensure message integrity, that data has not changed accidentally or intentionally.
- Deliberate changes that are made by a threat actor are still possible.

- In the figure, the sender is sending a $100 money transfer to Alex and wants to ensure the message is not accidentally altered on its way to the receiver.



- Hashing can be used to detect accidental changes but cannot be used to guard against deliberate changes that are made by a threat actor.

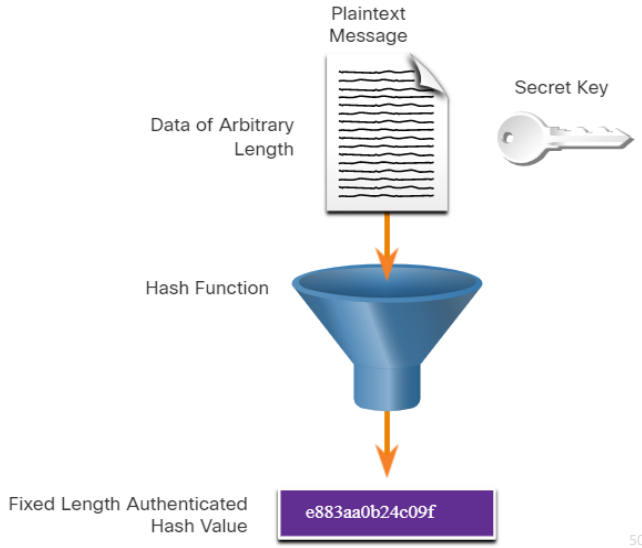
- Therefore, hashing is vulnerable to MiTM attacks and does not provide security to transmitted data.

MD5 and SHA (Cont.)

- There are four well-known hash functions:
  - **MD5 with 128-bit digest** - A one-way function that produces a 128-bit hashed message. It is a legacy algorithm and should be avoided and used only when no better alternatives are available. It is recommended that SHA-2 or SHA-3 be used instead.
  - **SHA-1** – It is very similar to the MD5 hash functions. Several versions exist. It creates a 160-bit hashed message and is slightly slower than MD5. It has known flaws and is a legacy algorithm.
  - **SHA-2** – It includes SHA-224 (224 bit), SHA-256 (256 bit), SHA-384 (384 bit), and SHA-512 (512 bit). If you are using SHA-2, then the SHA-256, SHA-384, and SHA-512 algorithms should be used whenever possible.
  - **SHA-3** – The newest hashing algorithm and an alternative and eventual replacement for the SHA-2 family of hashing algorithms. It includes SHA3-224 (224 bit), SHA3-256 (256 bit), SHA3-384 (384 bit), and SHA3-512 (512 bit). The SHA-3 family are next-generation algorithms and should be used whenever possible.

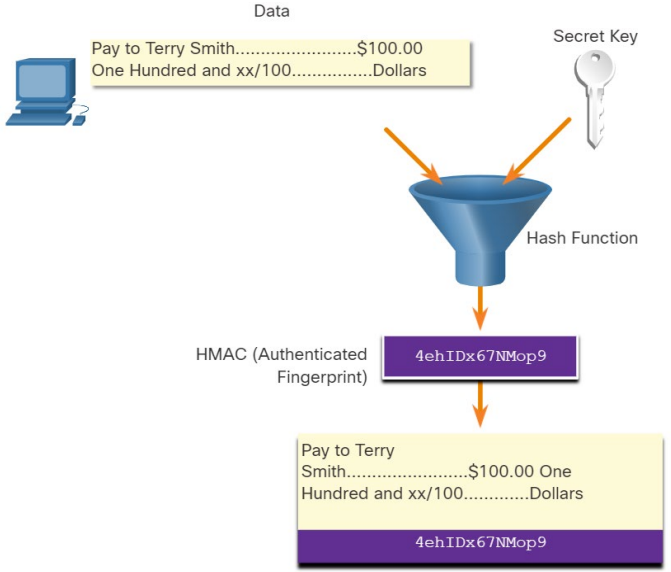
Origin Authentication

- Use a keyed-hash message authentication code (HMAC) to add origin authentication and integrity assurance.
- HMAC uses an additional secret key as input to the hash function.
- **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 now depends on the input data and the secret key.
  - 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.

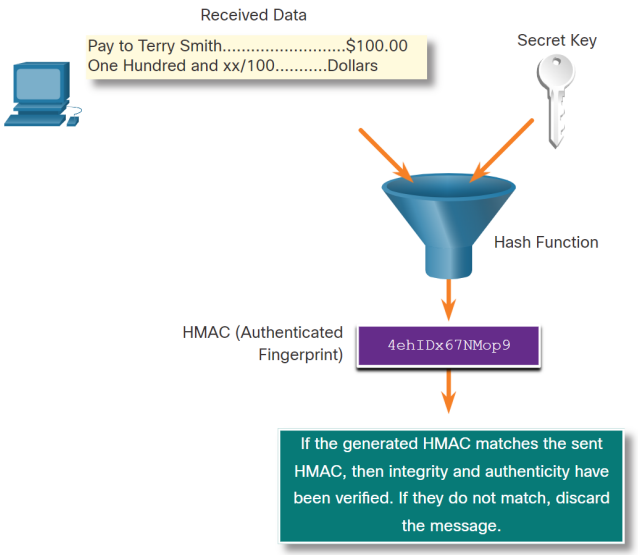


Origin Authentication (Cont.)

- **Creating the HMAC Value**
  - The sending device inputs data (such as Terry Smith's pay of $100 and the secret key) 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.



Origin Authentication (Cont.)

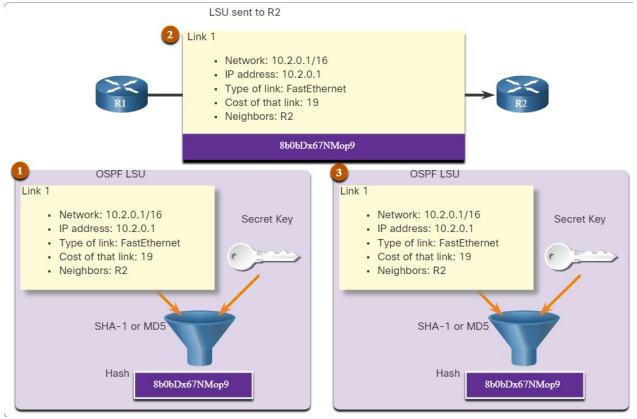


- **Verifying the HMAC Value**
  - The receiving device removes the digest from the message and uses the plaintext message with its secret key as input into the same hashing function.
  - If the digest that is calculated by the receiving device is equal to the digest that was sent, the message has not 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.

Origin Authentication (Cont.)

- The figure shows how HMACs are used by Cisco routers that are configured to use Open Shortest Path First (OSPF) routing authentication.

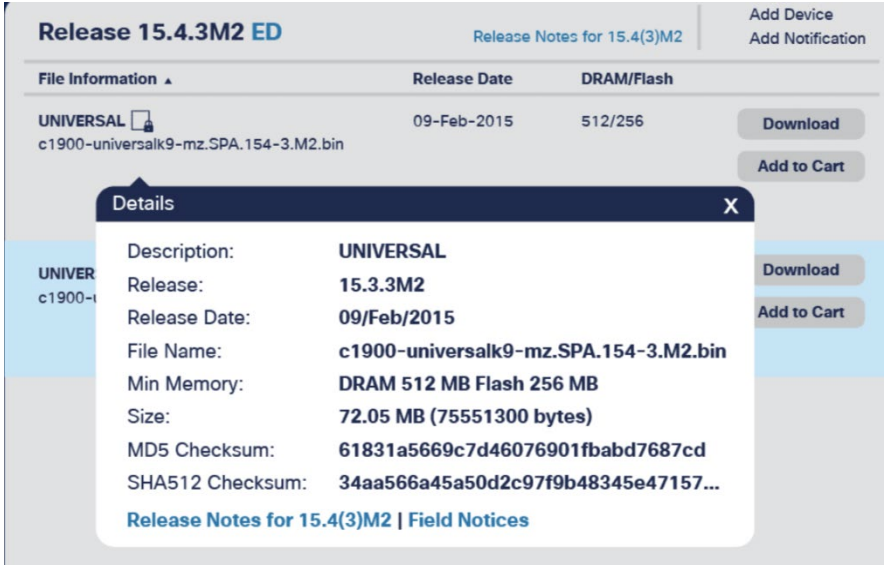
- R1 is sending a link state update (LSU) regarding a route to network 10.2.0.0/16:
  1. R1 calculates the hash value using the LSU message and the secret key.
  2. The resulting hash value is sent with the LSU to R2.
  3. R2 calculates the hash value using the LSU and its secret key. R2 accepts the update if the hash values match. If they do not match, R2 discards the update.



18.4 Using Hashes

Hashing Files and Digital Media

- Integrity ensures that data and information is complete and unaltered at the time of its acquisition.
- To verify the integrity of all Cisco IOS images, Cisco provides MD5 and SHA checksums at its Download Software portal.
- The user can make a comparison of this MD5 digest against the MD5 digest of a Cisco IOS image installed on a device and can feel confident that no one has tampered with or modified the Cisco IOS image file.



Hashing Files and Digital Media (Cont.)

- The field of digital forensics uses hashing to verify all digital media that contains files.

- For example, the examiner creates a hash and a bit-for-bit copy of the media containing the files to produce a digital clone.
- The examiner compares the hash of the original media with the copy and if the two values match, the copies are identical.
- The fact that one set of bits is identical to the original set of bits establishes fixity.
- Fixity helps to answer several questions:
  - Does the examiner have the files they expect?
  - Is the data corrupted or changed?
  - Can the examiner prove that the files are not corrupt?
- Now the forensics expert can examine the copy for any digital evidence while leaving the original intact and untouched.

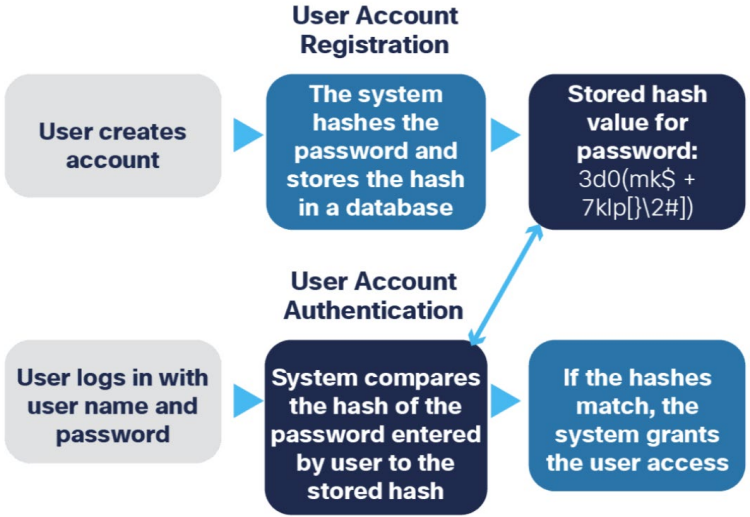
```
R1# verify / md5 flash:
c1900-universalk9-mz.SPA.154-3.M2.bin
.....
.....
.....
.....

.....MD5 of flash0:
c1900-universalk9-mz.SPA.154-3.M2.bin Done!
verify / md5 (flash0 : c1900-universalk9-mz . SPA .
154 - 3 . M@ . bin) -
61831a5669c7d46076901fbabd7687cd
```

Hashing Passwords

- Hashing algorithms can turn any amount of data into a fixed-length fingerprint or digital hash.
- Nobody can reverse a digital hash to discover the original input.
- If the input changes at all, it results in a different hash.
- This works to protect passwords.
- A system needs to store a password in a form that protects it and keeps it away from prying eyes, while also being able to still verify that a user's password is correct.

- This diagram shows the workflow for user account registration and authentication using a hash-based System.
- The system never writes the user's password to the hard drive, it only stores the digital hash.
- The password is truly only known to the user who set it.



Cracking Hashes

- To crack a hash, an attacker must guess the password.

- The top two attacks used to guess passwords are dictionary and brute-force attacks:
  - Dictionary attack
    - It uses a file containing common words, phrases and passwords.
    - The file has the hashes of these common passwords calculated.
    - It compares the hashes in the file with the password hashes and if a hash matches, the attacker will know a group of potentially good passwords used in this system.
  - Brute-force attack
    - It attempts every possible combination of characters up to a given length.
    - It takes a lot of processor power and time.
    - In theory, it is just a matter of time before this method discovers the password.

Salting

- Salting makes password hashing more secure.
- If two users have the same password, they will also have the same password hashes.
- A salt, which is a random string of characters, is an additional input added to the password before hashing.
- This creates a different hash result even when the two passwords are identical.
- Then, the database stores both the hash and the salt.
- The same password generates a different hash for different users, because the salt in each instance is different.
- Meanwhile, the salt does not have to be secret since it is a random number.

Implementing Salting

- A cryptographically secure pseudo-random number generator (CSPRNG) is the best way to generate salt.

- CSPRNGs generate a random number that has a high level of randomness and is completely unpredictable, so it is cryptographically secure.

- The following recommendations will help ensure successful implementation of salting:
  - The salt needs to be unique for every user password.
  - Never reuse a salt.
  - The length of the salt should match the length of the hash function's output.
  - Always hash on the server, in a web application.

- Key stretching makes attempts to figure out passwords work very slowly and is a technique that will also help to protect against attack.

- This makes high-end attacker hardware that can attempt to crack billions of hashes per second less effective.

Implementing Salting (Cont.)

- **To store a password:**
  - Use CSPRNG to generate a long, random salt.
  - Add the salt to the beginning of the password.
  - Hash it with SHA-256, a standard cryptographic hash function.
  - Save the salt and the hash in the user's database record.

- **To validate a password:**
  - Retrieve a user's salt and hash from the database.
  - Add the salt to the password and hash it with the same hash function.
  - Compare the hash of the password just submitted by the user trying to log in to the one stored in the database.
  - If the hashes do not match, the password the user has just tried to log in with is incorrect.

Preventing Attacks

- Salting prevents an attacker from using a dictionary attack to try to guess passwords and makes it impossible to use lookup tables and rainbow tables to crack a hash.

Lookup tables	It stores the pre-computed hashes of passwords in a password dictionary, along with the corresponding password. They are a data structure that processes hundreds of hash lookups per second.
Reverse lookup tables	This attack allows the cybercriminal to launch a dictionary or brute-force attack on many hashes without the pre-computed lookup table. The cybercriminal creates a lookup table that plots each password hash from the breached account database to a list of users. The cybercriminal hashes each password guess and uses the lookup table to get a list of users whose password matched the cybercriminal's guess.
Rainbow tables	They sacrifice hash-cracking speed to make the lookup tables smaller. A smaller table means that the table can store the solutions to more hashes in the same amount of space.

Lab – Hashing Things Out

- In this lab, you will complete the following objectives:
  - Part 1: Hashing a Text File with OpenSSL
  - Part 2: Verifying Hashes

18.5 Public Key Cryptography

Using Digital Signatures

- Digital Signatures are a mathematical technique used to provide authenticity, integrity, and nonrepudiation and have specific properties that enable entity authentication and data integrity.

- They serve as legal proof that the data exchange did take place.

- The properties of digital signatures are:
  - **Authentic:** The signature cannot be forged and provides proof that the signer, and no one else, signed the document.
  - **Unalterable:** After a document is signed, it cannot be altered.
  - **Not Reusable:** The document signature cannot be transferred to another document.
  - **Non-Repudiated:** The signed document is the same as a physical document because the signature is proof that the document has been signed by the actual person.

Using Digital Signatures (Cont.)

- Digital signatures are commonly used in the following two situations:
  - **Code signing**
    - It is used for data integrity and authentication purposes.
    - It 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 like a virtual ID card and used to authenticate the identity of a system with a vendor website and establish an encrypted connection to exchange confidential data.

Using Digital Signatures (Cont.)

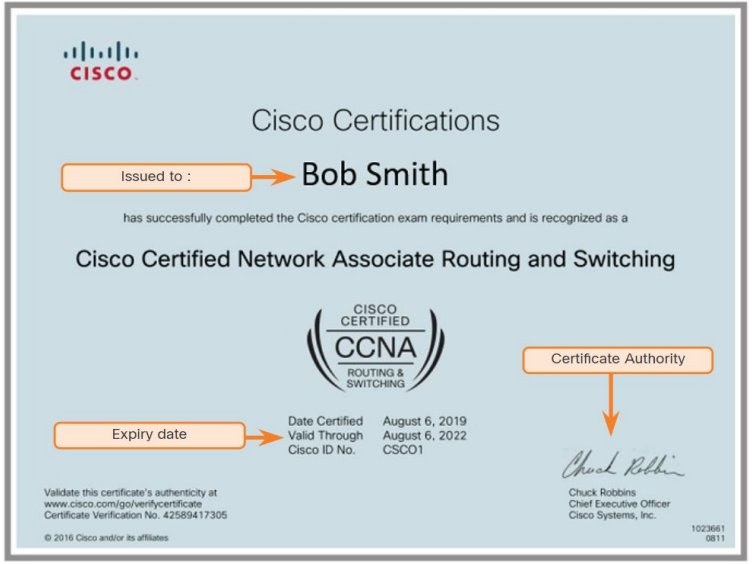
- Digital Signature Standard (DSS) algorithms used for generating and verifying digital signatures:
  - **Digital Signature Algorithm (DSA)**: It is the original standard for generating public and private key pairs, and for generating and verifying digital signatures.
  - **Rivest-Shamir Adelman Algorithm (RSA)**: It is an asymmetric algorithm commonly used for generating and verifying digital signatures.
  - **Elliptic Curve Digital Signature Algorithm (ECDSA)**: It is a newer variant of DSA and provides digital signature authentication and non-repudiation with the added benefits of computational efficiency, small signature sizes, and minimal bandwidth.

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 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.

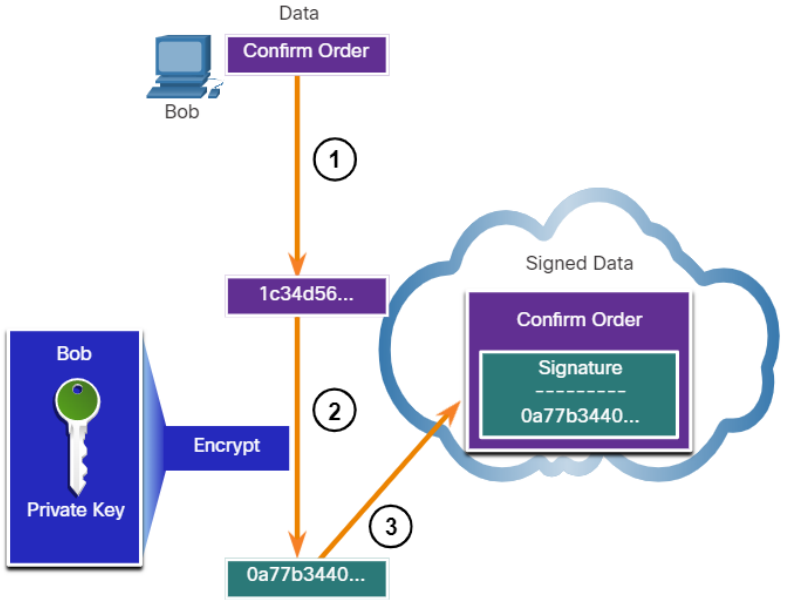
Digital Signatures for Digital Certificates

- A **digital certificate** is equivalent to an electronic passport and enables users, hosts, and organizations to securely exchange information over the Internet.
- Specifically, it is used to authenticate and verify that a user who is sending a message is who they claim to be.
- They can also be used to provide confidentiality for the receiver with the means to encrypt a reply.
- They are like physical certificates.
- In the figure, the paper-based CCNA-S certificate identifies who the certificate is issued to, who authorized the certificate, and for how long the certificate is valid.



Digital Signatures for Digital Certificates (Cont.)

- This scenario will help you understand how a digital signature is used:
  - Bob is confirming an order with Alice.
  - Alice is ordering from Bob's website.
  - Alice has connected with Bob's website, and after the certificate has been verified, Bob's certificate is stored on Alice's website.
  - The certificate contains Bob's public key that is used to verify Bob's digital signature.



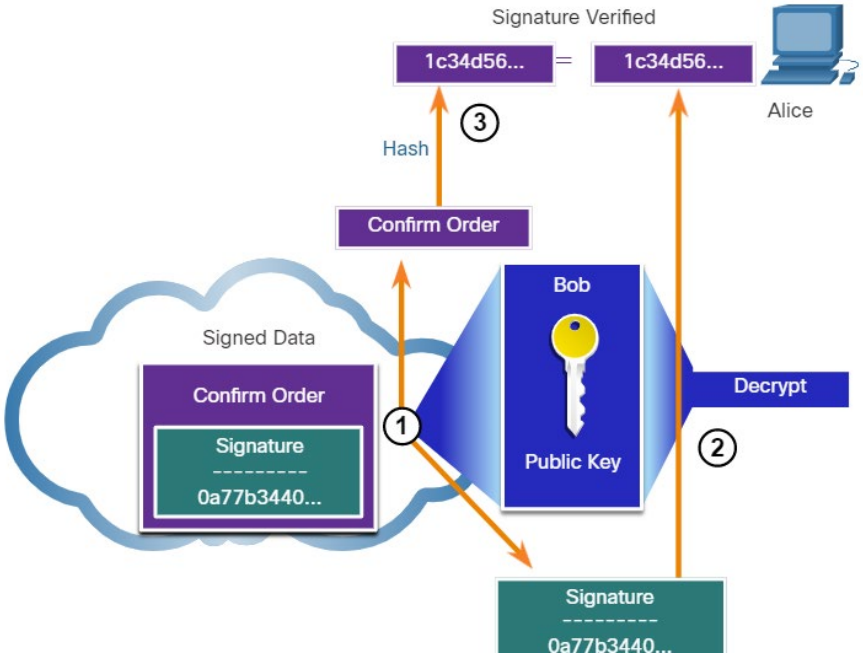
Digital Signatures for Digital Certificates (Cont.)

- When Alice receives the digital signature, the following process occurs.

1. Alice's receiving device accepts the order confirmation with the digital signature and obtains Bob's public key.

2. Alice's computer then decrypts the signature using Bob's public key. This step reveals the assumed hash value of the sending device.

3. Alice's computer creates a hash of the received document, without its signature, and compares this hash to the decrypted signature hash. If the hashes match, the document is authentic. This means the confirmation was sent by Bob and that it has not changed since it was signed.



Lab - Generate and Use a Digital Signature

- In this lab, you will complete the following objectives:
  - Part 1: Use OpenSSL to generate a digital signature.
  - Part 2: Sign a document with the digital signature.
  - Part 3: Verify that a signed document has been changed.

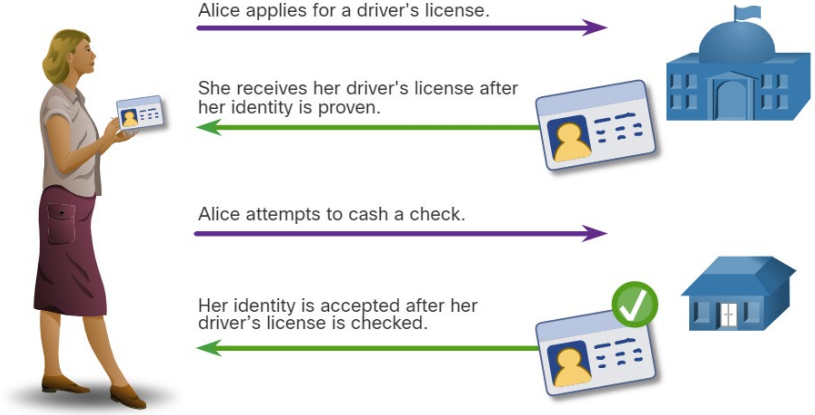
18.6 Authorities and the PKI Trust System

Public Key Management

- Internet traffic consists of traffic between two parties. When establishing an asymmetric connection between two hosts, the hosts will exchange their public key information.
- An SSL certificate is a digital certificate that confirms the identity of a website domain.
- To implement SSL on your website, you purchase an SSL certificate for your domain from an SSL Certificate provider.
- The trusted third party does an in-depth investigation prior to the issuance of credentials.
- After this in-depth investigation, the third-party issues credentials (i.e. digital certificate) 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.
- When computers attempt to connect to a website over HTTPS, the web browser checks the website's security certificate and verifies that it is valid and originated with a reliable certificate authority (CA).

Public Key Management (Cont.)

- These trusted third parties provide services like governmental licensing bureaus.
- The figure illustrates how a driver's license is analogous to a digital certificate.
- The CA is an organization that creates digital certificates by tying a public key to a confirmed identify, such as a website or individual.
- The Public Key Infrastructure (PKI) consists of specifications, systems, and tools that are used to create, manage, distribute, use, store, and revoke digital certificates.
- The PKI is an intricate system that is designed to safeguard digital identities from hacking by even the most sophisticated threat actors or nation states.



The Public Key Infrastructure

1. PKI Certificates contain an entity's or individual's public key, its purpose, the 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 CA is a trusted third party that issues PKI certificates to entities and individuals after verifying their identity. It signs these certificate using its private key.

4. The certificate database stores all certificates approved by the CA.

The Public Key Infrastructure (Cont.)

- 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.

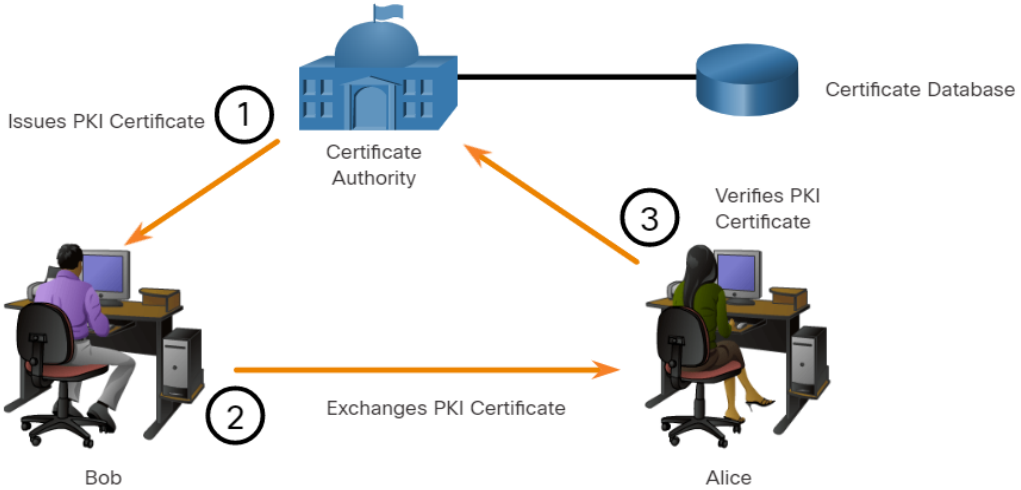
- The figure shows the main elements of the PKI.

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.

The Public Key Infrastructure (Cont.)

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

  - Bob has received his digital certificate from the CA.
  - This certificate is used whenever Bob communicates with other parties.
  - Bob communicates with Alice.
  - When Alice receives Bob's digital certificate, she communicates with the trusted CA to validate Bob's identity.



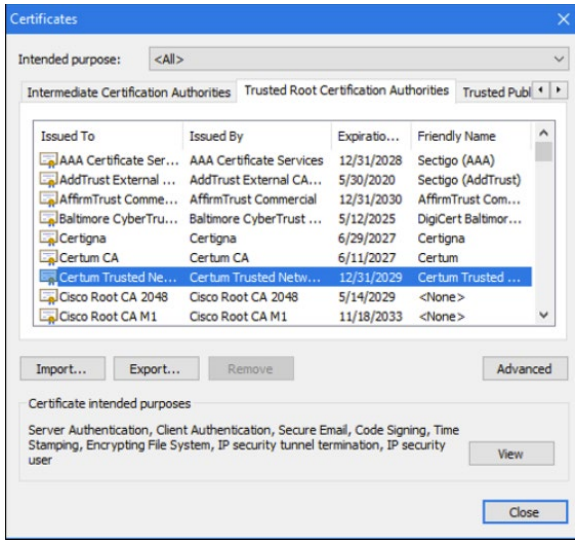
The PKI Authorities System

- 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, especially those that are outsourced, issue certificates based on classes which determine how trusted a certificate is.
- The table provides a description of the class number that determines 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.

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. Independent verification and checking of identity and authority is done by the certificate authority.
4	Used for online business transactions between companies.
5	Used for private organizations or government security.

The PKI Authorities System (Cont.)

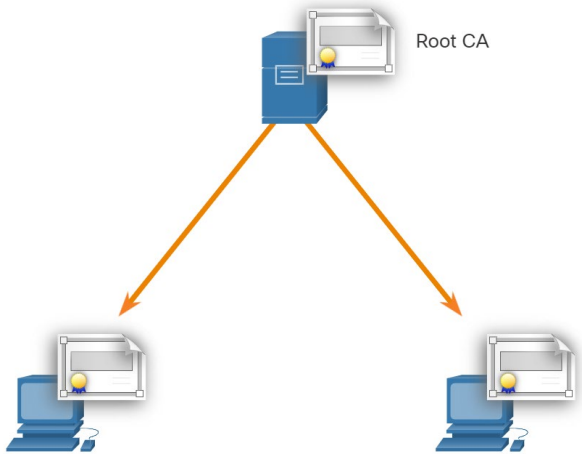
- **iFrame Component Test**
  - Some CA public keys are preloaded, such as those listed in web browsers.
  - The figure displays various VeriSign certificates contained in the certificate store on the host.
  - Any certificates signed by any of the CAs in the list will be seen by the browser as legitimate and will be trusted automatically.



The PKI Trust System

- PKIs can form different topologies of trust and the simplest is the single-root PKI topology.
- As shown in the figure below, a single CA (root CA), issues all the certificates to the end users, which are usually within the same organization.
- The benefit to this approach is its simplicity.
- However, it is difficult to scale to a large environment because it requires a strictly centralized administration, which creates a single point of failure.

Single-Root PKI Topology

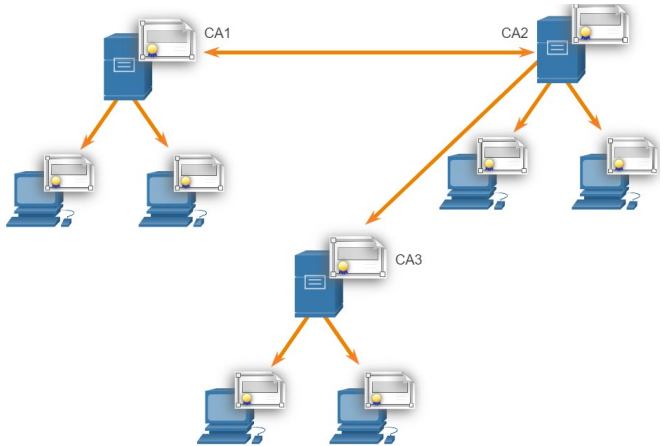


The PKI Trust System (Cont.)

On larger networks, PKI CAs may be linked using two basic architectures:

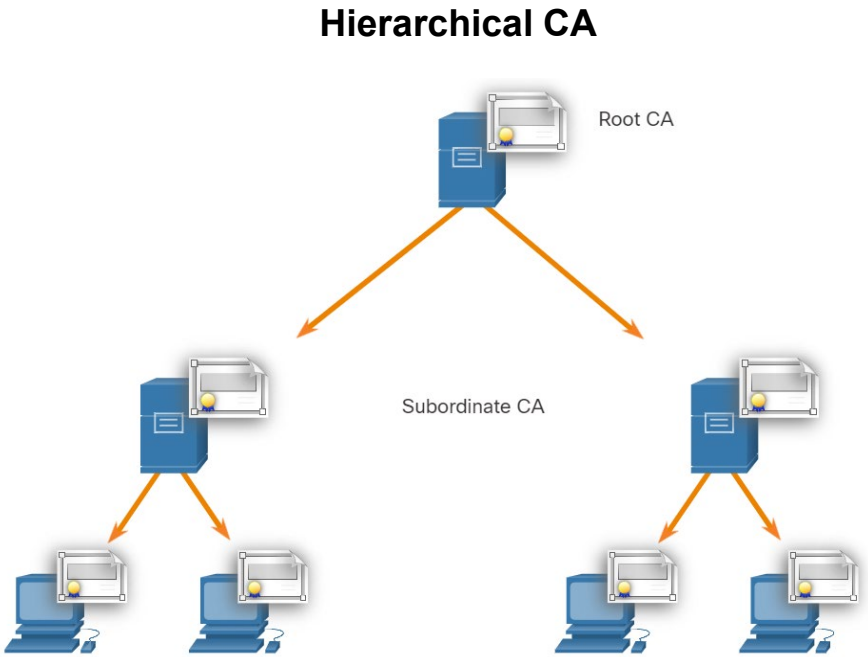
- **Cross-certified CA topologies**
  - As shown in the figure, this is a peer-to-peer model in which individual CAs establish trust relationships with other CAs by cross-certifying CA certificates.
  - Users in either CA domain are also assured that they can trust each other.
  - This provides redundancy and eliminates the single-point of failure.

Cross-Certified CA



The PKI Trust System (Cont.)

- **Hierarchical CA topologies**
  - The highest-level CA (root CA) can issue certificates to end users and to a subordinate CA.
  - The sub-CAs could be created to support various business units, domains, or communities of trust.
  - The root CA maintains the established "community of trust" by ensuring that each entity in the hierarchy conforms to a minimum set of practices.
  - The benefits include increased scalability and manageability. It also works well in most large organizations.
  - A hierarchical and cross-certification topology can be combined to create a hybrid infrastructure.



Interoperability of Different PKI Vendors

1. **SSL** - Secure web servers use 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 certificates when RSA-based authentication is used for IKE.

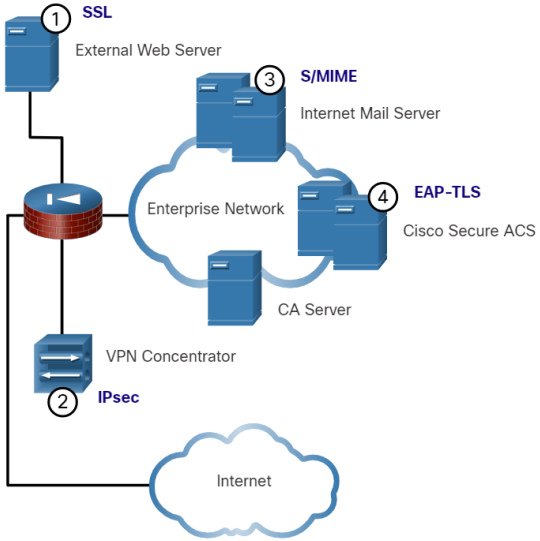
3. **S/MIME** - User mail agents that support mail protection with the S/MIME protocol use X.509 certificates.

4. **EAP-TLS** - Cisco switches can use certificates to authenticate end devices that connect to LAN ports using 802.1x between the adjacent devices. The authentication can be proxied to a central ACS via the EAP-TLS.

- Interoperability between a PKI and its supporting services, such as LDAP and X.500 directories, is a concern because many CA vendors have proposed and implemented proprietary solutions instead of waiting for standards to develop.

Interoperability of Different PKI Vendors (Cont.)

- To address this interoperability concern, the IETF published the Internet X.509 PKI Certificate Policy and Certification Practices Framework (RFC 2527).
- The X.509 version 3 (X.509 v3) standard defines the format of a digital certificate.
- Refer to the figure for more information about X.509 v3 applications.
- As shown in the figure, the X.509 format is already extensively used in the infrastructure of the internet.



Certificate Enrollment, Authentication, and Revocation

- The first step in the CA authentication procedure is to securely obtain a copy of the CA's public key.
- All systems that leverage the PKI must have the CA's public key (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.
- For many systems such as web browsers, the distribution of CA certificates is handled automatically.
- The web browser comes pre-installed with a set of public CA root certificates.
- Organizations and their website domains push their public certificates to website visitors.
- CAs and certificate domain registrars create and distribute private and public certificates to clients that purchase certificates.
- 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 OOB using the telephone.

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

Certificate Enrollment, Authentication, and Revocation (Cont.)

- 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.
- Authentication no longer requires the presence of the CA server, and each user exchanges their certificates containing public keys.
- Certificates must sometimes be revoked.
- The most common methods of revocation are:
  - **Certification Revocation List (CRL):** A list of revoked certificate serial numbers that have been invalidated because they expired. PKI entities regularly poll the CRL repository to receive the current CRL.
  - **Online Certificate Status Protocol (OCSP):** An internet protocol used to query an OCSP server for the revocation status of an X.509 digital certificate. Revocation information is immediately pushed to an online database.

Lab – Certificate Authority Stores

- In this lab, you will complete the following objectives:
  - Part 1: Certificates Trusted by Your Browser
  - Part 2: Checking for Man-in-the-Middle

18.7 Applications and Impacts of Cryptography

PKI Applications

- Where can PKI be used by an enterprise?

- 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

Encrypted Network Transactions

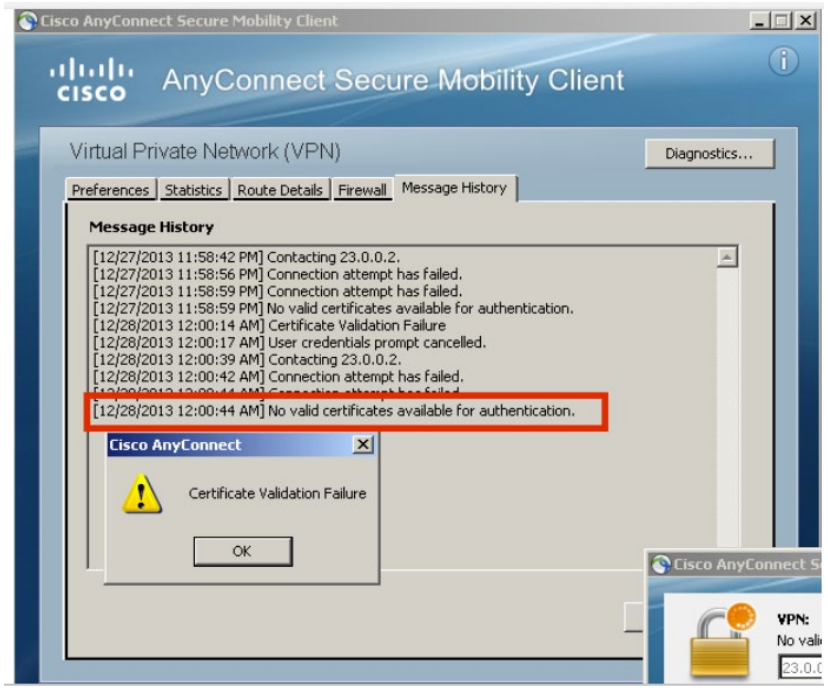
- Some of these issues can be avoided because the SSL/TLS protocols are extensible and modular.
- This is known as a cipher suite and its key components are the MAC algorithm, the encryption algorithm, the key exchange algorithm, and the authentication algorithm.
- These can be changed without replacing the entire protocol and is very helpful because the different algorithms continue to evolve.
- When the protocol versions within the cipher suite change, the SSL/TLS version number also changes.
- A security analyst must be able to recognize and solve potential problems related to permitting PKI-related solutions on the enterprise network.
- Consider how the increase of SSL/TLS traffic poses a major security risk to enterprises because the traffic is encrypted and cannot be intercepted and monitored by normal means.
- Users can introduce malware or leak confidential information over an SSL/TLS connection.
- Threat actors can use SSL/TLS to introduce regulatory compliance violations, viruses, malware, data loss, and intrusion attempts in a network.

Encrypted Network Transactions (Cont.)

- Other SSL/TLS-related issues may be associated with validating the certificate of a web server.
- When this occurs, web browsers will display a security warning.
- PKI-related issues that are 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. Expired certificates may simply be the result of administrator oversight, but they may also reflect more serious conditions.
  - **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. Signature validation will fail if the root certificate of the CA hierarchy is not available in the browser’s certificate store.

Encrypted Network Transactions (Cont.)

- The figure shows an example of a signature validation error with the Cisco AnyConnect Mobility VPN Client.



Encryption and Security Monitoring

- Network monitoring becomes more challenging when packets are encrypted.
- Security analysts must be aware of those challenges and address them as best as possible.
- The increased use of HTTPS in the enterprise network introduces new challenges.
- Since HTTPS introduces end-to-end encrypted HTTP traffic (via TLS/SSL), it is not as easy to peek into user traffic.
- 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 IPS appliances to identify risks normally hidden by SSL.

Encryption and Security Monitoring (Cont.)

- 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.

18.8 Cryptography Summary

What Did I Learn in this Module?

- Two classes of encryption are used to provide data confidentiality: asymmetric and symmetric.
- Symmetric encryption algorithms (DES, 3 DES, and AES) are based on the premise that each communicating party knows the pre-shared key.
- Asymmetric algorithms, including RSA and PKI, are designed to use different keys for encryption and decryption and use a public and private key, providing confidentiality without pre-sharing a password.
- DH is an asymmetric mathematical equation algorithm that allows two computers to generate an identical shared secret key without having communicated before.
- Steganography conceals data in another file such as a graphic, audio, or video file.
- There are four elements of secure communications: data integrity, origin authentication, data confidentiality, and data non-repudiation.
- Hash functions are one-way functions used to verify and ensure data integrity and are vulnerable to MiTM attacks.
- A hash function takes a variable block of binary data (message), and produces a fixed-length, condensed representation (hash).
- There are four well-known hash functions: MD5 with 128-bit digest, SHA-1, SHA-2, and SHA-3.

What Did I Learn in this Module? (Cont.)

- The two top attacks to crack a hash used are dictionary and brute-force attacks.
- A salt is an additional input added to the password, creating a different hash result even when the two passwords are identical.
- Digital signatures are a mathematical technique used to provide three basic security services: authenticity, integrity, and nonrepudiation, and are used in the following two situations: code signing and digital certificates.
- Three DSS algorithms used for generating and verifying digital signatures include: DSA, RSA, and ECDSA.
- When establishing a secure connection between two hosts, they exchange their public key information.
- Trusted third parties on the internet validate the authenticity of these public keys using digital certificates.
- The PKI consists of specifications, systems, and tools used to create, manage, distribute, use, store, and revoke digital certificates and are needed to support large-scale distribution of public encryption keys.
- Many vendors provide CA servers as a managed service or as an end-user product.
- The class number (0 through 5) is determined by how rigorous the procedure was that verified the identity of the holder when the certificate was issued, with five being the highest.
- PKIs can form different topologies of trust, the simplest being the single-root PKI topology.

What Did I Learn in this Module? (Cont.)

- Interoperability between PKI and its supporting services is a concern because many CA vendors have proposed and implemented proprietary solution instead of waiting for standards to develop.
- Common uses of PKIs include: 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.
- Other SSL/TSL related issues may be associated with validating the certificate of the web server.
- PKI-related issues that are associated with security warnings include validity date range and signature validation.
- The key components of the cipher suite are the MAC Algorithm, the encryption algorithm, the key exchange algorithm, and the authentication algorithm.
- Cryptography is dynamic and security analysts must maintain a good understanding of algorithms and operations to investigate cryptography-related security incidents.
- Encrypted communications can make network security data payloads unreadable by cybersecurity analysts.