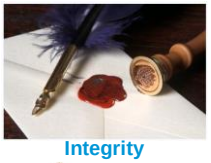
Cryptographic Systems

Managing Administrative Access

- A network LAN can be secured through:
  - Device hardening
  - AAA access control
  - Firewall features
  - IPS implementations
- How is network traffic protected when traversing the public Internet?
  - Using cryptographic methods

Secure Communications Requires …

Authentication



Integrity

Confidentiality

Authentication

- Authentication guarantees that the message:
  - Is not a forgery.
  - Does actually come from who it states it comes from.
- Authentication is similar to a secure PIN for banking at an ATM.
  - The PIN should only be known to the user and the financial institution.
  - The PIN is a shared secret that helps protect against forgeries.

Authentication

- Data nonrepudiation is a similar service that allows the sender of a message to be uniquely identified.
- This means that a sender / device cannot deny having been the source of that message.
  - It cannot repudiate, or refute, the validity of a message sent.

Integrity

- Data integrity ensures that messages are not altered in transit.
  - The receiver can verify that the received message is identical to the sent message and that no manipulation occurred.
- European nobility ensured the data integrity by creating a wax seal to close an envelope.
  - The seal was often created using a signet ring.
  - An unbroken seal on an envelope guaranteed the integrity of its contents.
  - It also guaranteed authenticity based on the unique signet ring impression.

Confidentiality

- Data confidentiality ensures privacy so that only the receiver can read the message.
- Encryption is the process of scrambling data so that it cannot be read by unauthorized parties.
  - Readable data is called plaintext, or cleartext.
  - Encrypted data is called ciphertext.
- A key is required to encrypt and decrypt a message.
  - The key is the link between the plaintext and ciphertext.

Managing Administrative Access

- Authentication, integrity, and confidentiality are components of cryptography.
- Cryptography is both the practice and the study of hiding information.
- It has been used for centuries to protect secret documents.
  - Today, modern day cryptographic methods are used in multiple ways to ensure secure communications.

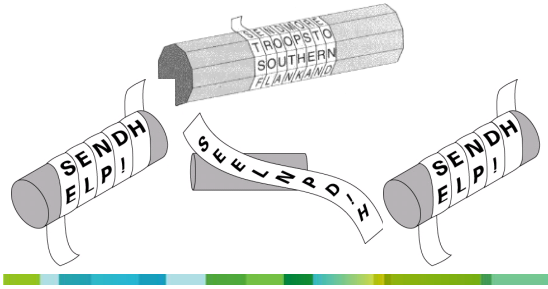
History of Cryptography

Scytale

- Earliest cryptography method.
  - Used by the Spartans in ancient Greece.

Scytale

- It is a rod used as an aid for a transposition cipher.
  - The sender and receiver had identical rods (scytale) on which to wrap a transposed messaged.



Caesar Cipher

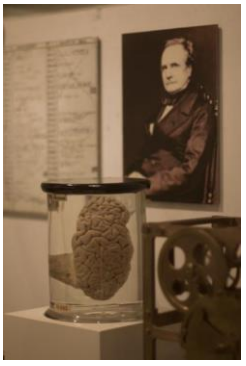
- When Julius Caesar sent messages to his generals, he didn't trust his messengers.
- He encrypted his messages by replacing every letter:
  - A with a D
  - B with an E
  - and so on
- His generals knew the "shift by 3" rule and could decipher his messages.

Vigenère Cipher

- In 1586, Frenchman Blaise de Vigenère described a poly alphabetic system of encryption.
  - It became known as the Vigenère Cipher.
- Based on the Caesar cipher, it encrypted plaintext using a multi-letter key.
  - It is also referred to as an autokey cipher.

Note of interest …

- It took 300 years for the Vigenère Cipher to be broken by Englishman Charles Babbage.
  - Father of modern computers
- Babbage created the first mechanical computer called the difference engine to calculate numerical tables.
  - He then designed a more complex version called the analytical engine that could use punch cards.
  - He also invented the pilot (cow-catcher).



Confederate Cipher Disk

- Thomas Jefferson, the third president of the United States, invented an encryption system that was believed to have been used when he served as secretary of state from 1790 to 1793.

German Enigma Machine

- Arthur Scherbius invented the Enigma in 1918 and sold it to Germany.
  - It served as a template for the machines that all the major participants in World War II used.
- It was estimated that if 1,000 cryptanalysts tested four keys per minute, all day, every day, it would take 1.8 billion years to try them all.
  - Germany knew their ciphered messages could be intercepted by the allies, but never thought they could be deciphered.

http://users.telenet.be/d.rijmenants/en/enigma.htm

Code Talkers

- During World War II, Japan was deciphering every code the Americans came up with.
  - A more elaborate coding system was needed.
  - The answer came in the form of the Navajo code talkers.
- Code talkers were bilingual Navajo speakers specially recruited during World War II by the Marines.
- Other Native American code talkers were Cherokee, Choctaw and Comanche soldiers.

Code Talkers

- Not only were there no words in the Navajo language for military terms, the language was unwritten and less than 30 people outside of the Navajo reservations could speak it, and not one of them was Japanese.
  - By the end of the war, more than 400 Navajo Indians were working as code talkers.

Cipher Text

Cipher Text

- A cipher is a series of well-defined steps that can be followed as a procedure when encrypting and decrypting messages.
- Each encryption method uses a specific algorithm, called a cipher, to encrypt and decrypt messages.
- There are several methods of creating cipher text:
  - Transposition
  - Substitution
  - Vernam

Transposition Ciphers

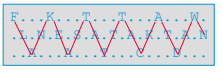
- In transposition ciphers, no letters are replaced; they are simply rearranged.
- For example:
  - Spell it backwards.
- Modern encryption algorithms, such as the DES (Data Encryption Standard) and 3DES, still use transposition as part of the algorithm.

Transposition Rail Fence Cipher

1. Solve the ciphertext.

Ciphered text

2. Use a rail fence cipher and a key of 3.



3. The clear text message.

Clear text

Substitution Cipher

- Substitution ciphers substitute one letter for another.
  - In their simplest form, substitution ciphers retain the letter frequency of the original message.
- Examples include:
  - Caesar Cipher
  - Vigenère Cipher

Let's Encode using the Caesar Cipher!

1. The cleartext message.

Clear text

2. Encode using a key of 3. Therefore, A becomes a D, B an E, …

3. The encrypted message becomes …

Ciphered text

Let's Decode

1. Solve the ciphertext.

Ciphered text

2. Use a shift of 6 (ROT6).

3. The clear text message.

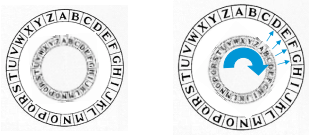
Clear text

Caesar Cipher Disk

1. 

Clear text

The clear text message would be encoded using a key of 3.

2. 

Shifting the inner wheel by 3, then the A becomes D, B becomes E, and so on.

3. 

Ciphered text

The clear text message would appear as follows using a key of 3.

Vigenère Cipher

- The Vigenère cipher is based on the Caesar cipher, except that it encrypts text by using a different polyalphabetic key shift for every plaintext letter.
  - The different key shift is identified using a shared key between sender and receiver.
  - The plaintext message can be encrypted and decrypted using the Vigenere Cipher Table.
- For example:
  - A sender and receiver have a shared secret key: SECRETKEY.
  - Sender uses the key to encode: FLANK EAST ATTACK AT DAWN.

Vernam Cipher

- In 1917, Gilbert Vernam, an AT&T Bell Labs engineer invented and patented the stream cipher and later co-invented the one-time pad cipher.
  - Vernam proposed a teletype cipher in which a prepared key consisting of an arbitrarily long, non-repeating sequence of numbers was kept on paper tape.
  - It was then combined character by character with the plaintext message to produce the ciphertext.
  - To decipher the ciphertext, the same paper tape key was again combined character by character, producing the plaintext.
- Each tape was used only once, hence the name one-time pad.
  - As long as the key tape does not repeat or is not reused, this type of cipher is immune to cryptanalytic attack because the available ciphertext does not display the pattern of the key.

Vernam Cipher

- Several difficulties are inherent in using one-time pads in the real world.
  - Key distribution is challenging.
  - Creating random data is challenging and if a key is used more than once, it becomes easier to break.
- Computers, because they have a mathematical foundation, are incapable of creating true random data.
- RC4 is a one-time pad cipher that is widely used on the Internet.
  - However, because the key is generated by a computer, it is not truly random.

Cryptology in Networking

- Authentication, integrity, and data confidentiality are implemented in many ways using various protocols and algorithms.
  - Choice depends on the security level required in the security policy.

	Integrity	Authentication	Confidentiality
Common cryptographic hashes, protocols, and algorithms	MD5 (weaker) SHA (stronger)	HMAC-MD5 HMAC-SHA-1 RSA and DSA	DES (weaker) 3DES AES (stronger)

CISCO

Cryptographic Hashes

Cryptographic Hashes

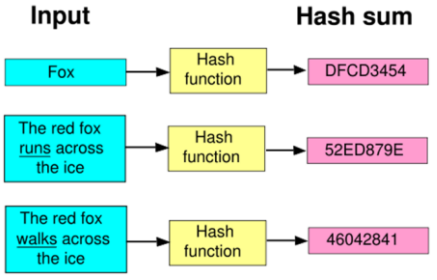
- A hash function takes binary data (message), and produces a condensed representation, called a hash.
  - The hash is also commonly called a Hash value, Message digest, or Digital fingerprint.
- Hashing is based on a one-way mathematical function that is relatively easy to compute, but significantly harder to reverse.
- Hashing is designed to verify and ensure:
  - Data integrity
  - Authentication

Hashes are used …

- To provide proof of authenticity when it is used with a symmetric secret authentication key, such as IP Security (IPsec) or routing protocol authentication.
- To provide authentication by generating one-time and one-way responses to challenges in authentication protocols such as the PPP CHAP.
- To provide a message integrity check proof such as those accepted when accessing a secure site using a browser.
- To confirm that a downloaded file (e.g., Cisco IOS images) has not been altered.

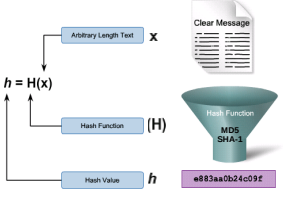
Collision Free

- Hashing is collision free which means that two different input values will result in different hash results.



Cryptographic Hash Math

- Take an arbitrarily length of clear text data to be hashed.
- Put it through a hash function.
- It produces a fixed length message digest (hash value).
- H(x) is:
  - Relatively easy to computer for any given x.
  - One way and not reversible.
- If a hash function is hard to invert, it is considered a one-way hash.



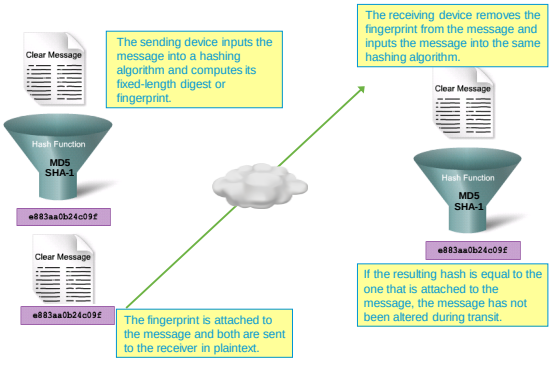
cisco

Hashing for Integrity

Hash for Integrity

- Hash functions (MD5 and SHA-1) can ensure message integrity but not confidentiality.
  - For instance, the sender wants to ensure that the message is not altered on its way to the receiver.

Hash for Integrity



Hash for Integrity

- Hashing only prevents the message from being changed accidentally, such as by a communication error.
- It's still susceptible to man-in-the-middle attacks.
  - A potential attacker could intercept the message, change it, recalculate the hash, and append it to the message.
  - There is nothing unique to the sender in the hashing procedure, so anyone can compute a hash for any data, as long as they have the correct hash function.
- These are two well-known hash functions:
  - Message Digest 5 (MD5) with 128-bit digests
  - Secure Hash Algorithm 1 (SHA-1) with 160-bit digests

Message Digest 5 (MD5)

- The MD5 algorithm was developed by Ron Rivest and is used in a variety of Internet applications today.
  - It is a one-way function.
  - It is also collision resistant.
- MD5 is essentially a complex sequence of simple binary operations, such as exclusive OR (XORs) and rotations, that are performed on input data and produce a 128-bit digest.

Secure Hash Algorithm (SHA)

- The U.S. National Institute of Standards and Technology (NIST) developed the Secure Hash Algorithm (SHA).
  - SHA-1, published in 1994, corrected an unpublished flaw in SHA.
  - It's very similar to the MD4 and MD5 hash functions.
- The SHA-1 algorithm takes a message of less than 2^{64} bits in length and produces a 160-bit message digest.
- This makes SHA-1 slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

MD5 versus SHA-1

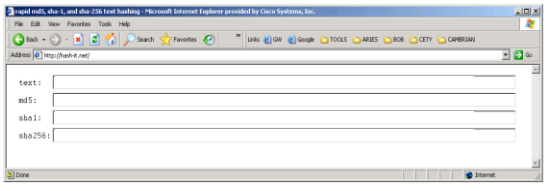
MD5	SHA-1
Based on MD4	Based on MD4
Computation involves 64 steps	Computation involves 80 steps
Algorithm must process a 128-bit buffer	Algorithm must process a 160-bit buffer
Faster	Slower
Less Secure	More secure

Secure Hash Algorithm (SHA)

- NIST published four additional hash functions collectively known as SHA-2 with longer digests:
  - SHA-224 (224 bit)
  - SHA-256 (256 bit)
  - SHA-384 (384 bit)
  - SHA-512 (512 bit)
- In response to a SHA-1 vulnerability announced in 2005, NIST recommends a transition from SHA-1 to the approved SHA-2 family.
- A newer more secure cryptographic hashing algorithm called SHA-3 has been developed by NIST. SHA-3 will eventually replace SHA-1 and SHA-2 and it should be used if available.

Secure Hash Algorithm (SHA)

- SHA-1 and SHA-2 are more resistant to brute-force attacks because their digest is at least 32 bits longer than the MD5 digest.



Hashing for Authenticity

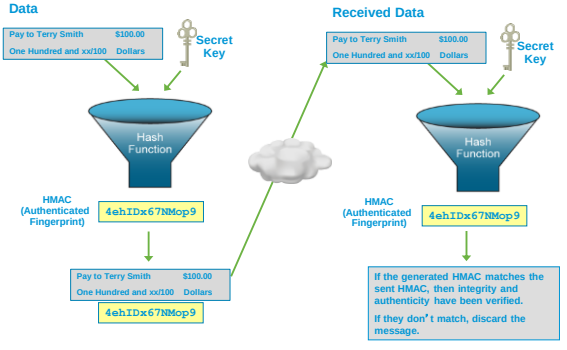
Keyed-Hash Message Authentication Code

- HMAC (or KHMAC) is a message authentication code (MAC) that is calculated using a hash function and a secret key.
  - Hash functions are the basis of the protection mechanism of HMACs.
  - The output of the hash function now depends on the input data and the secret key.
- Authenticity is guaranteed because only the sender and the receiver know the secret key.
  - Only they can compute the digest of an HMAC function.
  - This characteristic defeats man-in-the-middle attacks and provides authentication of the data origin.

Keyed-Hash Message Authentication Code

- The cryptographic strength of the HMAC depends on the:
  - Cryptographic strength of the underlying hash function.
  - Size and quality of the key.
  - Size of the hash output length in bits.
- Cisco technologies use two well-known HMAC functions:
  - Keyed MD5 or HMAC-MD5 is based on the MD5 hashing algorithm.
  - Keyed SHA-1 or HMAC-SHA-1 is based on the SHA-1 hashing algorithm.

HMAC in Action



HMAC and Cisco Products

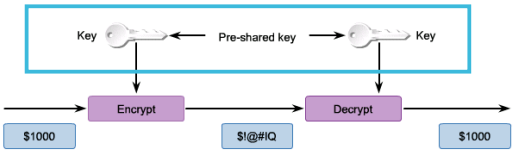
- Cisco products use hashing for entity authentication, data integrity, and data authenticity purposes.
- For example:
  - Authenticating routing protocol updates.
  - IPsec VPNs use MD5 and SHA-1 in HMAC mode, to provide packet integrity and authenticity.
  - IOS images downloaded from Cisco.com have an MD5-based checksum to check the integrity of downloaded images.
  - TACACS+ uses an MD5 hash as the key to encrypt the session.

Encryption = Confidentiality

Symmetric Encryption

- Symmetric encryption algorithms, also called shared secret-key algorithms, use the same pre-shared secret key to encrypt and decrypt data.
  - The pre-shared key is known by the sender and receiver before any encrypted communications begins.
- Because both parties are guarding a shared secret, the encryption algorithms used can have shorter key lengths.
  - Shorter key lengths mean faster execution.
- For this reason symmetric algorithms are generally much less computationally intensive than asymmetric algorithms.

Symmetric Encryption

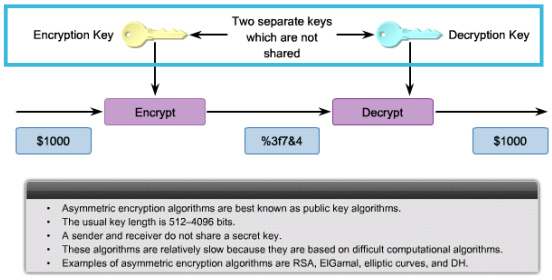


- Symmetric encryption algorithms are best known as shared-secret key algorithms.
- The usual key length is 80 - 256 bits.
- A sender and receiver must share a secret key.
- They are usually quite fast (wire speed) because these algorithms are based on simple mathematical operations.
- Examples of symmetric encryption algorithms are DES, 3DES, AES, IDEA, RC2/4/5/6, and Blowfish.

Asymmetric Encryption

- Asymmetric encryption algorithms, also called public key algorithms, use different keys to encrypt and decrypt data.
- Secure messages can be exchanged without having to have a pre-shared key.
- Because both parties do not have a shared secret, very long key lengths must be used to thwart attackers.
  - These algorithms are resource intensive and slower to execute.
- In practice, asymmetric algorithms are typically 100 to 1,000 times slower than symmetric algorithms.

Asymmetric Encryption



- Asymmetric encryption algorithms are best known as public key algorithms.
- The usual key length is 512–4096 bits.
- A sender and receiver do not share a secret key.
- These algorithms are relatively slow because they are based on difficult computational algorithms.
- Examples of asymmetric encryption algorithms are RSA, ElGamal, elliptic curves, and DH.

Digital Signatures

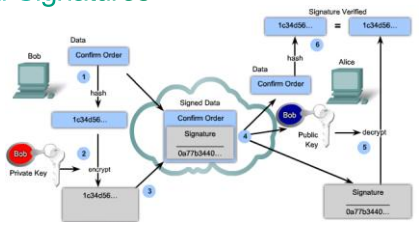
Digital Signatures Security Services

- Authenticity of digitally signed data:
  - Digital signatures authenticate a source, proving that a certain party has seen and signed the data in question.
- Integrity of digitally signed data:
  - Digital signatures guarantee that the data has not changed from the time it was signed.
- Nonrepudiation of the transaction:
  - The recipient can take the data to a third party, and the third party accepts the digital signature as a proof that this data exchange did take place.
  - The signing party cannot repudiate that it has signed the data.

Digital Signatures

- Digital signatures are often used in the following situations:
  - To provide a unique proof of data source, which can only be generated by a single party, such as contract signing in e-commerce environments.
  - To authenticate a user by using the private key of that user and the signature it generates.
  - To prove the authenticity and integrity of PKI certificates.
  - To provide nonrepudiation using a secure timestamp and a trusted time source.
    - Each party has a unique, secret signature key, which is not shared with any other party, making nonrepudiation possible.

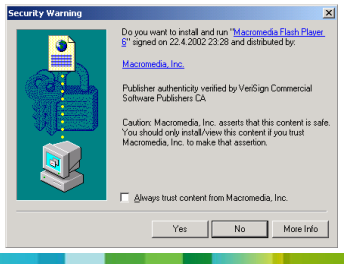
Digital Signatures



1. Bob creates a hash of the document.
2. Bob encrypts the hash with the private key.
3. The encrypted hash, known as the signature, is appended to the document.
4. Alice accepts the document with the digital signature and obtains Bob's public key.
5. Alice decrypts the signature using Bob's public key to unveil the assumed hash value.
6. Alice calculates the hash of the received document, without its signature, and compares this hash to the decrypted signature hash and if the hashes match = document is authentic.

Code Signing

- Digital signatures are commonly used for code signing:
  - Provide assurance of the authenticity and integrity of software codes.
  - The executable files, or possibly the entire installation package of a program, are wrapped with a digitally signed envelope, which allows the end user to verify the signature before installing the software.

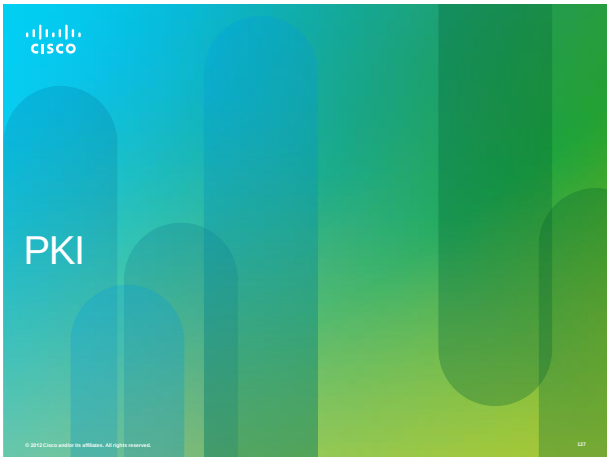


Digital Signing

- Well-known asymmetric algorithms, such as RSA or Digital Signature Algorithm (DSA), are typically used to perform digital signing.
- In 1994, the U.S. NIST selected the DSA as the Digital Signature Standard (DSS).
  - DSA is based on the discrete logarithm problem and can only provide digital signatures.
- A network administrator must decide whether RSA or DSA is more appropriate for a given situation.
  - DSA signature generation is faster than DSA signature verification.
  - RSA signature verification is much faster than signature generation.

RSA Scorecard

Description	Ron Rivest, Adi Shamir, and Len Adleman
Timeline	1977
Type of Algorithm	Asymmetric algorithm
Key size (in bits)	512 - 2048
Advantages	Signature verification is fast
Disadvantages	Signature generation is slow



PKI

- PKI is the service framework needed to support large-scale public key-based technologies.
  - Very scalable solutions which is an extremely important authentication solution for VPNs.
- PKI is a set of technical, organizational, and legal components that are needed to establish a system that enables large-scale use of public key cryptography to provide authenticity, confidentiality, integrity, and nonrepudiation services.
  - The PKI framework consists of the hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates.

PKI Terms

- Certificates:
  - Published public information containing the binding between the names and public keys of entities.
- Certificate authority:
  - A trusted third-party entity that issues certificates.
  - The certificate of a user is always signed by a CA.
  - Every CA also has a certificate containing its public key, signed by itself.
  - This is called a CA certificate or, more properly, a self-signed CA certificate.

Vendors Certificate

VeriSign
http://www.verisign.com

Entrust
Securing Digital Identities & Information
http://www.entrust.com

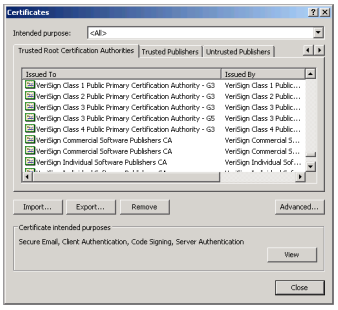
cybertrust certified perimeter
http://www.verizonbusiness.com/

Novell
http://www.novell.com

RSA
The Security Division of EMC
http://www.rsa.com/

Microsoft
http://www.microsoft.com

PKI Example

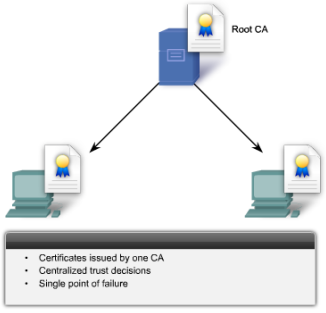


Certificate Authorities

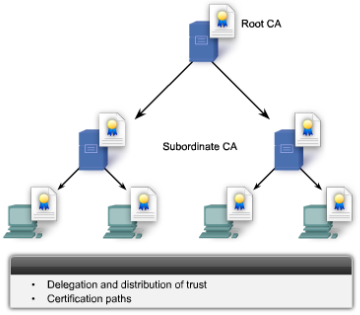
Level of Trust

- PKIs can form different topologies of trust, including:
  - Single-root PKI topologies
  - Hierarchical CA topologies
  - Cross-certified CA topologies

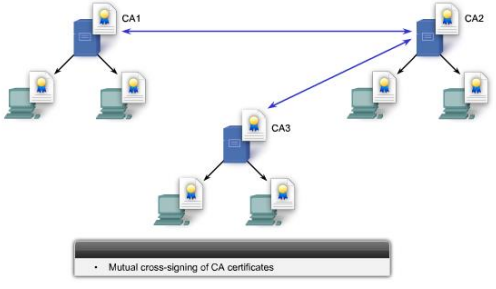
Single-Root PKI Topology (Root CA)



Hierarchical CA Topology



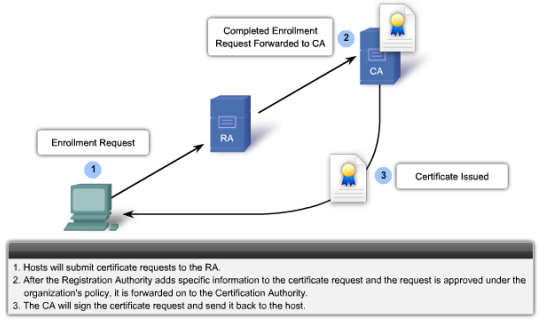
Cross-certified CA Topology



PKI Enrollment Process

- The issuing CA may be a:
  - Root CA (the top-level CA in the hierarchy)
  - Subordinate CA
- The PKI might employ registration authorities (RAs) to accept requests for enrollment in the PKI.
  - This reduces the burden on CAs in an environment that supports a large number of certificate transactions or where the CA is offline.

PKI Enrollment Process



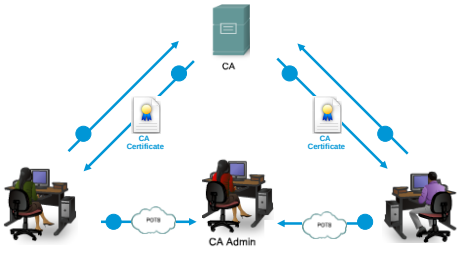
PKI Enrollment Process

- Usually tasks offloaded to an RA:
  - Authentication of users when they enroll with the PKI.
  - Key generation for users that cannot generate their own keys.
  - Distribution of certificates after enrollment.
- Additional tasks include:
  - Verifying user identity.
  - Establishing passwords for certificate management transactions.
  - Submitting enrollment requests to the CA.
  - Handling certificate revocation and re-enrollment.

CA Authentication Procedure

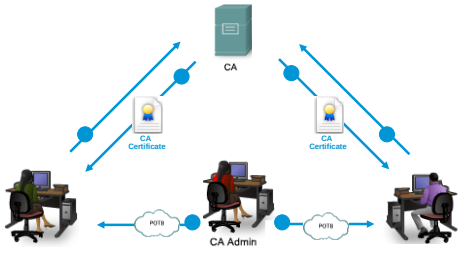
- The first step of the user is to securely obtain a copy of the public key of the CA.
  - The public key verifies all the certificates issued by the CA and is vital for the proper operation of the PKI.
- The public key, called the self-signed certificate, is also distributed in the form of a certificate issued by the CA itself.
- Only a root CA issues self-signed certificates.

CA Authentication Procedure



1. Alice and Bob request the CA certificate that contains the CA public key.
2. Upon receipt of the CA certificate, each system (of Alice and Bob) verifies the validity of the certificate using public key cryptography.
3. Alice and Bob follow up the technical verification done by their system by telephoning the CA administrator and verifying the public key and serial number of the certificate.

CA Authentication Retrieval



1. Alice and Bob forward a certificate request which includes their public key along and information that is encrypted using the public key of the CA.
2. Upon receipt of the certificate requests, the CA administrator telephones Alice and Bob to confirm their submittal and the public key and issues the certificate by adding some additional data to the certificate request, and digitally signing it all.
3. Either the end user manually retrieves the certificate or SCEP automatically retrieves the certificate, and the certificate is installed onto the system.