Public Key Infrastructure (PKI)

The Basics of Public Key Infrastructures

- **A PKI**
  - Binds public key to identity
  - Enables other entities to verify key-identity binding
  - Provides services for management of keys in a distributed system
- **Goal:**

Protect and distribute information that is needed in a widely distributed environment, where the users, resources and stakeholders may all be in different places at different times

The Basics of Public Key Infrastructures cont.

1. **Consists of:**
    1. Hardware, software, policies, services, programming interfaces, cryptographic algorithms, protocols, users, and utilities
2. **Provides:**
    1. Data integrity
    2. Data confidentiality
    3. Authentication
3. **Integrates:**
    1. Public key cryptography
    2. Digital certificates
    3. Certification authorities

Components of PKI

1. **Certificate/Certification Authority (CA)**
   - Confirms the identity of entities by issuing certificates
2. **Registration Authority (RA)**
   - Trusted by CA to authenticate users requesting digital certificates from CA
3. **Validation Authority / Repository (VA)**
   - Provides services used to validate a certificate
   - Database of active digital certificates for a domain
4. **Archive**
   - Stores and protects sufficient information to determine if a digital signature on an old document should be trusted
5. **Certificates**
   - Includes public key, identity, and other information

KEY TERMS

- Authority revocation list (ARL)
- CA certificate
- Certificate
- Certificate Authority (CA)
- Certificate path
- Certificate repository
- Certificate Revocation List (CRL)
- Certificate server
- Certificate signing request (CSR)
- Certification practices statement (CPS)
- Cross-certification certificate

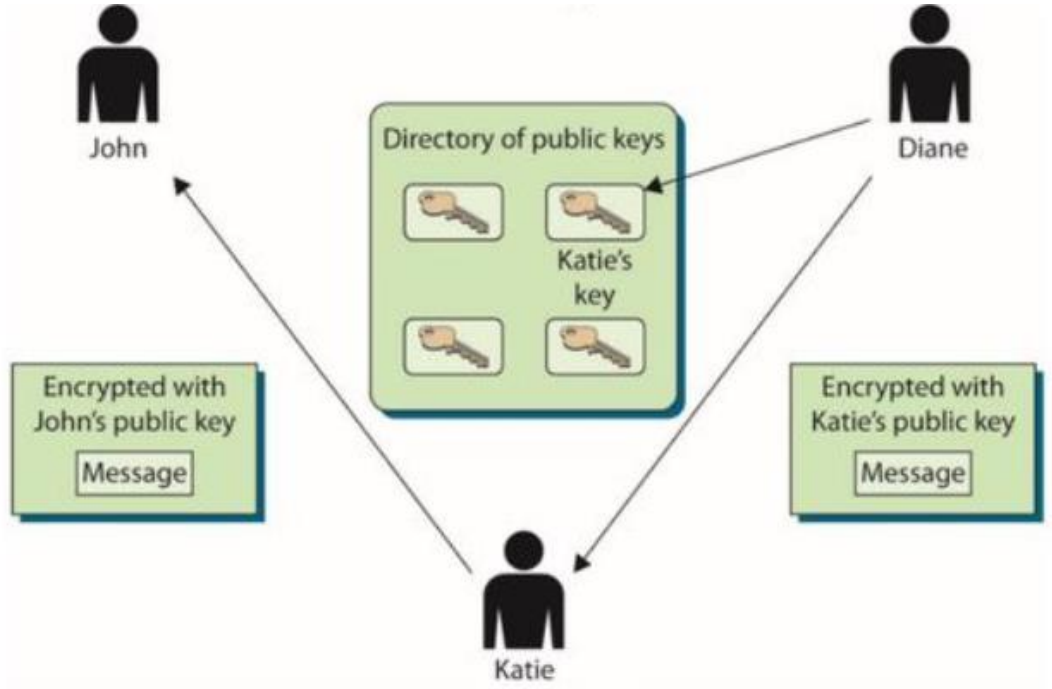
KEY TERMS

- Digital certificate
- Dual control
- End-entity certificate
- Hardware security module (HSM)
- Hierarchical trust model
- Hybrid trust model
- Internet Security Association and Key Management Protocol (ISAKMP)
- Key archiving
- Key escrow
- Key recovery
- Local registration authority (LRA)

KEY TERMS

- Online Certificate Status Protocol (OCSP)
- Peer-to-peer trust model
- Policy certificate
- Public key infrastructure (PKI)
- Registration authority (RA)
- X.509

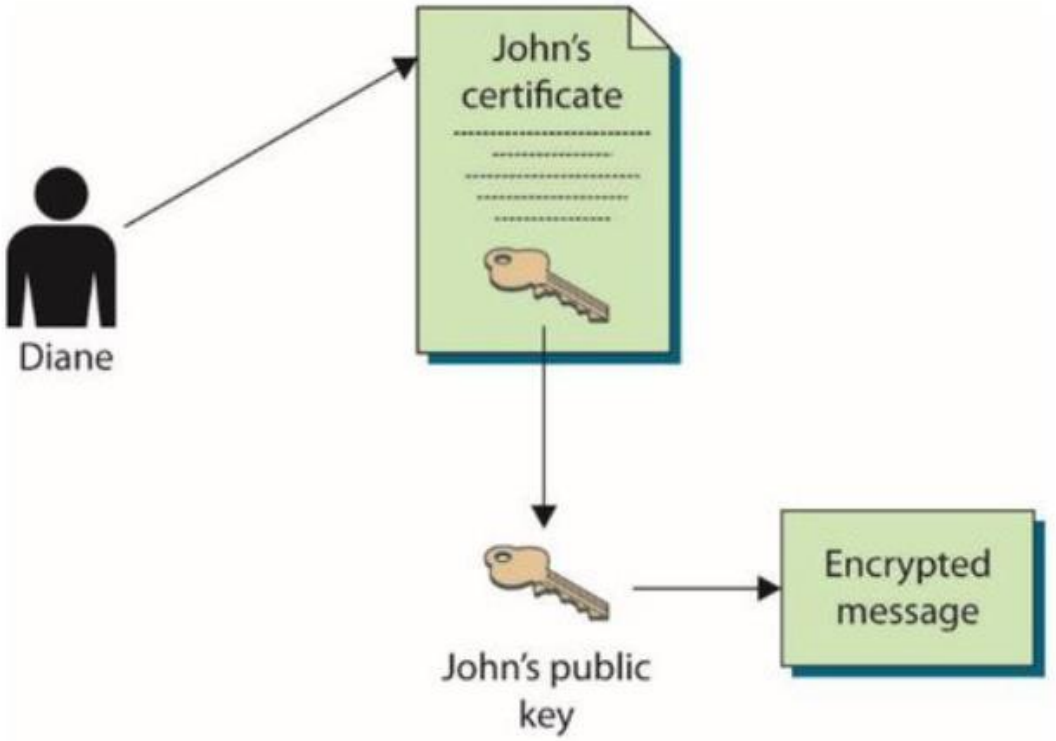
Without PKI, individuals could spoof others' identities



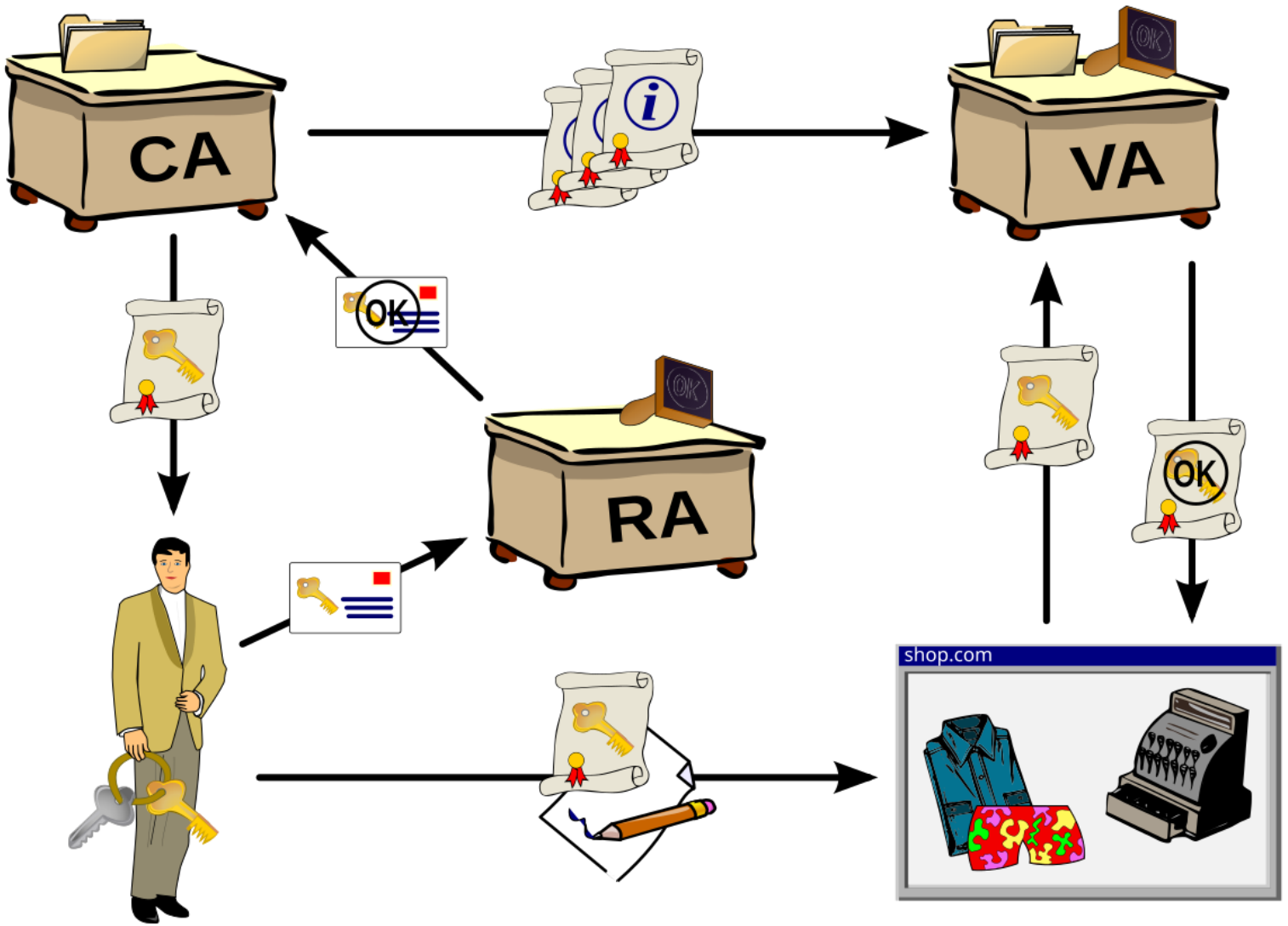
Third-party trust model

- RA requires proof of identity from the individual requesting a certificate and will validate this information.
- RA advises the CA to generate a certificate.
- CA digitally signs the certificate using its private key.
- The use of the CA's private key ensures that the certificate came from the CA.

Public keys are components of digital certificates



PKI Interaction Diagram



What does Infrastructure really mean?

1. Generating key-pairs and validating certificates does not a PKI make

2. No 3rd party trusted identifier → trust each other and/or the channel

3. PKI provides trust that you cannot / don't provide

4. **Infrastructure** – sustaining groundwork upon which other things can be built.

    1. Low level, predictable, uniform

    1. Supports high-level applications

Certificate Authorities

- A **certificate authority (CA)** is a trusted authority that certifies individuals' identities and creates electronic documents indicating that individuals are who they say they are.
- The electronic document is referred to as a **digital certificate**, and it establishes an association between the subject's identity and a public key.
- The private key that is paired with the public key in the certificate is stored separately.

Certificate Authorities

- If one CA component is compromised, it can negatively affect the CA integrity overall.
- Every CA should have a **certification practices statement** (**CPS**).
  - It outlines how identities are verified; the steps the CA follows to generate, maintain, and transmit certificates; and why the CA can be trusted to fulfill its responsibilities.
- A **certificate server** is the actual service that issues certificates based on the data provided during the initial registration process.

Registration Authorities

- A **registration authority (RA)** is the PKI component that accepts a request for a digital certificate and performs the necessary steps of registering and authenticating the person requesting the certificate.
- The authentication requirements differ depending on the type of certificate being requested.
- Most CAs offer a series of classes of certificates with increasing trust by class.

Local Registration Authorities

- A **local registration authority (LRA)** performs the same functions as an RA.
  - It is closer to the end users and reduces WAN traffic.
  - It is implemented in companies with their own internal PKIs and in companies with distributed sites
  - It performs identification, verification, registration functions; sends request, along with the user's public key, to a centralized CA so that the certificate can be generated.
  - It acts as an interface between the users and the CA.
  - LRAs simplify the RA/CA process for entities that desire certificates only for in-house use.

Public Certificate Authorities

- Public CAs are already established and being used by many other individuals and companies.
  - Specialize in verifying individual identities and creating and maintaining their certificates
  - Issue certificates that are not bound to specific companies or departments
- Examples of public CAs include:
  - VeriSign (including GeoTrust and Thawte), Entrust, and Go Daddy

Public Certificate Authorities

- Advantage of using a public CA is that it is usually well known and easily accessible to many people.
- Certificate policy (CP) allows the company to decide what certification classes are acceptable and how they will be used within the organization.

In-House Certificate Authorities

- An in-house CA is implemented, maintained, and controlled by the company that implemented it.
  - This type of CA can be used to create certificates for internal employees, devices, applications, partners, and customers.
  - This approach gives the company complete control over how individuals are identified, what certification classifications are created, who can and cannot have access to the CA, and how the certifications can be used.

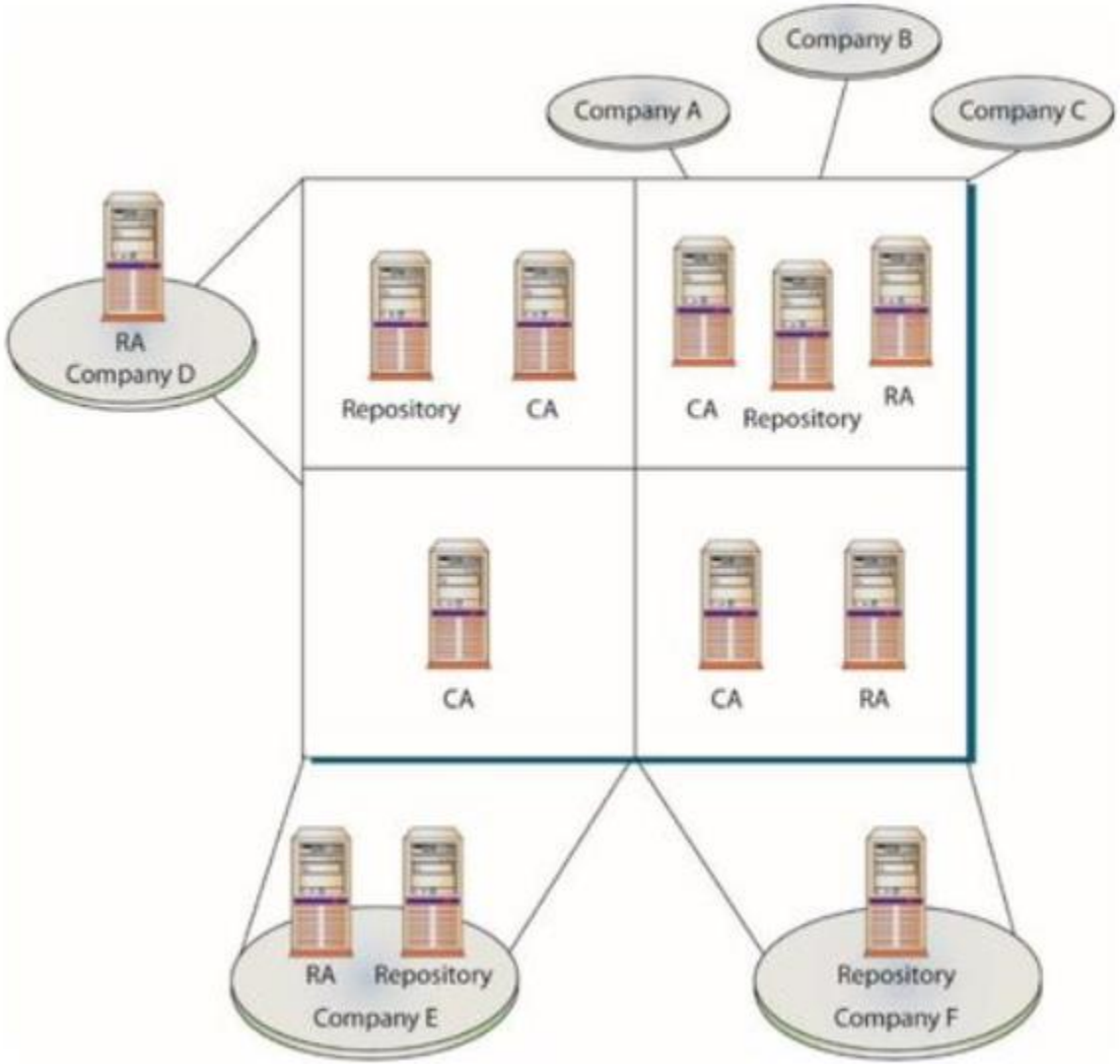
Choosing Between a Public CA and an In-House CA

- Factors need to be identified and taken into accounted.
  - Time and cost need to be considered.
  - Public CAs already have the necessary equipment, skills, and technologies.
  - Each company has various goals, security requirements, functionality needs, budgetary restraints, and ideologies.
  - Some companies do not trust an outside authority to generate and maintain their company's certificates.

Outsourced Certificate Authorities

- Usually, the more complex parts (the CA, RA, CRL, and key recovery mechanisms) are outsourced.
  - Level of trust the company willing to give to service provider and level of risk willing to accept must be determined.
  - Large vertical markets can have their own outsourced PKI environments set up to split costs and follow industry-specific standards.
  - A set of standards can be drawn up about how each different facility should integrate its own infrastructure and how it should integrate with the centralized PKI components.

Outsourced Certificate Authorities



Outsourced Certificate Authorities

- Offline server for security purposes
- Stapling is the process of combining related items to reduce communication steps.
- Pinning is the process of associating a host with a previously provided X.509 certificate or public key.
- Key continuity is the process of reusing a certificate or public key.

Pinning

- Process of associating a host with a previously provided X.509 certificate or public key.
  - Save the cert for later
- If pinned cert and host cert (from TLS) don't match → refuse to connect
- Protects against misissuance, CA compromise, man-in-the-middle

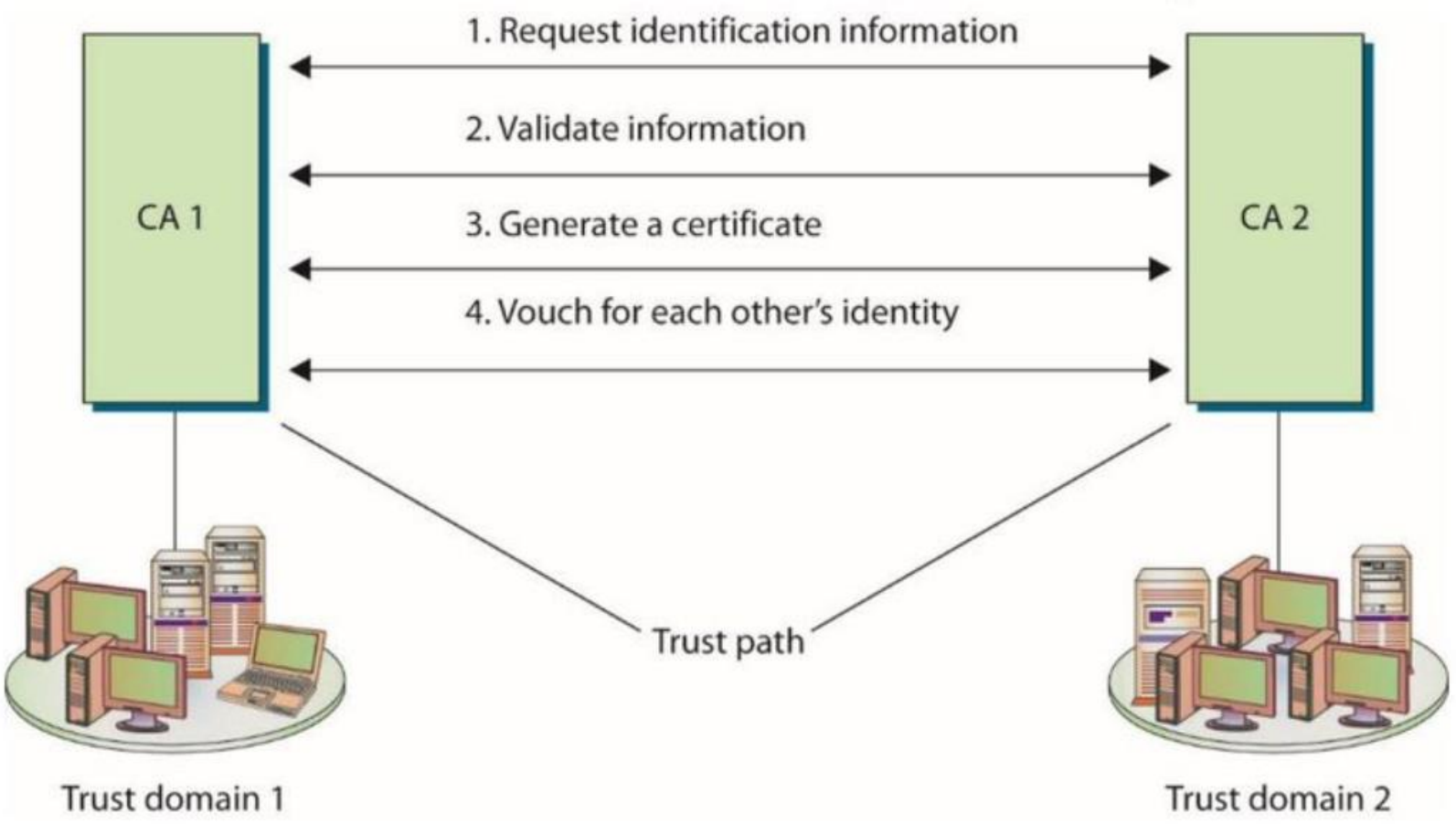
Trust Models

- A trust domain is a construct of systems, personnel, applications, protocols, technologies, and policies that work together to provide a certain level of protection.
  - Most trust domains need to communicate with other, less-trusted domains.
  - Must figure out how much two different domains should trust each other, and how to implement and configure an infrastructure that would allow these two domains to communicate in a way that will not allow security compromises or breaches.

Trust Anchor

- The trust anchor is the agreed-upon trusted third party.
  - Two separate trust domains involved if two companies need to communicate using their individual PKIs or two departments within the same company use different CAs.
    - The users and devices from these different trust domains need to communicate with each other.
    - They need to exchange certificates and public keys.
    - Trust anchors must be identified and a communication channel constructed and maintained.

Trust Relationship



Trust Models

- Trust models describe and outline the trust relationships between the different CAs and different environments.
  - Indicate where the trust paths reside
- Trust models and paths need to be thought out before implementation.
  - Restrict and control access properly
  - Ensure as few trust paths as possible are used
- Several different trust models can be used:
  - Hierarchical, peer-to-peer, and hybrid models

Trust Models

- A trust relationship must be established between two issuing authorities (CAs).
  - CA issues a certificate for the other CA's public key.
  - Each CA validates the other CA's identification info and generates a certificate containing a public key for that CA.
    - A trust path established between the two entities.
    - The trust path can be unidirectional or bidirectional – either the two CAs trust each other (bidirectional) or only one trusts the other (unidirectional).
- Certificate chain is a chain of trust

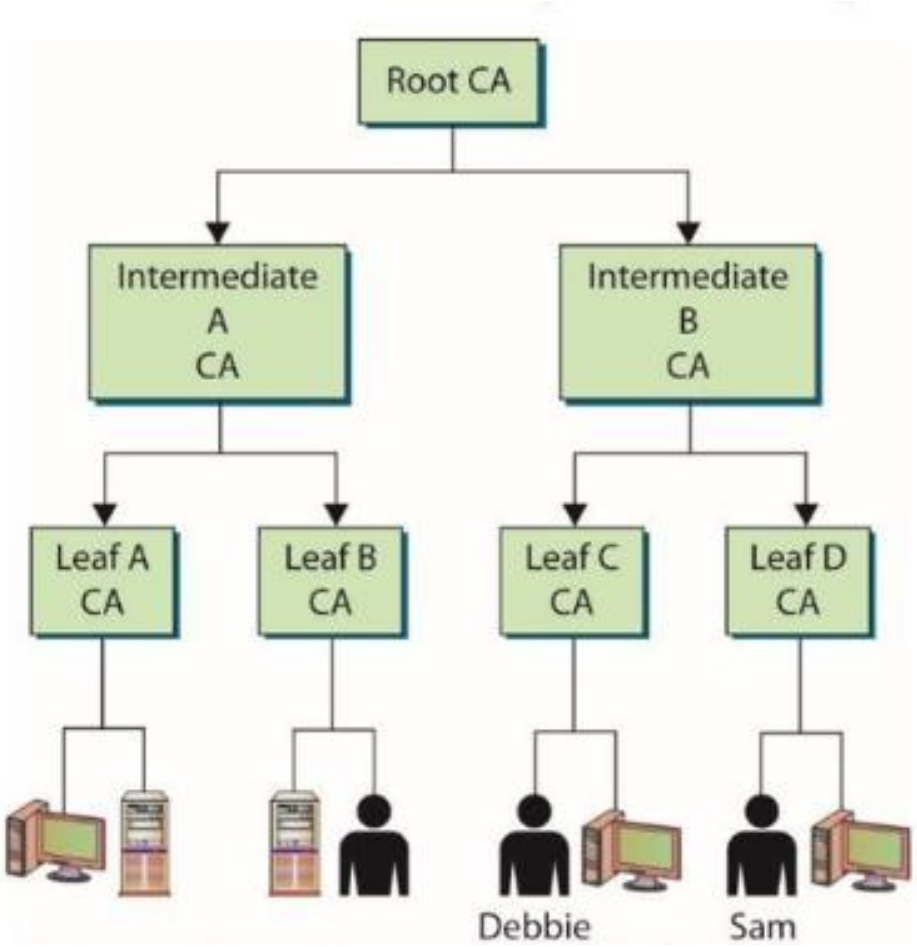
Certificate Chaining

- Certs bind identity to public key
- **Why trust a cert?**
  - Because of the chain
- Certificate chain is a chain of trust
- Chain/Intermediate certs sit in the middle
- Root cert sits at the top/end of the chain
  - Root cert self-signed by root CA
- Valid sigs + trusted root = trusted certs

Trust Models

- The **hierarchical trust model** is a basic hierarchical structure that contains a root CA, intermediate CAs, leaf CAs, and end-entities.
  - The configuration is that of an inverted tree.
  - The root CA is the ultimate trust anchor for all other entities in this infrastructure.
    - Root CA generates certificates for the intermediate CAs, which in turn generate certificates for the leaf CAs; leaf CAs generate certificates for the end-entities.
    - *Subordinate CAs* are subordinate to the CA they reference.

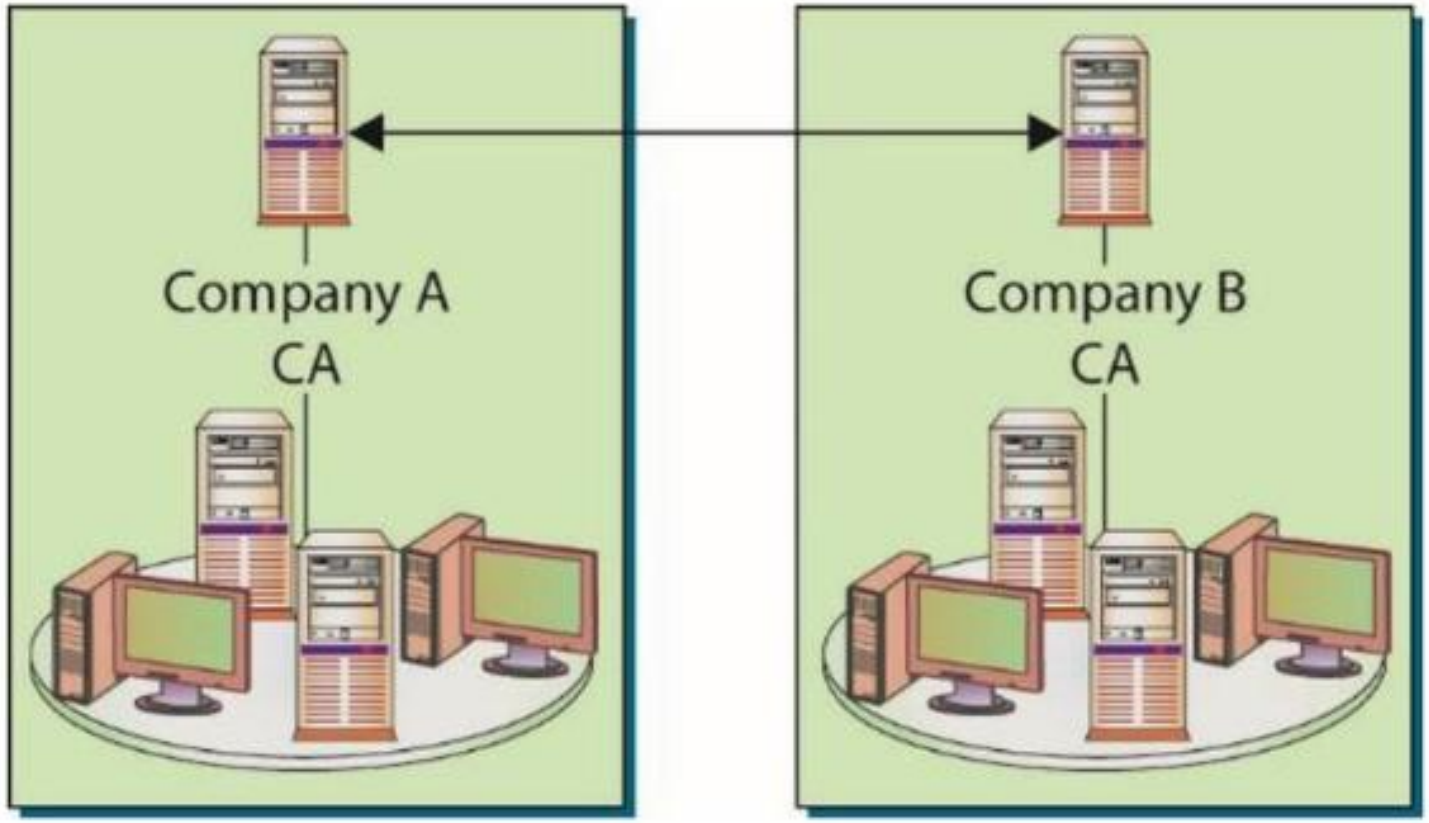
Hierarchical Trust Models



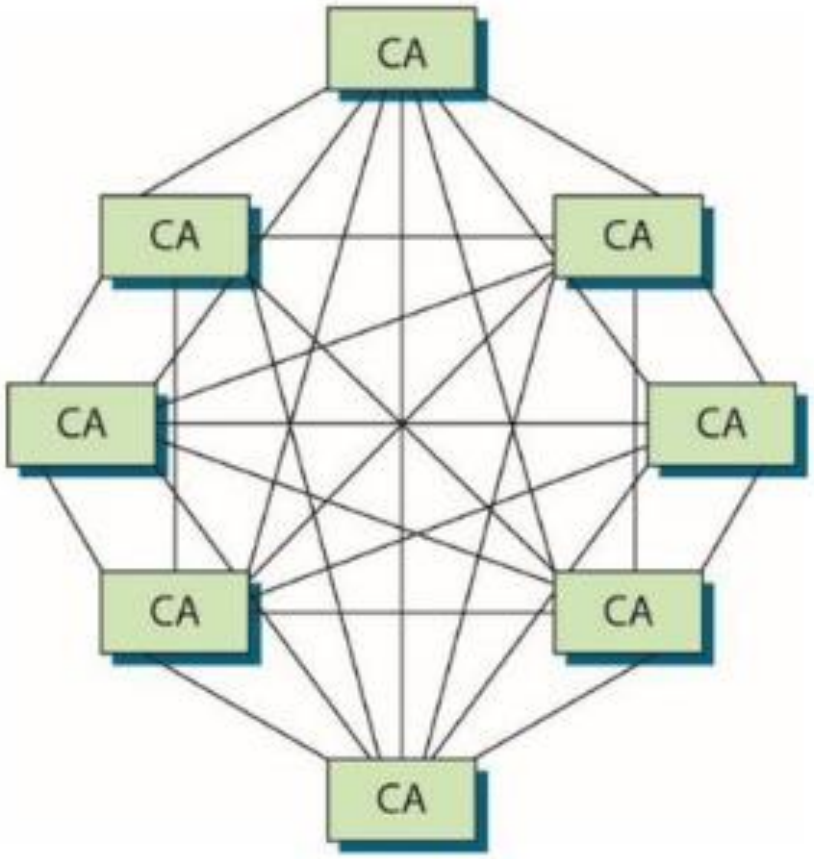
Trust Models

- In a **peer-to-peer trust model**, one CA is not subordinate to another CA, and no established trusted anchor between the CAs is involved.
  - The end-entities look to their issuing CA as their trusted anchor, but different CAs will not have a common anchor.
  - Cross-certification occurs when two different CAs certify the public key for each other creating a bidirectional trust.
  - Main drawback is scalability.
  - A fully connected mesh architecture has each CA directly connected to and has a bidirectional trust relationship with every other CA.

Trust Models



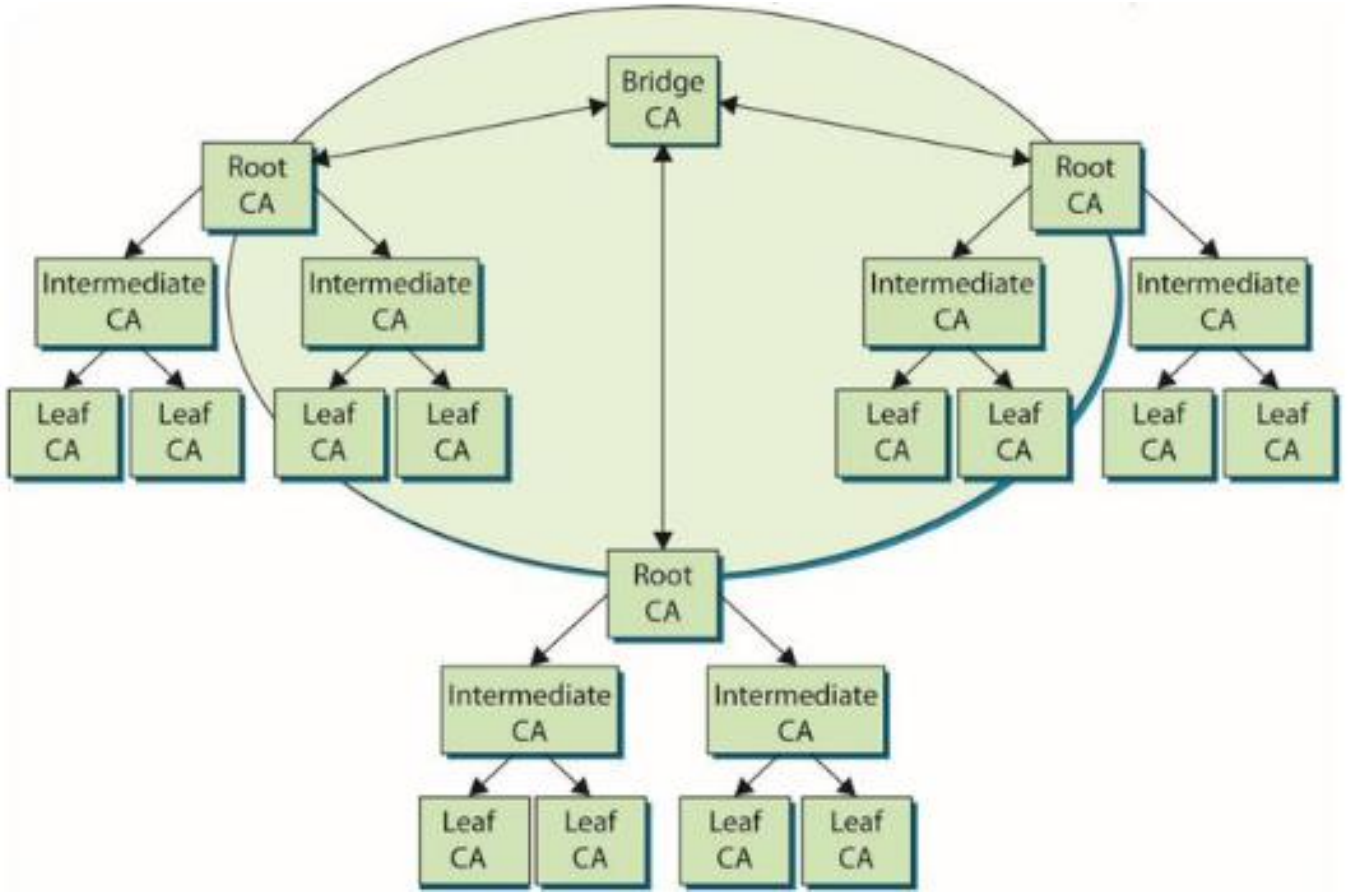
Trust Models



Trust Models

- In a **hybrid trust model**, the two companies have their own internal hierarchical models and are connected through a peer-to-peer model using cross-certification.
  - Another option is to implement a bridge CA responsible for issuing cross-certificates for all connected CAs and trust domains.
    - The bridge is not considered a root or trust anchor, but merely the entity that generates and maintains the cross-certification for the connected environments.

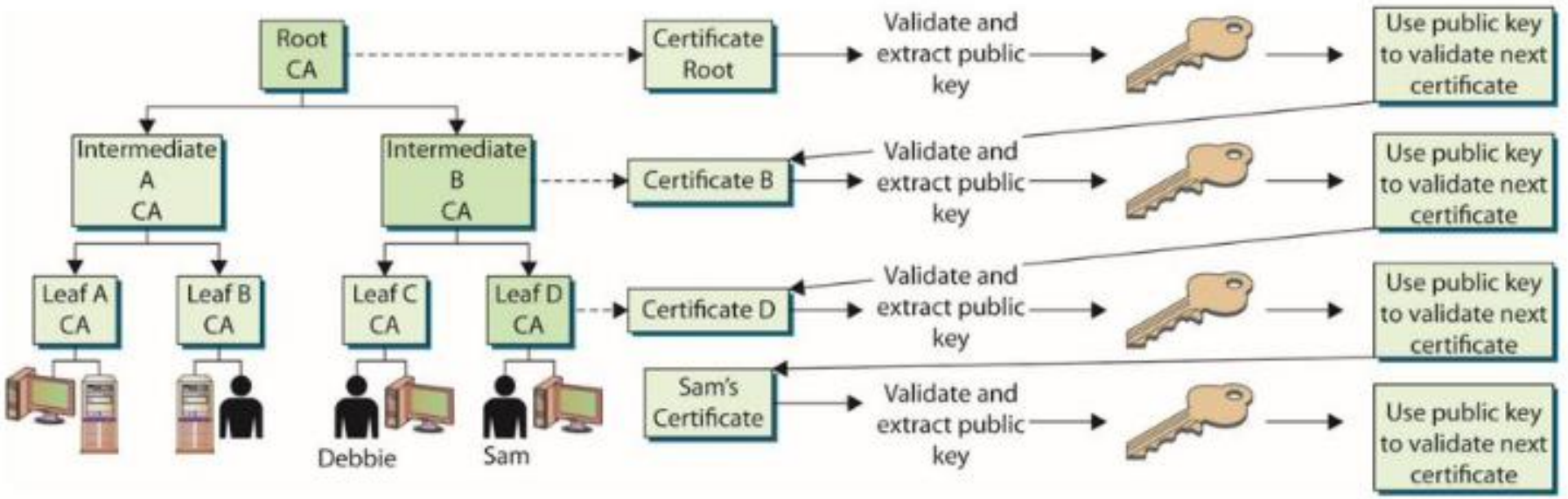
Trust Models



Trust Models

- Walking the certificate path
  - Following the **certificate path** means the client software had to continue to track down and collect certificates until it came upon a self-signed certificate.
  - A *self-signed certificate* indicates that it was signed by a root CA.
  - Simplistic trust model works well within an enterprise that easily follows a hierarchical organizational chart.
    - Many companies cannot use this trust model because different departments or offices require their own trust anchors.

Walking the certificate path



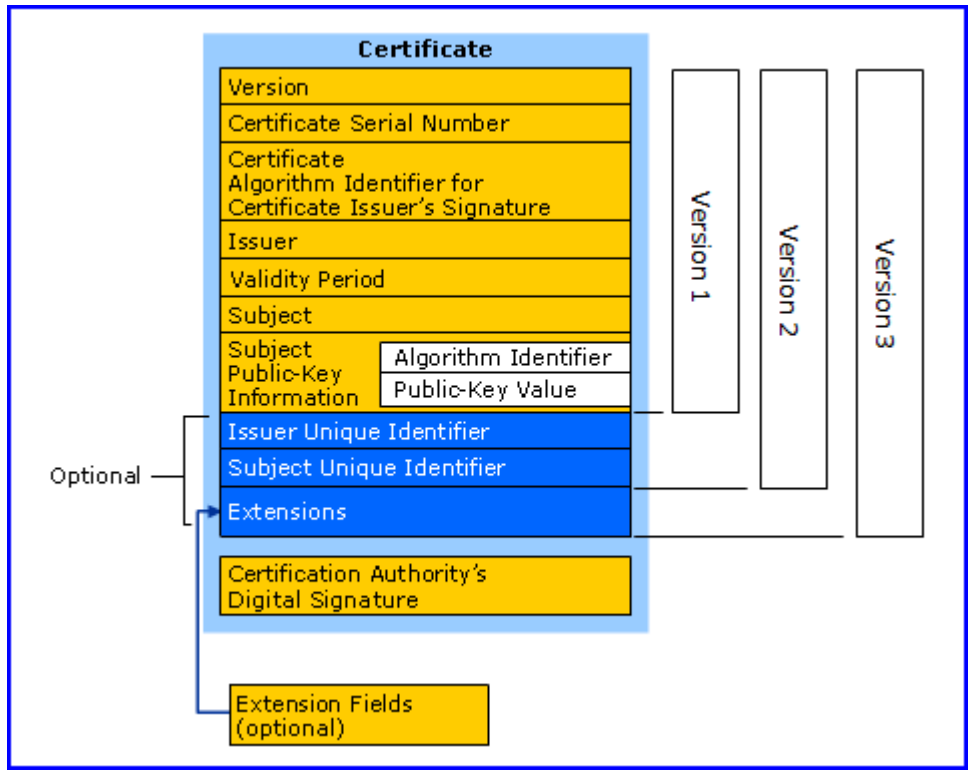
Digital Certificates

- A digital certificate binds an individual's identity to a public key.
  - It contains information a receiver needs to be assured of the identity of the public key owner.
  - It is created and formatted based on the **X.509 standard**.
    - Outlines necessary fields of a certificate and the possible values that can be inserted into the fields
    - Most current version: X.509 version 3, a standard of the International Telecommunication Union (International Telecommunication Union)

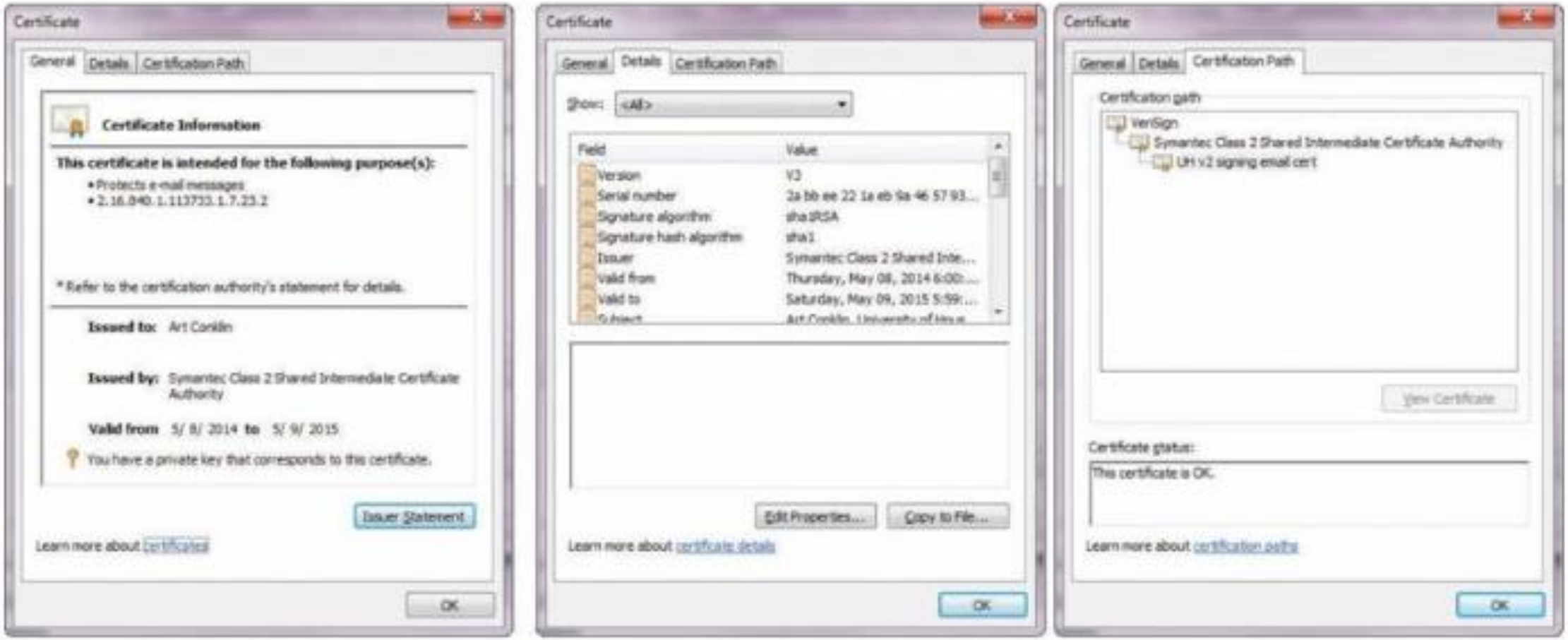
Digital Certificates

Table 7.1 X.509 Certificate Fields

Field Name	Field Description
Certificate Version	X.509 version used for this certificate: Version 1 = 0 Version 2 = 1 Version 3 = 2
Serial Number	A nonnegative integer assigned by the certificate issuer that must be unique to the certificate.
Signature Algorithm Parameters (optional)	The algorithm identifier for the algorithm used by the CA to sign the certificate. The optional Parameters field is used to provide the cryptographic algorithm parameters used in generating the signature.
Issuer	Identification for the entity that signed and issued the certificate. This must be a distinguished name within the hierarchy of CAs.
Validity Not valid before time Not valid after time	Specifies a period of time during which the certificate is valid, using a "not valid before" time and a "not valid after" time (expressed in UTC or in a generalized time).
Subject	The name for the certificate owner.
Subject Public Key Info	An encryption algorithm identifier followed by a bit string for the public key.
Issuer Unique ID	Optional for versions 2 and 3. This is a unique bit-string identifier for the CA that issued the certificate.
Subject Unique ID	Optional for versions 2 and 3. This is a unique bit-string identifier for the subject of the certificate.
Extensions Extension ID Critical Extension Value	Optional for version 3. The extensions area consists of a sequence of extension fields containing an extension identifier, a Boolean field indicating whether the extension is critical, and an octet string representing the value of the extension. Extensions can be defined in standards or defined and registered by organizations or communities.
Thumbprint Algorithm Algorithm Parameters (optional)	Identifies the algorithm used by the CA to sign this certificate. This field must match the algorithm identified in the Signature Algorithm field.
Thumbprint	The signature is the bit-string hash value obtained when the CA signed the certificate. The signature certifies the contents of the certificate, binding the public key to the subject.



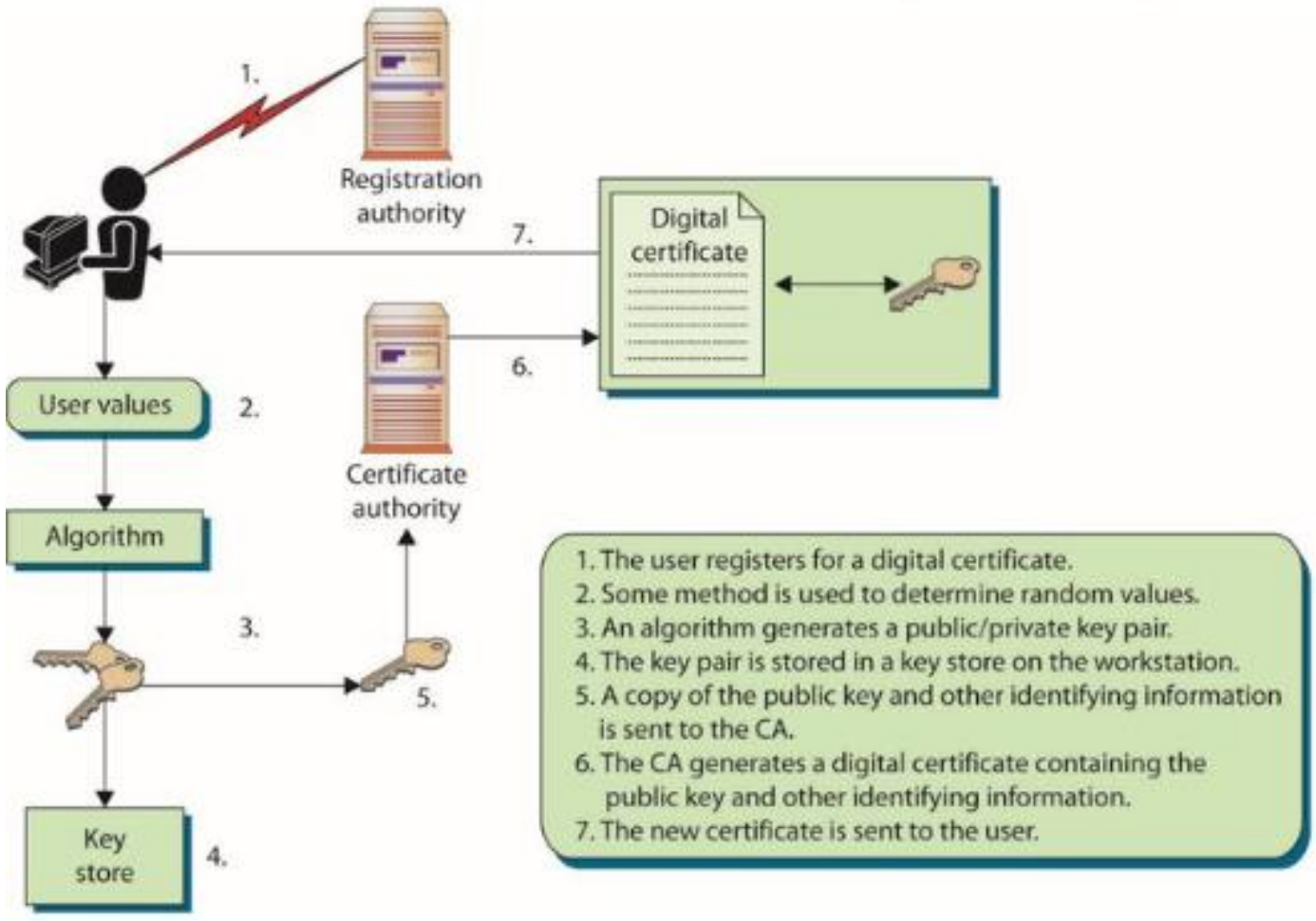
Digital Certificates



Certificates Classes

- Class 1 to verify an individual's identity through e-mail. Can use public/private key pair to digitally sign e-mail and encrypt message contents.
- Class 2 for software signing. A software vendor would register for this type of certificate so that it could digitally sign its software.
- Class 3 used by a company to set up its own CA, which will allow it to carry out its own identification verification and generate certificates internally.

Certificates Classes



Certificates Extensions

- Certificate extensions allow for further information to be inserted within the certificate.
  - Extensions provide more functionality in a PKI implementation.
  - Standard certificate extensions are implemented for every PKI implementation.
  - Private certificate extensions are defined for specific organizations (or domains within one organization), and they allow companies to further define different, specific uses for digital certificates to best fit their business needs.

Certificates Extensions

- Key usage extensions dictate how the public key that is held within the certificate can be used.
  - Public keys are used for different functions:
    - Symmetric key encryption, data encryption, verifying digital signatures, and more.
  - Nonrepudiation service can be provided by third-party notary.
  - A trusted time source can be a trusted third party called a time stamp authority (TSA).
    - Gives users a higher level of confidence as to *when* specific messages were digitally signed.

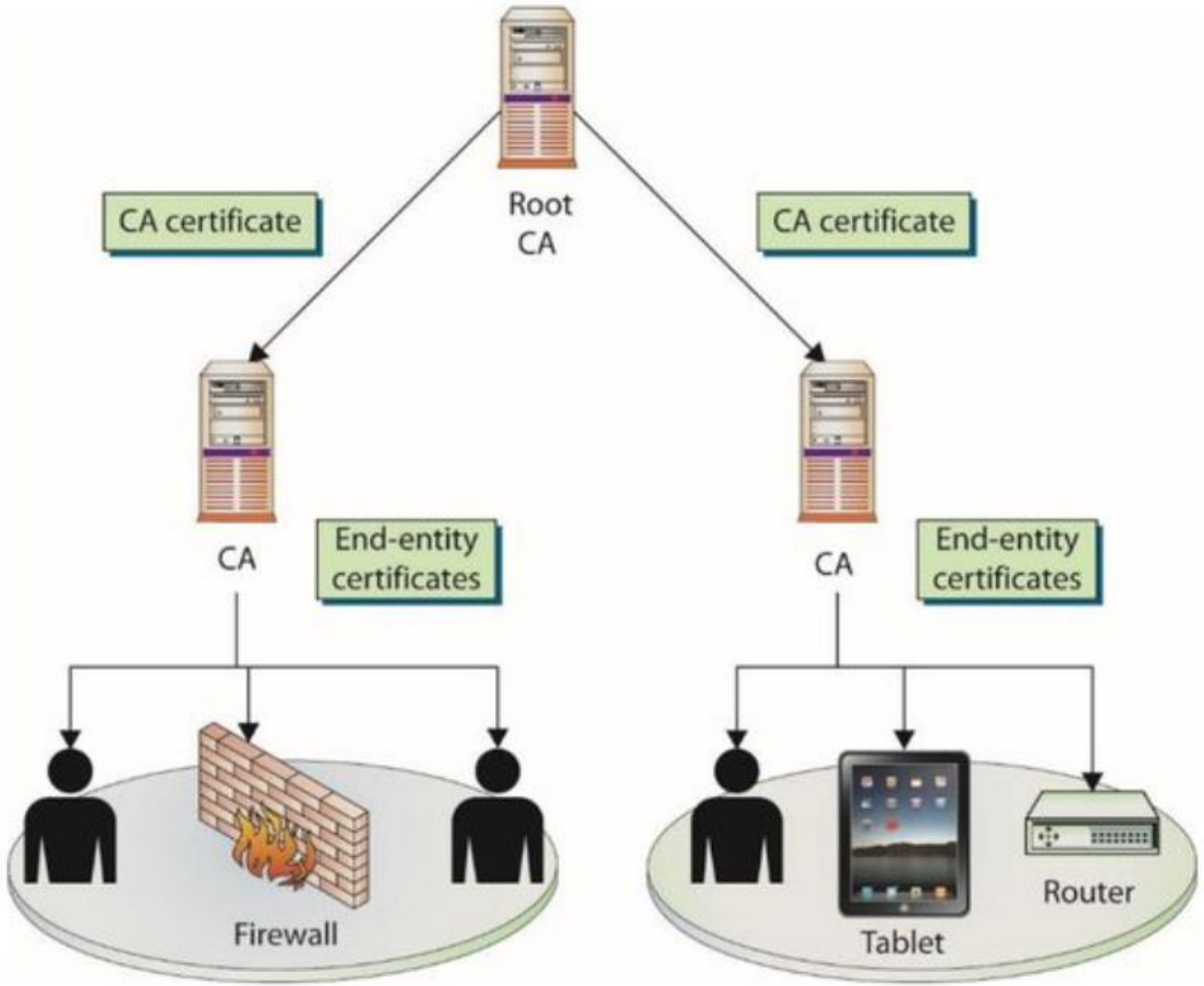
Certificates Extensions

- Critical and noncritical extensions are indicated by a specific flag within the certificate itself.
  - Flag set to *critical* indicates that the extension *must* be understood and processed by the receiver.
  - If flag set to *noncritical*, the certificate can be used for the intended purpose, even if the receiver does not process the appended extension.
- Object Identifiers (OID)
  - Each extension to a certificate has its own ID, expressed as an object identifier, which is a set of values, together with either a critical or noncritical indication.

Certificate Attributes

- Four main certificate types
  - **End-entity certificates** – issued by a CA to a specific subject
  - **CA certificates** – self-signed, in the case of a standalone or root CA, or issued by a superior CA within a hierarchical model
  - **Cross-certification certificates** (*cross certificate*) – used when independent CAs establish peer-to-peer trust relationships
  - **Policy certificates** – used for high-security applications where a mechanism is required to provide centrally controlled policy information to PKI clients

Certificate Attributes



Certificate Attributes

- Wildcard certificates to multiple entities
  - For example, *.example.com works for one.example.com and two.example.com
- Subject Alternative Name (SAN) is a field (extension) in a certificate that has several uses.
- Code signing allows managing certificates for specific functions and reducing the risk in the event of compromise.
- Self-signed: highest node (root) on a given trust chain

Certificate Attributes

- Active Directory Domain Services (AD DS) can keep track of machines in a system via machines identifying themselves using *machine certificates*.
- Root certificate is the name given to a certificate that forms the initial basis of trust in a trust chain.
- Domain validation is a low-trust means of validation based on control over Transport Layer Security (TLS)
- Extended validation (EV) certificates are used for HTTPS websites and software to provide a high level of assurance as to the originator's identity.

Certificate Formats

- **DER**: One of the Abstract Syntax Notation One (ASN.1) encoding rules that can be used to encode any data object into a binary file.
- **PEM**: The most common format used by certificate authorities when issuing certificates.
- **CER/CRT**: Certificate files.
  - **CER** is used mostly for Windows
  - **CRT** for UNIX
- **KEY**: A .key file can be used for both public and private PKCS#8 keys.
- **PFX**: A PKCS#12 file with a .pfx extension; a portable file format.
- **P12**: An alternative file extension for a PKCS#12 file format (same as PFX).
- **P7B**: The PKCS#7 or P7B format is stored in Base64 ASCII format and has an extension of .p7b or .p7c.

Certificate Lifecycles

- Keys and certificates should have lifetime settings.
  - Forces the user to register for a new certificate after a certain amount of time
- There are tradeoff in determining the proper length.
  - Shorter lifetimes limit the ability of attackers to crack them.
  - Longer lifetimes lower system overhead.
- More-sophisticated PKI implementations perform automated and often transparent key updates.
  - Avoids the time and expense of having users register for new certificates when old ones expire

Registration and Generation

- A key pair (public and private keys) is generated locally by an application and can be stored in a local key store on the user's workstation.
- A key pair created by a central key-generation server requires secure transmission of the keys to the user.
- Key pair created on the centralized server can be stored on the user's workstation or on the user's smart card
  - Allows for more flexibility and mobility

Registration and Generation

- The act of verifying that an individual indeed has the corresponding private key for a given public key is referred to as proof of possession.
- Key regeneration and replacement is usually done to protect against threats.
- The PKI administrator usually configures the minimum required key size that users must use to have a key generated for the first time, and then for each renewal.

Certificate signing request

- A **certificate signing request (CSR)** is the actual request to a CA containing a public key and the requisite information needed to generate a certificate.
- The CSR contains all of the identifying information that is to be bound to the key by the certificate generation process.

Renewal

- The certificate itself has its own lifetime.
  - It can be different from the key pair's lifetime.
  - The certificate's lifetime is specified by the validity dates inserted into the digital certificate.
    - These are beginning and ending dates indicating the time period during which the certificate is valid.
  - The certificate cannot be used before the start date, and once the end date is met, the certificate is expired and a new certificate will need to be issued.

Renewal

- A renewal process is different from the registration phase.
  - The RA assumes the individual has already successfully completed one registration round.
  - If the certificate has not actually been revoked, the original keys and certificate can be used to provide the necessary authentication information and proof of identity for the renewal phase.
  - The certificate may or may not need to change during the renewal process; this usually depends on why the renewal is taking place.

Suspension

- Instead of being revoked, a certificate can be *suspended*, meaning it is temporarily put on hold.
- Reasons to suspend
  - Extended vacation – ensure certificate will not be compromised or used during that time
  - Suspicion that a private key might have been compromised

Revocation

- A certificate can be revoked when its validity needs to be ended before its actual expiration date is met.
  - Lost laptop or a smart card that stored a private key
  - Improper software implementation directly affecting the security of a private key
  - Social engineering attack obtained a private key
  - Data within certificate no longer applies to the individual
  - Employee left a company
- Certificate revocation is permanent and final—once revoked a certificate cannot be reinstated.

Certificate Revocation List (CRL)

- A **certificate revocation list (CRL)** is a list of serial numbers of certificates that have been revoked.
  - The CRL contains a statement indicating why the individual certificates were revoked and a date when the revocation took place.
  - The list usually contains all certificates that have been revoked within the lifetime of the CA.
  - Certificates that have expired are not the same as those that have been revoked.
  - An expired certificate means that its end validity date was reached.

CRL Reason Codes

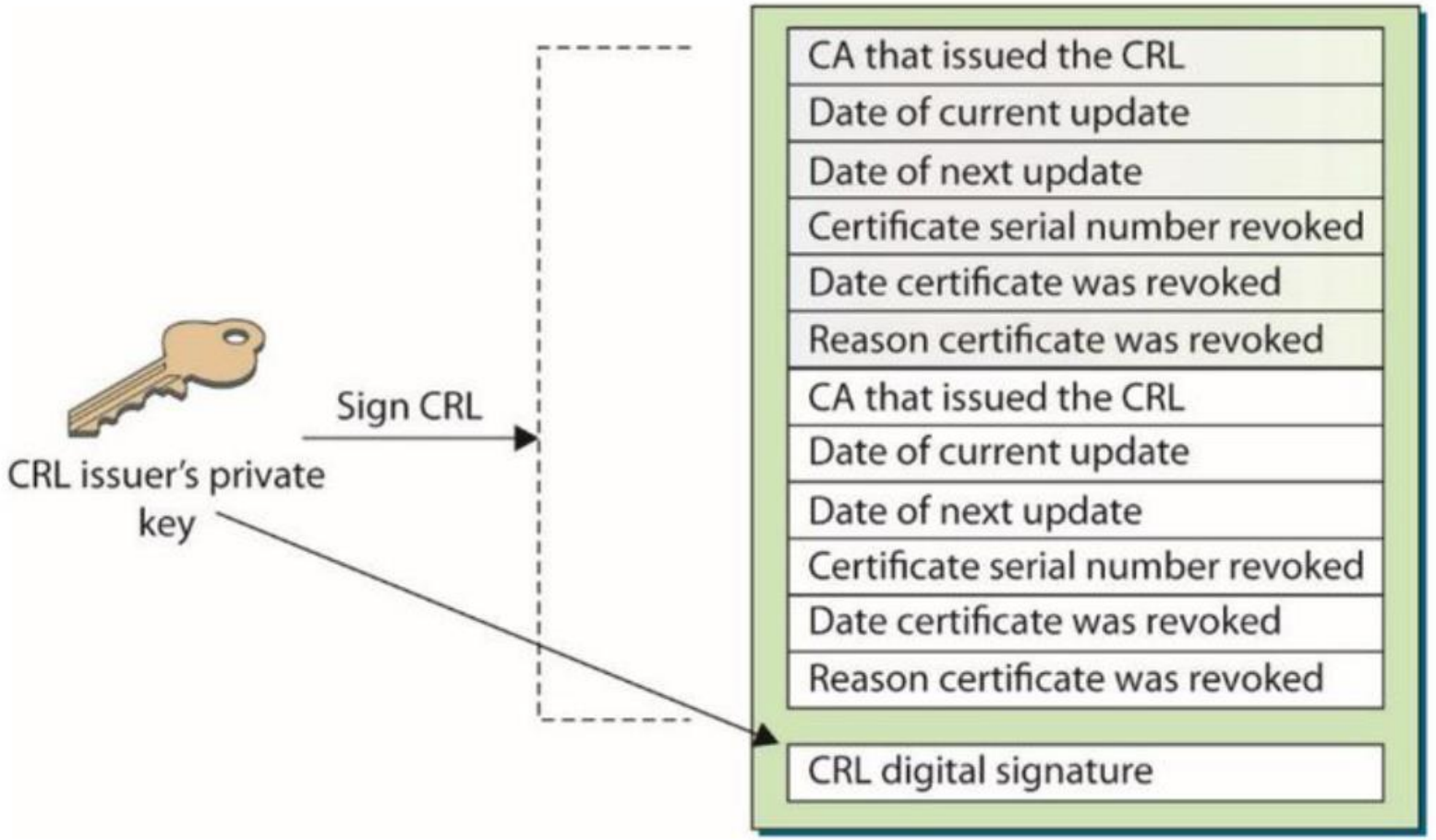
X.509v3

reasonCode	Identifier	Description
0	unspecified	Can be used to revoke certificates for reasons other than the specific codes. (default)
1	keyCompromise	It is known or suspected that the subject's private key, or other aspects of the subject validated in the certificate, have been compromised.
2	cACompromise	The certificate authority that issued this certificate was compromised, which means all of the certificates it has ever issued are now compromised.
3	affiliationChanged	The subject's name or other information in the certificate has been modified but there is no cause to suspect that the private key has been compromised.
4	superceded	The certificate has been replaced but there is no cause to suspect that the private key has been compromised.
5	cessationOfOperation	The certificate is no longer needed for the purpose for which it was issued but there is no cause to suspect that the private key has been compromised.
6	certificateHold	Certificate is suspended.
7		Not used.
8	removeFromCRL	Used with delta CRL to indicate a CRL entry should be removed (unsuspended)
9	privilegeWithdrawn	The certificate was revoked because a privilege contained within that certificate has been withdrawn.
10	aACompromise	It is known or suspected that aspects of the attribute authority validated in the attribute certificate have been compromised.

X.509v2

Reason Code	*Reason*
0	*Unspecified*
1	*All keys compromised; indicates compromise or suspected compromise*
2	*CA compromise; used only to revoke CA keys*
3	*Affiliation changed; indicates a change of affiliation on the certificate*
4	*Superseded; the certificate has been replaced by a more current one*
5	*Cessation; the certificate is no longer needed, but no reason exists to suspect it has been compromised*
6	*Certificate hold; indicates the certificate will not be issued at this point in time*
7	*Remove from CRL; used with delta CRL to indicate a CRL entry should be removed*

The CA digitally signs the CRL to protect its integrity



CRL Distribution

- CRL distribution
  - CRLs can be requested by individuals or pushed down periodically.
  - CRLs can grow substantially in size.
    - The smaller the better to reduce load on resources.
    - Updates may be only changes to CRL.
  - A certificate might have an extension that points the validating user to the necessary *CRL distribution point*.

Online Certificate Status Protocol (OCSP)

- **Online Certificate Status Protocol (OCSP)**
  - It is used for online revocation services.
  - It is a request and response protocol that obtains the serial number of the certificate being validated and reviews revocation lists for the client.
  - The protocol has a responder service that reports the status of the certificate back to the client.
    - Indicates revoked, valid, or unknown status
  - This protocol and service saves the client from having to find, download, and process the right lists.

Key Destruction

- Key pairs and certificates have set *lifetimes*
  - They will expire at some specified time.
  - It is important that the certificates and keys are properly destroyed when that time comes, wherever the keys are stored.
  - The goal is to make sure that no one can gain access to a key after its lifetime has ended and use that key for malicious purposes.

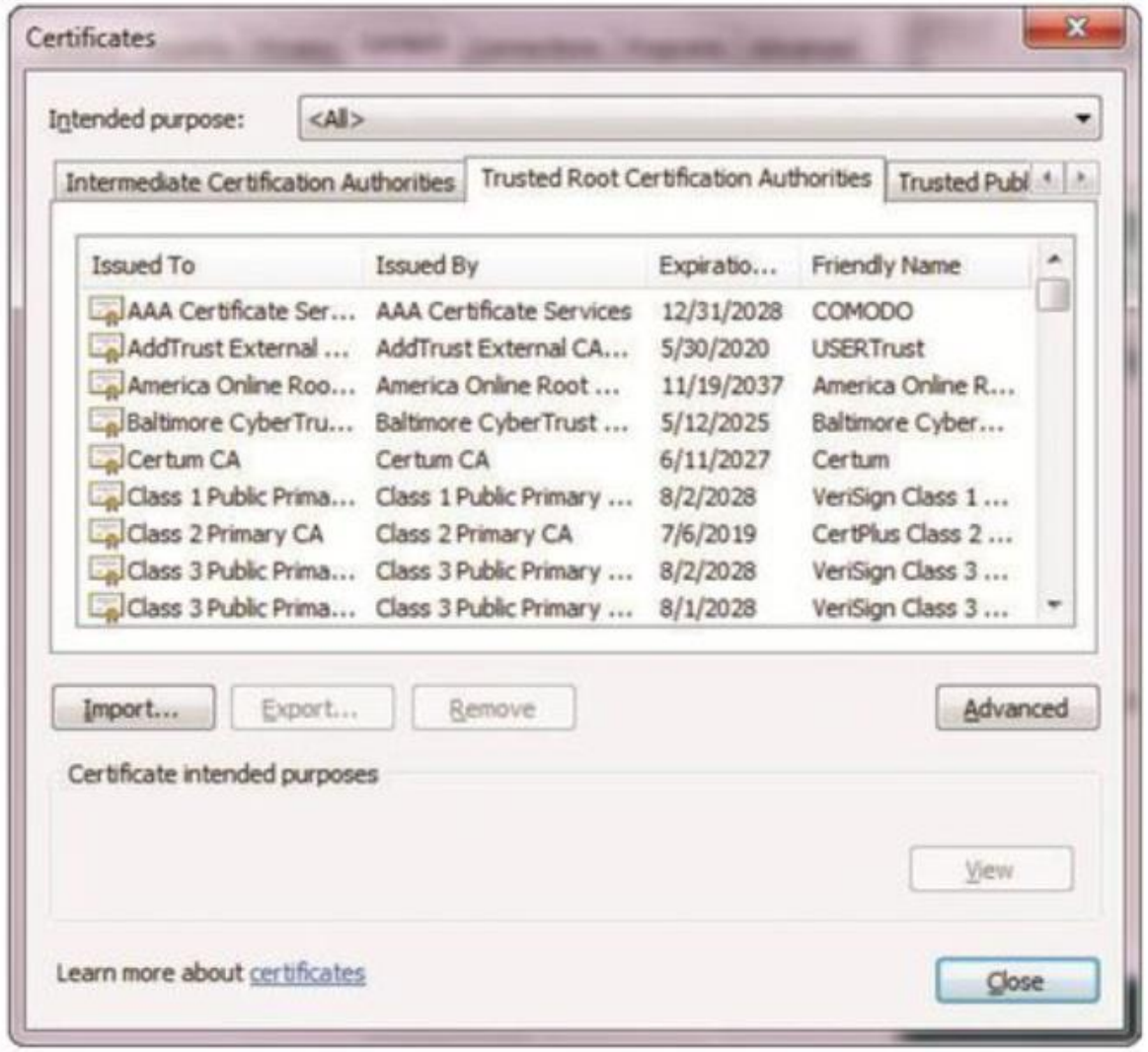
Certificate Repositories

- A **certificate repository** is a centralized directory that can be accessed by a subset of individuals.
  - Directories are usually Lightweight Directory Access Protocol (LDAP)-compliant.
    - Can be accessed and searched via an LDAP query from an LDAP client.
  - A certificate repository is a holding place for individuals' certificates and public keys that are participating in a particular PKI environment.
- Different applications from the same vendor may share key stores

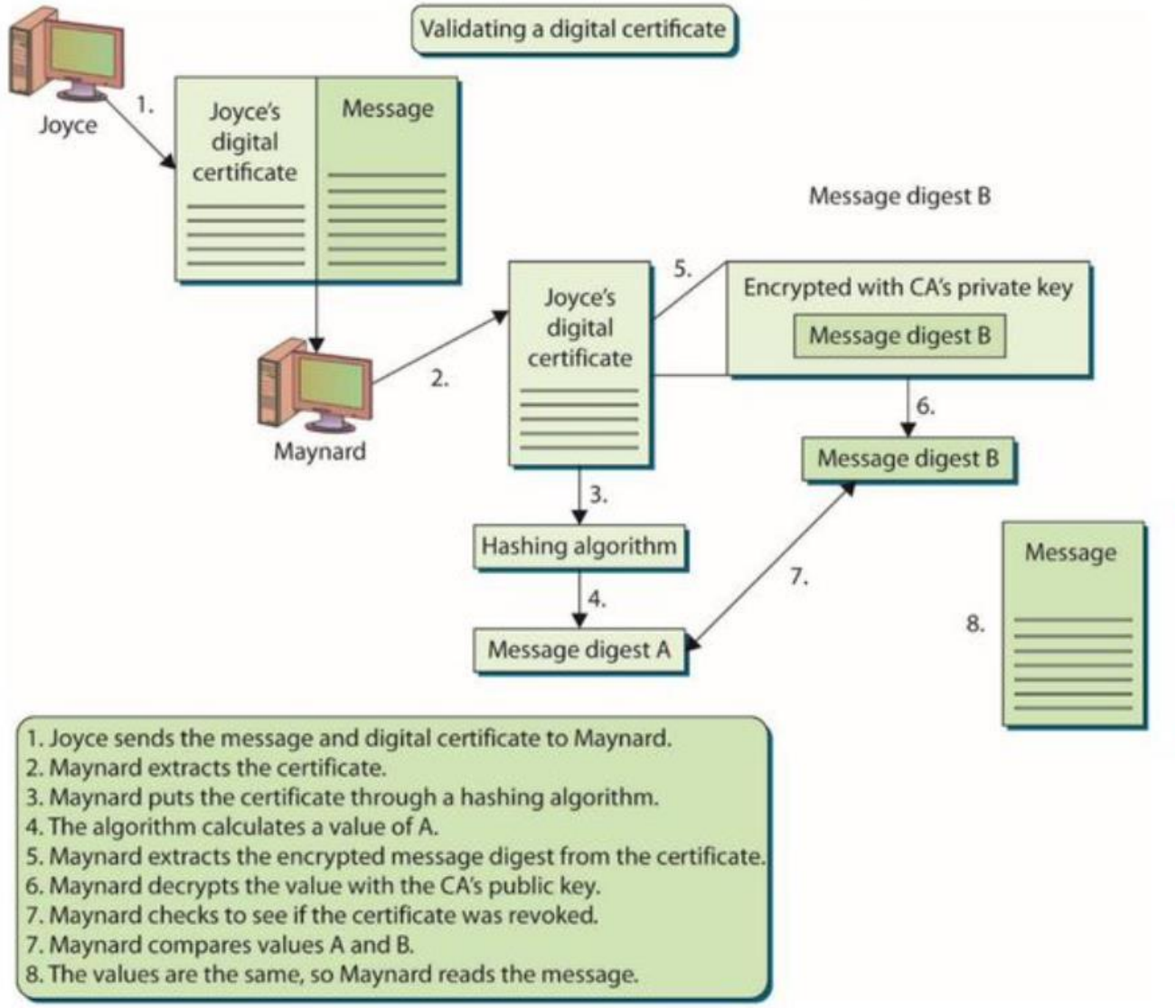
Trust Certificate Verification

- Use PKI if you do not automatically trust individuals you do not know.
- Use a trusted third party to vouch for the other individual so confidence can be instilled and sensitive communication can take place.
- When a user trusts a CA, she will download that CA's digital certificate and public key and store them on her local computer.
- Trusted certificate authorities can be found in your browser's list.

Trust Certificate Verification



Trust Certificate Verification



Centralized and Decentralized Infrastructures

- In a decentralized approach, software on individual computers generates and stores cryptographic keys local to the systems themselves.
- In a centralized infrastructure, the keys are generated and stored on a central server, and the keys are transmitted to the individual systems as needed.
  - Easier to back up the keys and implement key recovery procedures

Centralized and Decentralized Infrastructures

- Centralized infrastructure disadvantages
  - Technology is needed to send keys in an encrypted manner.
    - Encryption ensures keys' integrity, and ensure intended user is receiving the key.
  - Fault tolerance or redundancy mechanism is needed for server that centrally stores the keys.
  - To provide *true* authenticity and nonrepudiation, digital signature public/private key pair should not be generated at a centralized server.

Hardware Security Modules

- A **hardware security module (HSM)** is a physical device that safeguards cryptographic keys.
  - HSMs enable a higher level of security for the use of keys, including generation and authentication.
  - In most situations, HSM solutions are used only for the most critical and sensitive keys, which are the root key and possibly the intermediate CA private key.
  - If those keys are compromised, the whole security of the PKI is gravely threatened.

Private Key Protection

- The private key needs to stay private.
  - Digital signature created solely for the purpose of proving who sent a particular message by using a private key.
  - *Key store* is a storage area for the private key.
    - Key store usually created by the application registering for a certificate.
    - Many applications do not require strong password to protect the key store.
  - If digital signatures will be used for legal purposes, these points and others may need to be audited to ensure that true authenticity and nonrepudiation are provided.

Private Key Protection

- The key size should provide the necessary level of protection for the environment.
- The lifetime of the key should correspond with how often it is used and the sensitivity of the data it is protecting.
- The key should be changed at the end of its lifetime and not used past its allowed lifetime.
- Where appropriate, the key should be properly destroyed at the end of its lifetime.
- The key should never be exposed in clear text.
- No copies of the private key should be made if it is being used for digital signatures.
- The key should not be shared.
- The key should be stored securely.
- Authentication should be required before the key can be used.
- The key should be transported securely.
- Software implementations that store and use the key should be evaluated to ensure they provide the necessary level of protection.

Key Recovery

- Two systems are important for backing up and restoring cryptographic keys:
  - **Key archiving** is a way of backing up keys and securely storing them in a repository.
  - **Key recovery** is the process of restoring lost keys to the users or the company.
  - If keys are backed up and stored in a centralized computer, this system must be tightly controlled.
  - It is usually unwise to authorize a single person to be able to recover all the keys within the environment.

Key Recovery

- Dual control approach to key recovery is referred to as the *m of n authentication*.
  - Number of people involved in key recovery process is *n*.
  - At least *m* (smaller than *n*) must be involved before the task can be completed
  - Too many people for *m* increases issues associated with availability; too few increases risk of a small number of people colluding to compromise a secret.
  - Goal is to minimize fraudulent or improper use of access and permissions.
  - All key recovery procedures should be highly audited.

Key Recovery

- Dual control can be used as part of a system to back up and archive data encryption keys.
  - PKI systems can be configured to require multiple individuals in any key recovery process.
  - *Separation of duties* means that one person cannot complete a critical task by himself.
  - Requiring two individuals to recover a lost key together is called **dual control**.
    - This means two people must be present to carry out a specific task.

Key Escrow

- **Key escrow** is the process of giving keys to a third party so that they can decrypt and read sensitive information if the need arises.
  - Almost always pertains to handing over encryption keys to the government, or to another higher authority, so keys can be used to collect evidence during investigations.
  - Two reasons employer may escrow a key pair:
    - Keys are property of the enterprise, issued to the worker for use.
    - The firm may have need for them after an employee leaves the firm.

Certificate-Based Threats

- Attacks prey on user false sense of security.
  - Industry responded with a high-assurance certificate that is signed and recognized by browsers.
  - Forging a false certificate is challenging because of the public key signing of certificates by CAs.
  - Hacker must install a false, self-signed root certificate.
  - This attack preys on the fact that end users do not know the contents of their root certificate store, nor do they have a means to validate changes.
  - Attack thwarted by locking down the certificate store and validating against a white list.

Stolen Certificates

- Stolen certificates used in multiple cases of computer intrusions/system attacks.
- Specially crafted malware designed to steal both private keys and digital certificates from machines.
  - Zeus bot has functionality to perform this task.
  - The Stuxnet attack on the Iranian nuclear production facility used stolen certificates from third parties.
  - After the Sony Pictures Entertainment attack became public in 2014, malware using Sony certificates appeared.

PKIX Standards

- Using X.509 v3, the PKIX working group addresses five major areas:
  - PKIX outlines certificate extensions and content not covered by X.509 v3 and the format of version 2 CRLs, thus providing compatibility standards for sharing certificates and CRLs between CAs and end-entities in different PKIs.

PKIX Standards

- Using X.509 v3, the PKIX working group addresses five major areas (continued):
  - PKIX provides certificate management message formats and protocols, defining the data structures, management messages, and management functions for PKIs

PKIX Standards

- Using X.509 v3, the PKIX working group addresses five major areas (continued):
  - PKIX outlines certificate policies and certification practices statements (CPSs), establishing the relationship between policies and CPSs.

PKIX Standards

- Using X.509 v3, the PKIX working group addresses five major areas (continued):
  - PKIX specifies operational protocols, defining the protocols for certificate handling.
  - PKIX includes time-stamping and data certification and validation services, which are areas of interest to the PKIX working group, and which will probably grow in use over time.
- PKIX working group has been working on two other types of certificates: Attribute Certificates and Qualified Certificates.

PKCS

- RSA Laboratories created the Public Key Cryptography Standards (PKCS) to fill some of the gaps in the standards that existed in PKI implementation.
- Why You Need to Know the PKIX and PKCS Standards
  - If your company is planning to use one of the existing certificate servers to support e-commerce, you might not need to know the specifics of these standards

ISAKMP

- **Internet Security Association and Key Management Protocol (ISAKMP)**
  - Provides a method for implementing a key exchange protocol and for negotiating a security policy
  - Defines procedures and packet formats to negotiate, establish, modify, and delete security associates.
- A security association (SA) is a relationship in which two or more entities define how they will communicate securely.

CMP

- The PKIX Certificate Management Protocol (CMP) is specified in RFC 4210
  - Defines the messages and operations required to provide certificate management services within the PKIX model.
  - Provides a set of certificate operations.

XKMS

- The XML Key Management Specification defines services to manage PKI operations within the Extensible Markup Language (XML) environment.
  - Intended to simplify integration of PKIs and management of certificates in applications.
  - Services reside on a separate server that interacts with an established PKI.

XKMS

- Three tiers of service are based on the client requests and application requirements.
  - Tier 0 provides a means of retrieving key information by embedding references to the key within the XML signature
  - With tier 1 operations, the client forwards the key-information portions of the XML signature to the XKMS server, relying on the server to perform the retrieval of the desired key information
  - In tier 2, the XKMS server is actively involved in verifying the relation between the PKI information and the document containing the XML signature.

CEP

- Certificate Enrollment Protocol (CEP) was originally developed by VeriSign for Cisco Systems
  - Designed to support certificate issuance, distribution, and revocation using existing technologies
  - Operations supported include CA and RA public key distribution, certificate enrollment, certificate revocation, certificate query, and CRL query.
  - Uses both PKCS #7 (Cryptographic Message Syntax Standard) and PKCS #10 (Certification Request Syntax Standard) to define a common message syntax