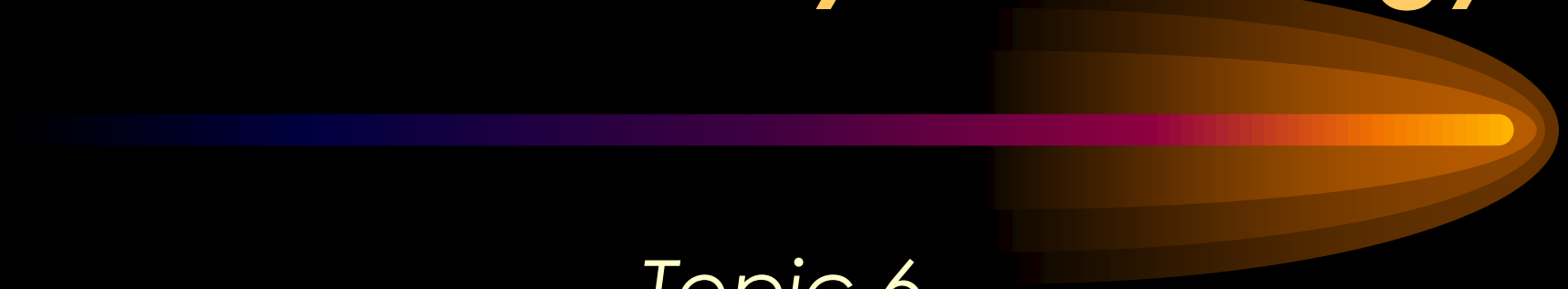
# *Info Security Technology*



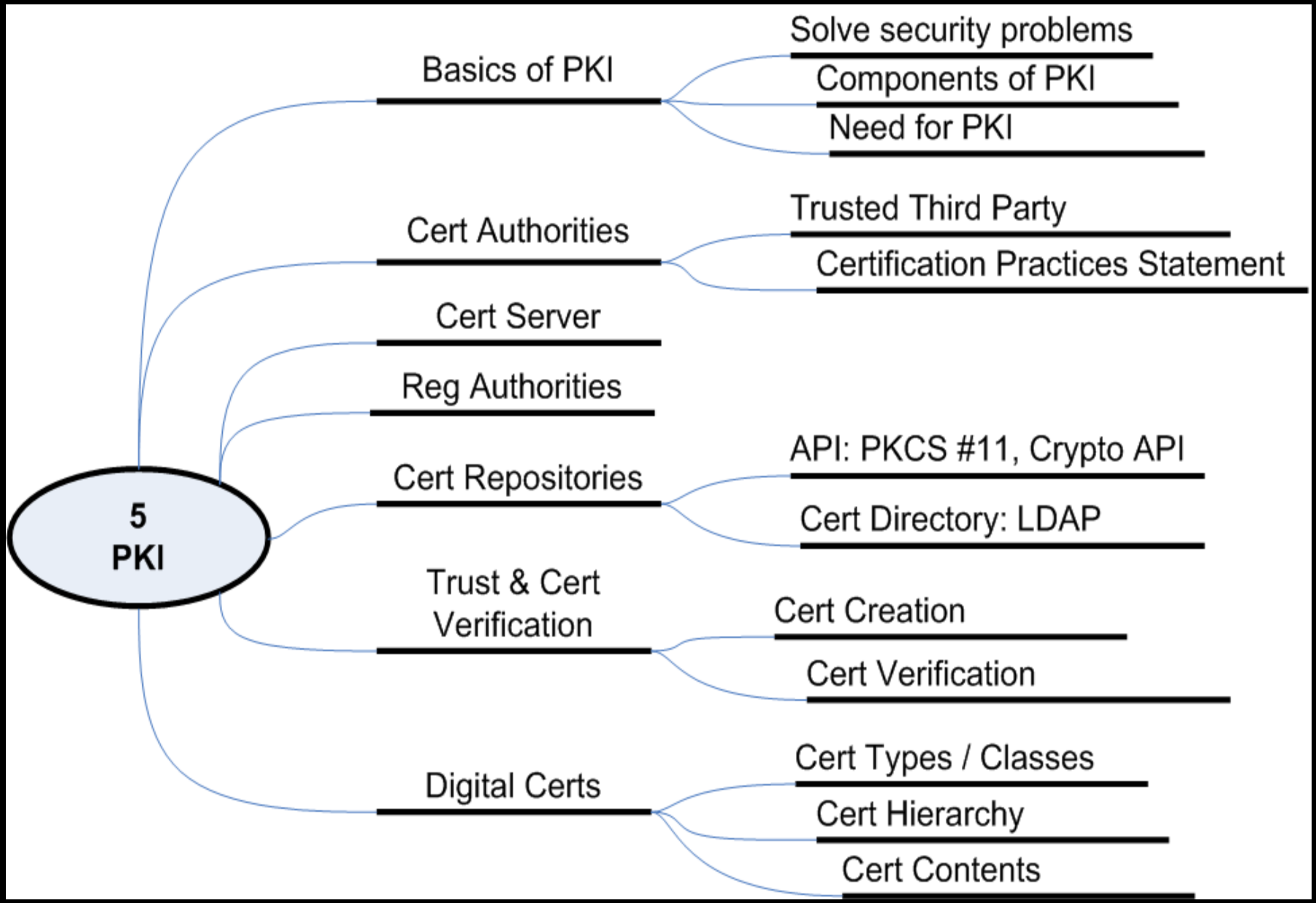
*Topic 6*

*PKI*

*(Public Key Infrastructure)*

# Overview

- Public Key Infrastructure
  - The Basics of PKI
  - Certificate Authorities
  - Registration Authorities
  - Certificate Repositories
  - Trust and Certificate Verification
  - Digital Certificates


5
PKI
Basics of PKI
Solve security problems
Components of PKI
Need for PKI
Cert Authorities
Trusted Third Party
Certification Practices Statement
Cert Server
Reg Authorities
Cert Repositories
API: PKCS #11, Crypto API
Cert Directory: LDAP
Trust & Cert Verification
Cert Creation
Cert Verification
Digital Certs
Cert Types / Classes
Cert Hierarchy
Cert Contents

# *Basics of Public Key Infrastructures*

- **Public key infrastructures (PKIs)** are central security foundation for organizations.

- It provides the underlying mechanisms that offer **confidentiality**, **integrity**, **authentication**, **and nonrepudiation**.

# *Basics of Public Key Infrastructures*

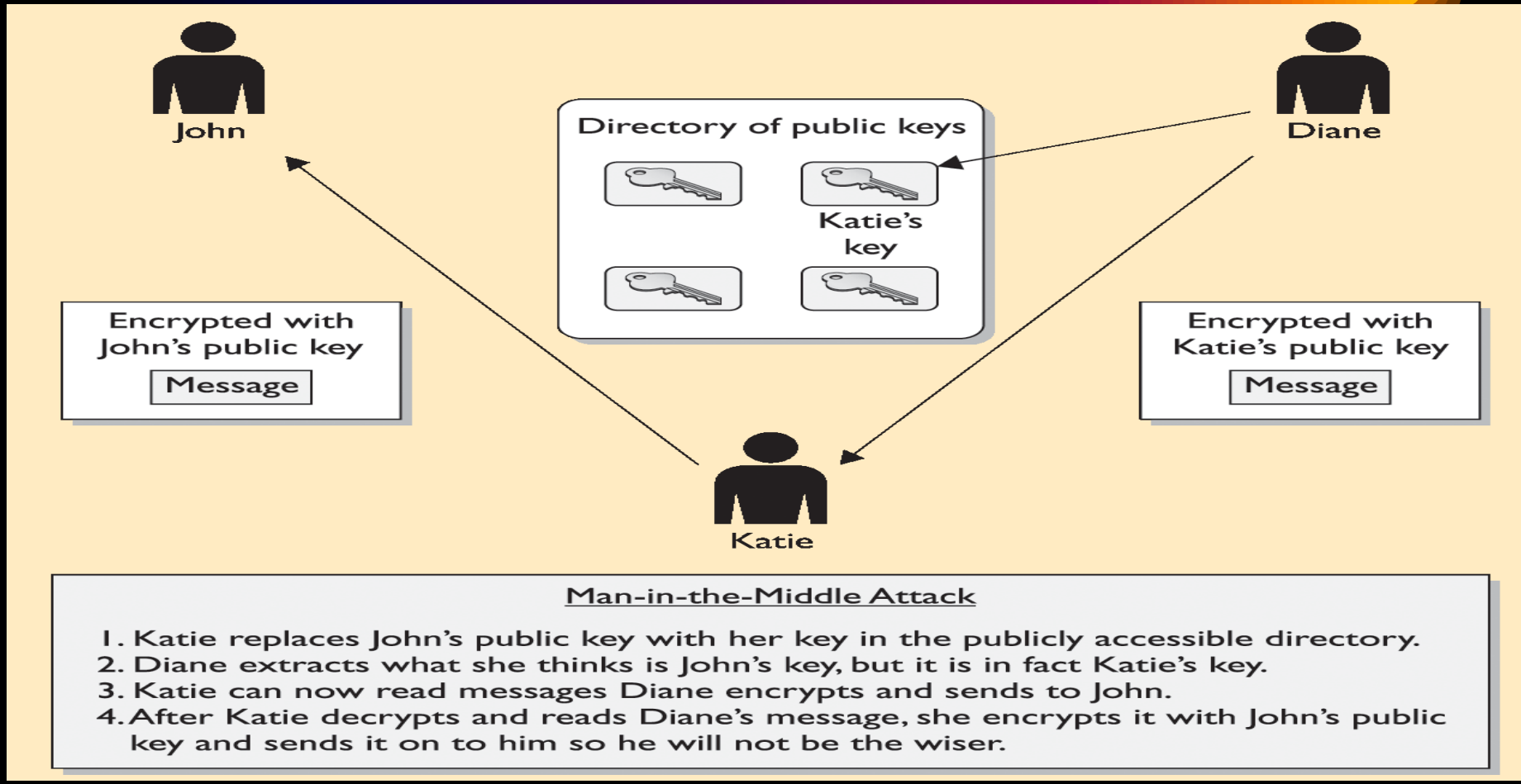
- PKI is made up of:
  - Hardware
  - Applications
  - Policies
  - Services
  - Programming interfaces
  - Cryptographic algorithms
  - Protocols
  - Users
  - Utilities

# *Basics of Public Key Infrastructures*

- PKI involves entities called registration authorities and certificate authorities.
  - They require proof of identity from the **individual** requesting a certificate and validate this information.
  - The **registration authority** then advises the certificate authority (CA) to generate a certificate, which is analogous to a driver's license.
  - The **certificate authority** digitally signs the certificate using its private key.

- This is commonly referred to as a **third-party trust model**.

# Example 6.1




Without a PKI, individuals could spoof each other's identities.

# Example 6.1

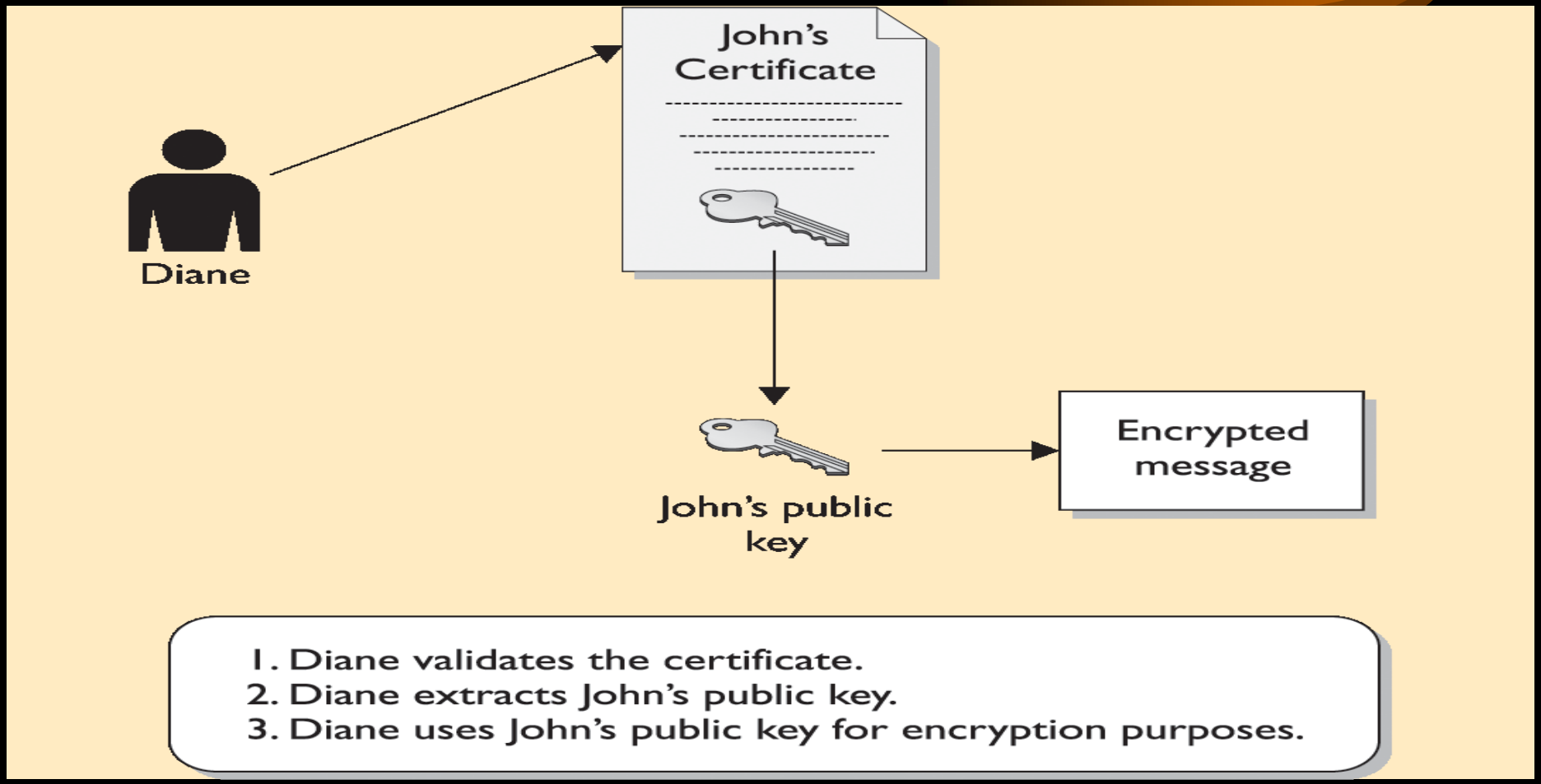
- **John and Diane want to communicate securely.**
- John can generate his own public/private key pair and send his public key to Diane or place it in a directory that is available to everyone.

- **If Diane receives John's public key, either from him or from a public directory, how does she know it really came from John?**
  - Perhaps an individual is masquerading as John and has replaced John's public key with their own.
- If this took place, Diane would believe that her messages could be read only by John and that the replies were actually from him.
- However, she would really be communicating with Katie.

# *Example 6.1*

- **What is needed** is a way
  - to verify an individual's identity,
  - to ensure that the public key is bound to the person's identity,
  - and thus ensure that the previous scenario does not take place.

# Example 6.2




Public keys are components of digital certificates

# *Example 6.2*

- This process allows John to **authenticate himself to Diane** and communicate with her through encryption **without prior communication or a preexisting relationship**.

- Once Diane is convinced of the **legitimacy of John's public key**, she can use it to encrypt and decrypt messages between them.

# *Certificate Authorities*

- The **certificate authority (CA)** is the trusted authority for certifying an individual's identity and creating an electronic document indicating that individuals are who they claim to be.
  - The electronic document is referred to as a **digital certificate**.
  - 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*

- The CA is made up of the software, hardware, procedures, policies, and people who are involved in validating identities and generating certificates.

- If one of these components is compromised, it negatively affects the CA and can threaten the integrity of the certificates it produces.

# CPS

- Every CA should have a **certification practices statement (CPS)**, which outlines:
  - How **identities** are verified.
  - The **steps** the CA follows to generate, maintain, and transmit certificates.
  - Why the CA can be **trusted** to fulfill its responsibilities.

- In addition, it describes:
  - How **keys** are secured.
  - What **data** is placed within a digital certificate.
  - How **revocations** will be handled.

# *Trust and CA*

- A critical aspect of a PKI is the **trust between the users and the CA**.
  - The CPS should be reviewed and understood to ensure that this level of trust is warranted.

# *Certificate Server*

- The **certificate server** is the actual service that issues certificates based on the data provided during the initial registration process.
- The server constructs and populates the **digital certificate** with the necessary information and combines the user's public key with the resulting certificate.
- The certificate is **digitally signed** with the CA's private key.

# *Registration Authority*

- The **registration authority (RA)** is the component that accepts a request for a digital certificate.
- The registration authorities perform the necessary steps of **registering and authenticating** the person requesting a certificate.
- The **types of certificates** available can vary between different CAs, but there are usually at least three different types, and they are referred to as classes.

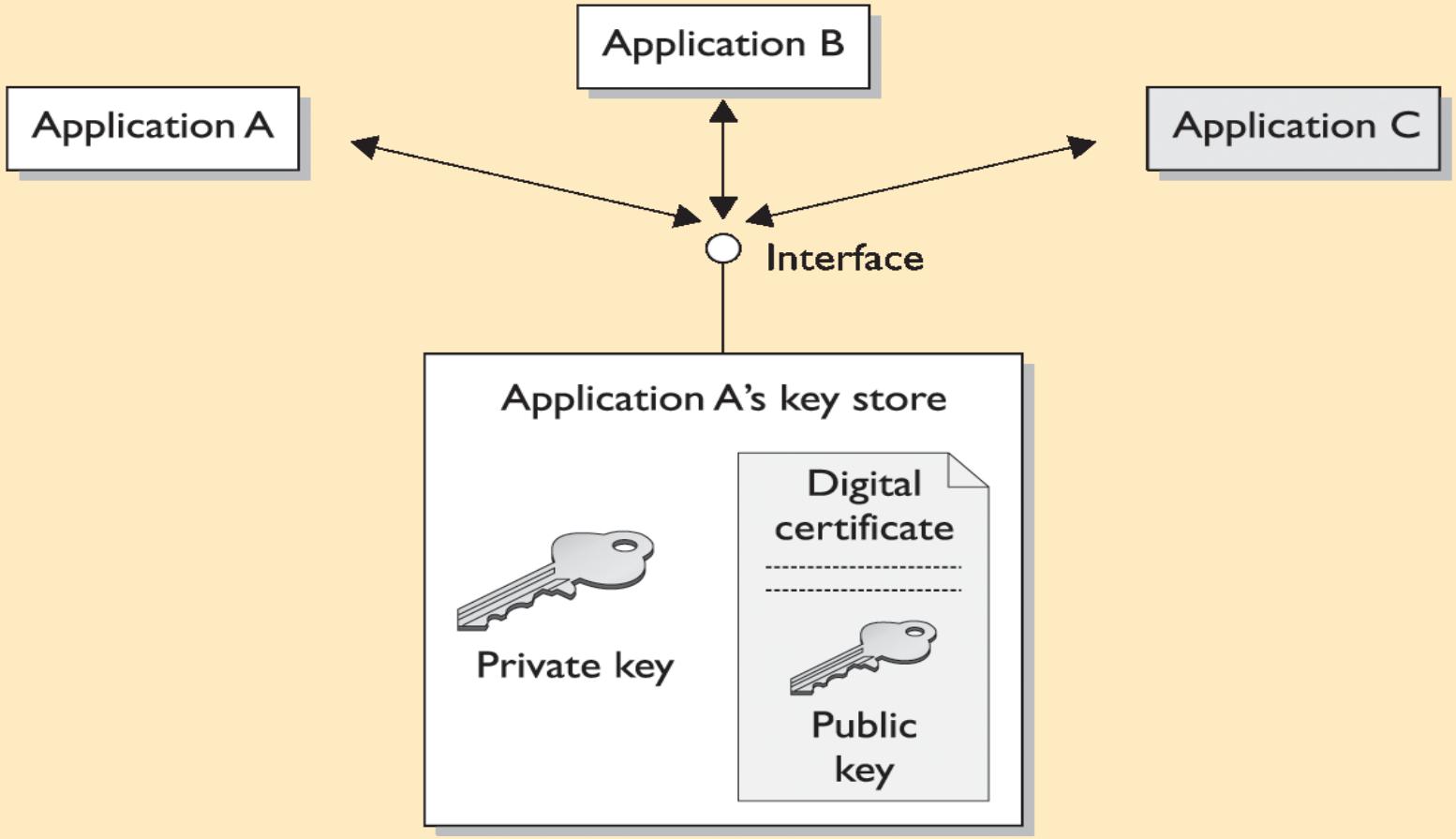
# *Certificate Types / Classes*

- A **Class 1 certificate** is used to digitally sign e-mail and encrypt message contents.
- A **Class 2 certificate** is used for software signing.
- A **Class 3 certificate** may be used by a company to set up its own certificate authority.

# *Hierarchy*

- Each **higher class** of certificate can perform more powerful and critical tasks than the ones before it.
  - Each CA outlines the certification classes it provides and the identification requirements that must be met to acquire each type of certificate.

# Storing a Certificate

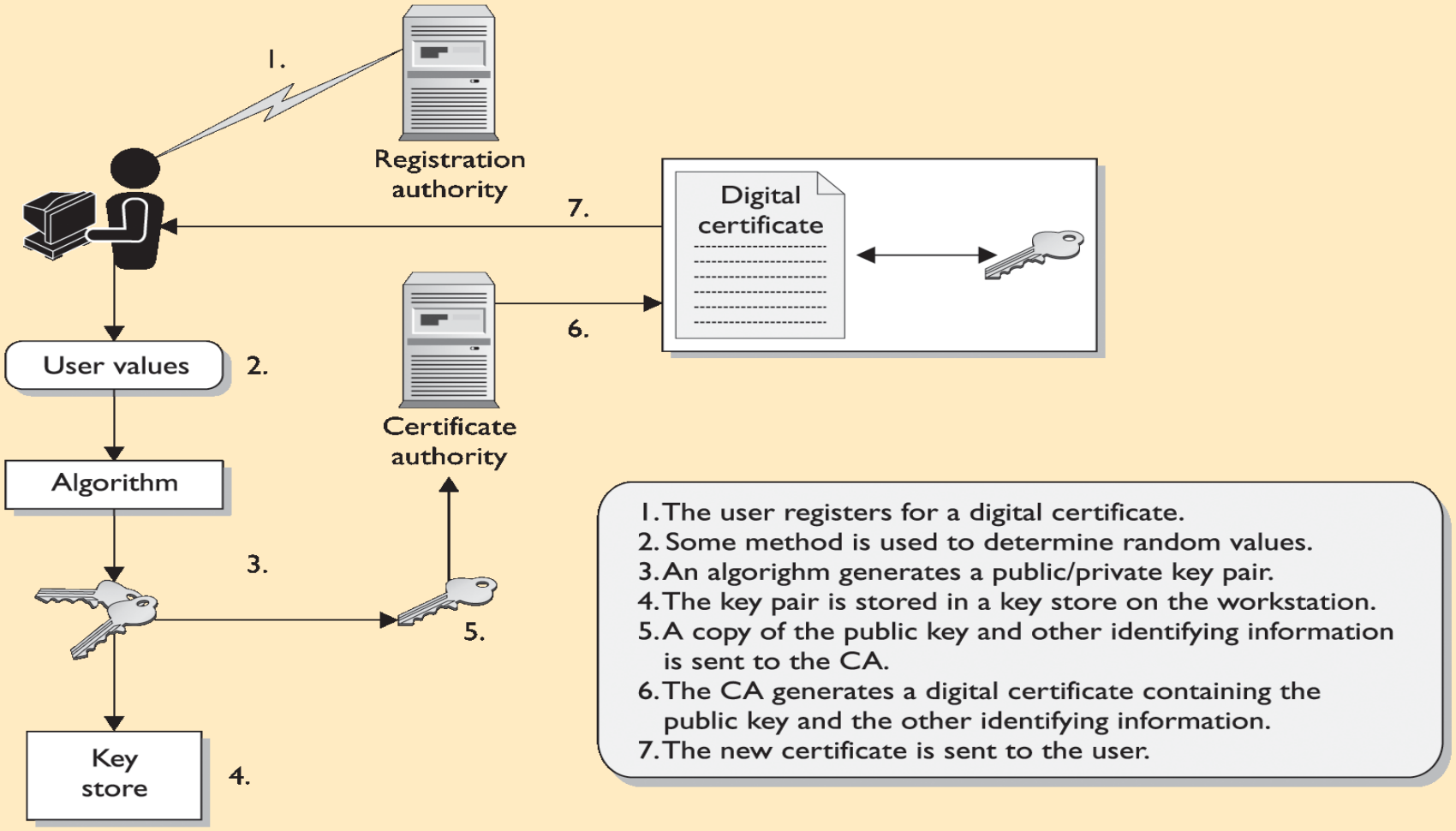



Some key stores can be shared by different applications.

# *Storing a Certificate*

- If an application creates a **key store** to be accessed by other applications, it will provide an **interface**, referred to as an application-programming interface (API).
- In **UNIX** systems, this interface is usually PKCS #11.
- In **Microsoft** applications, the interface is called Crypto API (CAPI).

# Obtaining a Digital Certificate




Steps for obtaining a digital certificate

# *Certificate Repositories*

- Once the certificate is registered, identity is proven, and a key pair generated, the certificate must be stored somewhere.

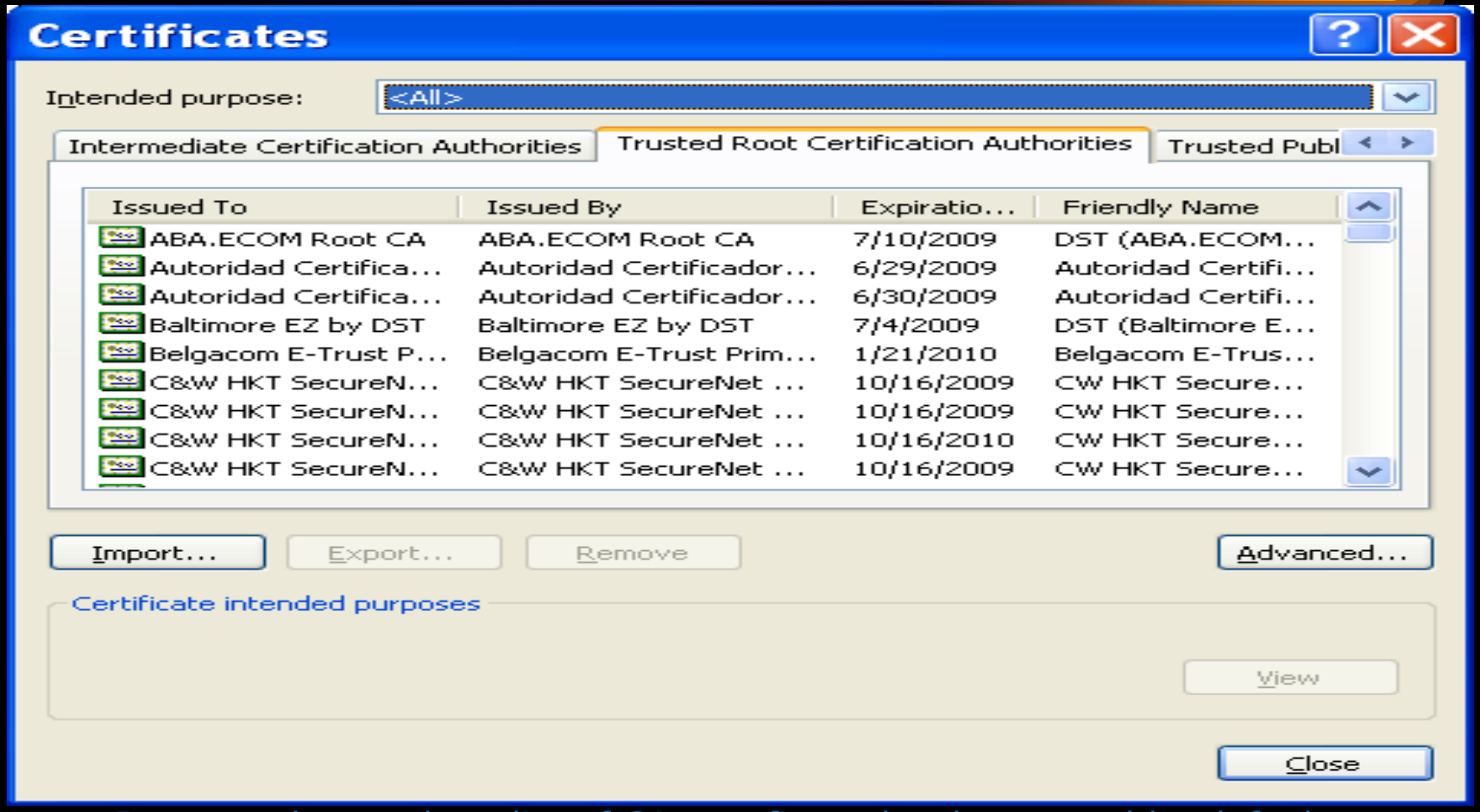
- Public keys, and their corresponding **certificates, are held in a publicly available repository**.
  - They must be available to whoever requires them to communicate within a PKI environment.

- They can be accessed and searched using the **Lightweight Directory Access Protocol (LDAP)**.

# *Trust and Certificate Verification*

- **When users trust a CA**, they will download that CA's digital certificate and public key, which will be stored on their local computer.
  - Most **browsers** have a list of CAs configured to be trusted by default.
  - When a user installs a new Web browser, several of the well-known and most trusted CAs will be trusted without any change of settings.

# Trust and Certificate Verification



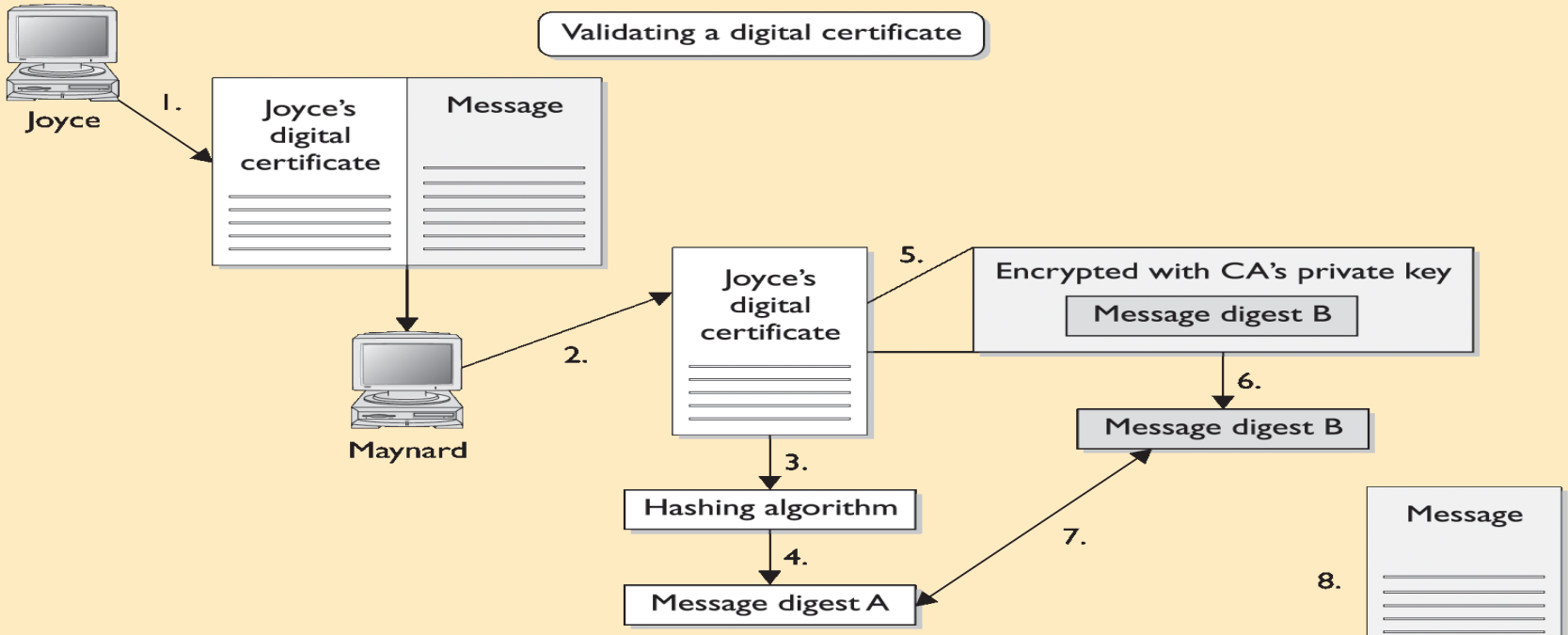

Browsers have a long list of CAs configured to be trusted by default.

# *Distinguished Names*

- A **distinguished name** is a label that follows the **X.500 standard**.
  - This standard defines a naming convention that can be employed so that each subject within an organization has a unique name.
  - An example is **{Country = US, Organization = Real Secure, Organizational Unit = R&D, Location = Washington}**.

- CAs use distinguished names to identify the owners of specific certificates.

# Verifying Authenticity and Integrity of a Cert




1. Joyce sends the message and digital certificate to Maynard.
2. Maynard extracts the certificate.
3. He puts the certificate through a hashing algorithm.
4. The algorithm calculates a value of A.
5. Maynard extracts the encrypted message digest from the certificate.
6. Maynard decrypts the value with the CA's public key.
7. Maynard checks to see if the certificate was revoked.
   Maynard compares values A and B.
8. The values are the same, so Maynard reads the message.

Steps for verifying the authenticity and integrity of a certificate

# *Verifying Authenticity and Integrity of a Cert*

- To **verify the authenticity and integrity of a certificate**:
  - **Compare the CA** that digitally signed the certificate to a list of CAs that has already been loaded into the receiver's computer.
  - **Calculate a message digest** for the certificate.
  - **Use the CA's public key to decrypt the digital signature** and recover what is claimed to be the original message digest embedded within the certificate (validating the digital signature).
  - **Compare the two resulting message digest values** to ensure the integrity of the certificate.
  - **Review the identification information** within the certificate, such as the e-mail address.
  - **Review the validity dates**.
  - **Check a revocation list** to see if the certificate has been revoked.

# *Trust and Certificates in Production*

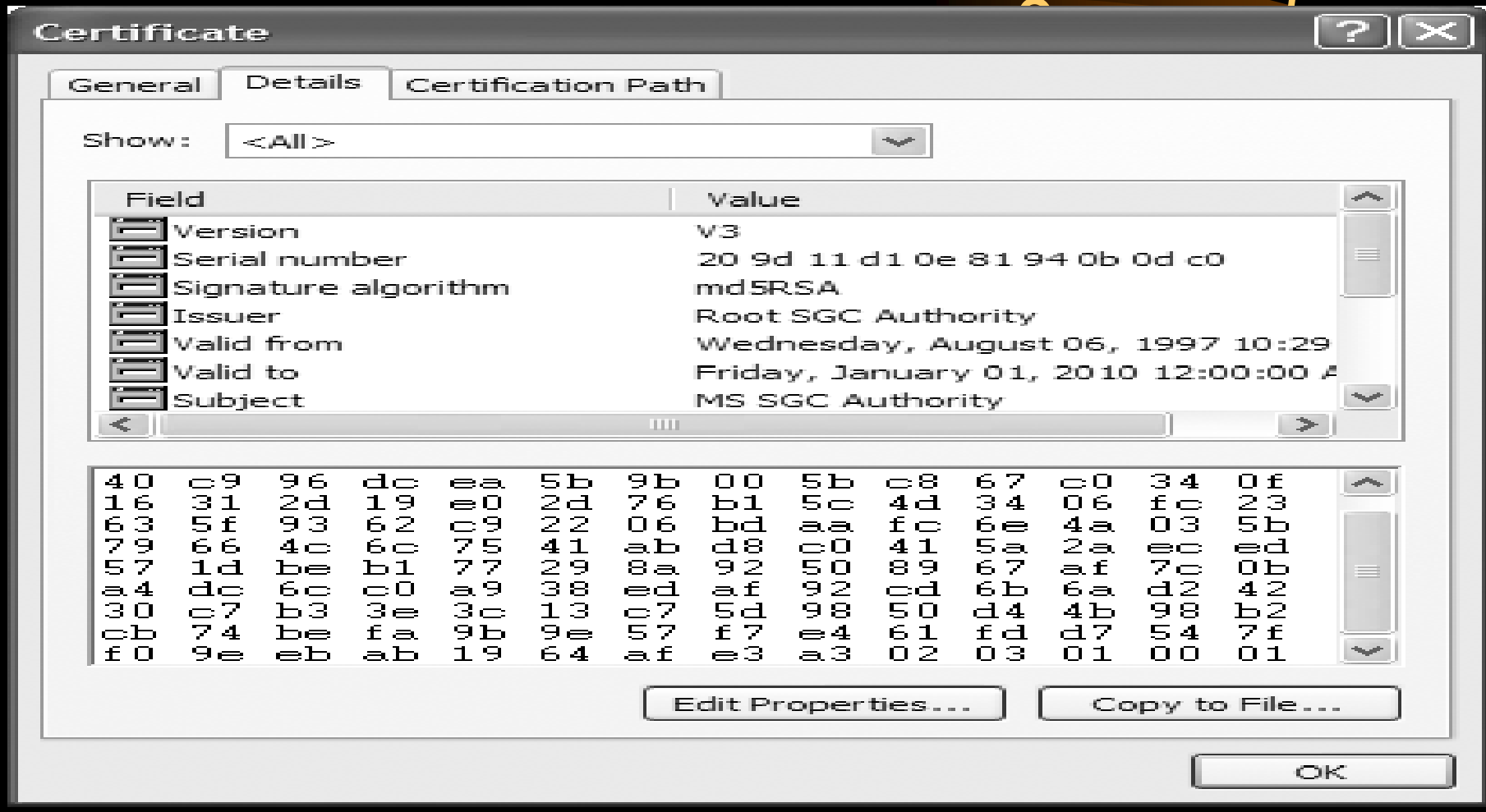
- In **production environments** that require a higher degree of protection, this list [revocation list] will be pruned, and **possibly the only CAs listed will be the company's internal CAs**.
  - This ensures that digitally signed software will only be automatically installed if the company's CA signed it.

- Other products, like Entrust, **use centrally controlled policies to determine which CAs should be trusted** instead of expecting the user to make these critical decisions.

# *Digital Certificates*

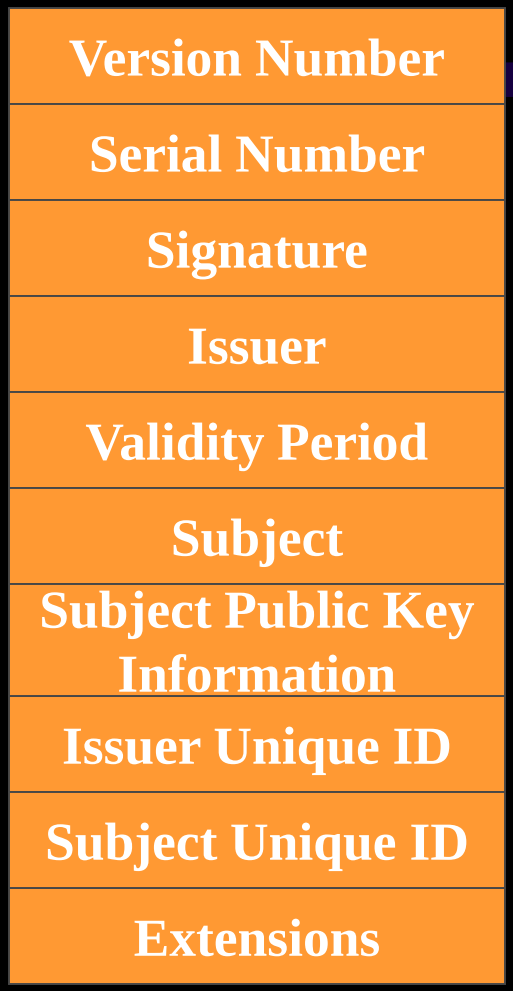
- A **digital certificate** binds an individual's identity to a public key, and it contains all the information a receiver needs to be assured of the identity of the public key owner.

- After an RA verifies an individual's identity, the CA generates the digital certificate.
  - The certificates are created and formatted based on the **X.509** standard, which outlines the necessary fields of a certificate and the possible values that can be inserted into the fields.



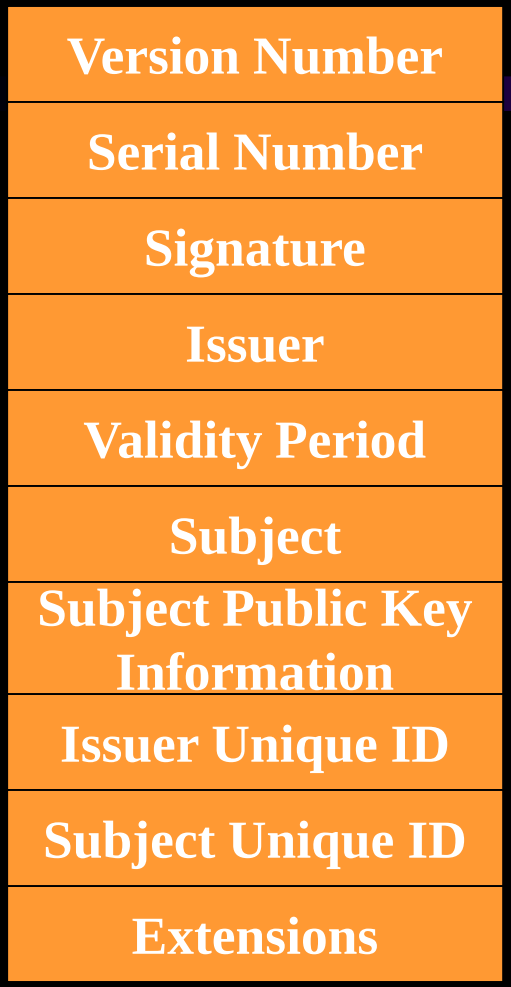

Browsers have a long list of CAs configured to be trusted by default.

# Standard Fields



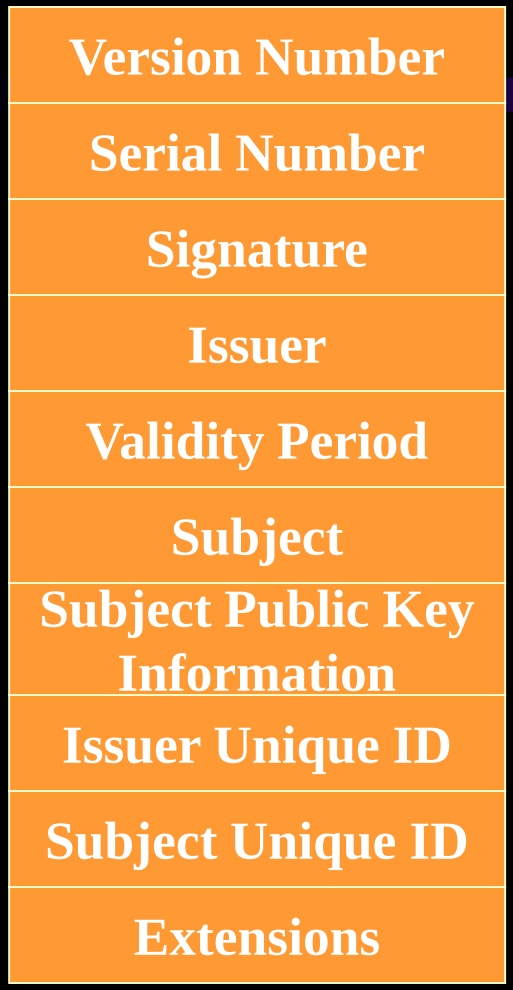

- Version Number
  - Identifies the version of the X.509 standard that was followed to create the certificate.
  - The version number indicates the format and fields that can be used.
- Serial number
  - Contains a unique number identifying a specific certificate issued by a particular CA.
- Signature algorithm
  - Identifies the hashing algorithm and the digital signature algorithm used to digitally sign the certificate.

# Standard Fields




- Issuer
  - Identifies the CA that generated and digitally signed the certificate.
- Validity
  - Specifies the dates through which the certificate is valid for use.
- Subject
  - Specifies the owner of the certificate and can be a network device (router, Web server, firewall, and so on), an application, a department, a company, or a person.

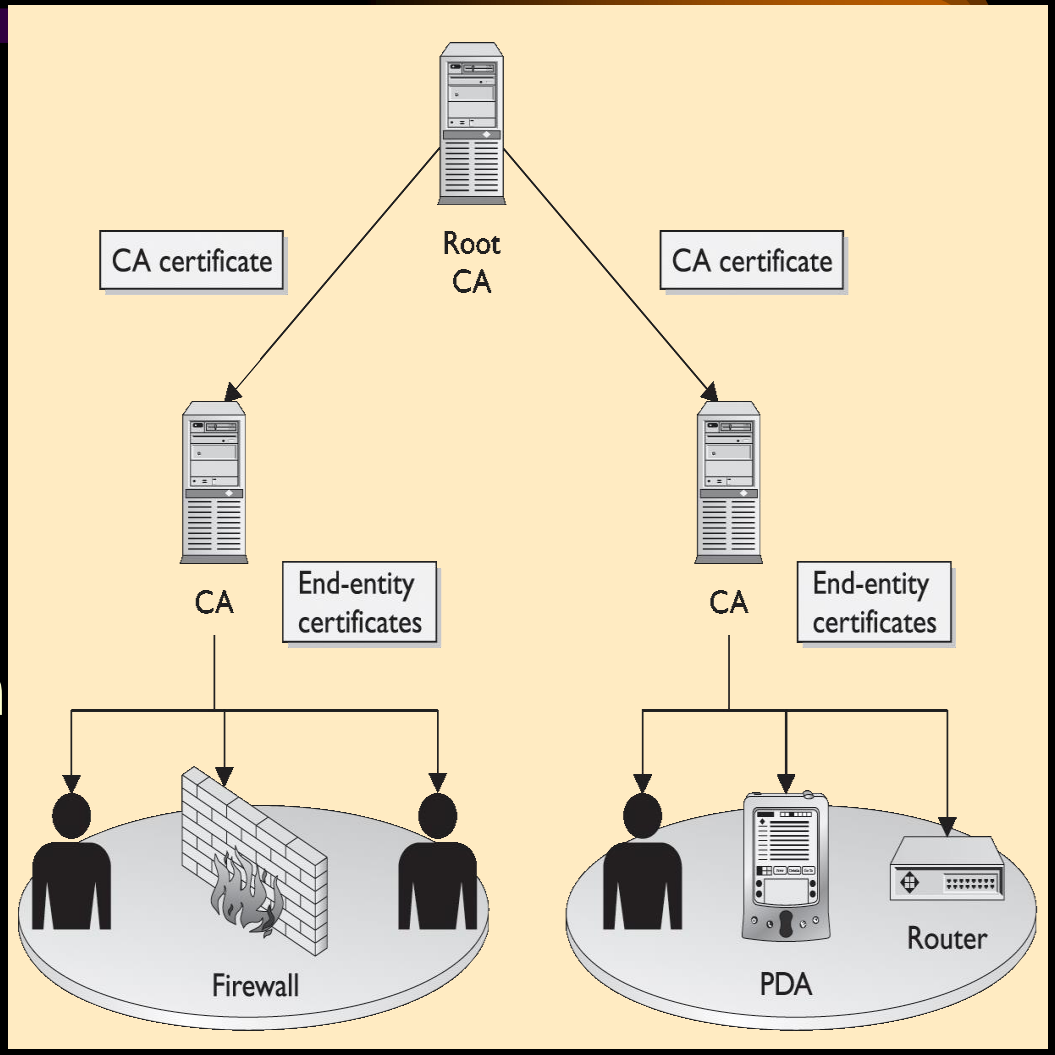
# Standard Fields




- Public key
  - Contains the public key being bound to the certified subject, which also identifies the algorithm that was used to create the private/public key pair.
- Certificate usage
  - Specifies the approved use of a certificate, which dictates the use of this public key.
- Extensions
  - Allow additional data to be encoded into the certificate to expand the functionality of the certificate.

# Certificate Attributes

- The four main types of certificates are:
  - End-entity
  - CA
  - Cross-certification
  - Policy




End-entity and CA certificates

# *Types of Certificates*

- **End-entity certificates**
  - End-entity certificates are issued by a CA to a **specific subject** such as Joyce, the accounting department, or a firewall.

- **CA certificates**
  - A CA certificate may be **self-signed in case of a stand-alone or root CA**, **or** it may be **issued by a superior CA** within a hierarchical model.
  - The superior CA gives the authority and allows the subordinate CA to accept certificate requests and generate the individual certificates.

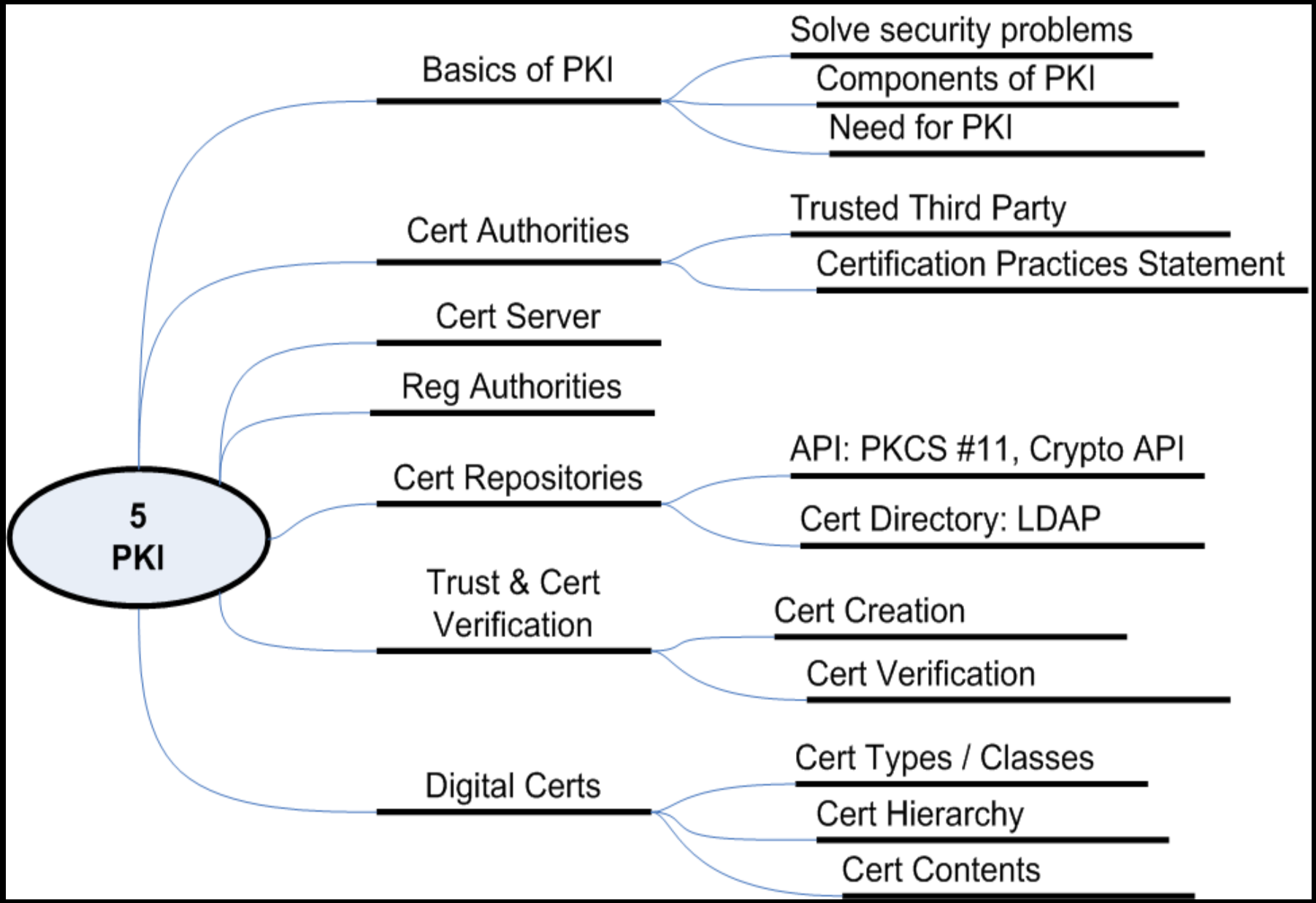
# *Types of Certificates*

- **Cross-certification certificates**
  - Cross-certificates, or cross-certification certificates, are used when **independent CAs establish peer-to-peer trust relationships**.
  - They are a mechanism through which one CA can issue a certificate allowing its users to trust another CA.

- **Policy certificates**
  - Within sophisticated CAs used for high-security applications, a mechanism is required to provide centrally controlled policy information to PKI clients.
  - This is often done by placing the policy information in a policy certificate.

# Summary

- The Basics of PKI
- Certificate Authorities
- Registration Authorities
- Certificate Repositories
- Trust and Certificate Verification
- Digital Certificates


5
PKI
Basics of PKI
Solve security problems
Components of PKI
Need for PKI
Cert Authorities
Trusted Third Party
Certification Practices Statement
Cert Server
Reg Authorities
Cert Repositories
API: PKCS #11, Crypto API
Cert Directory: LDAP
Trust & Cert Verification
Cert Creation
Cert Verification
Digital Certs
Cert Types / Classes
Cert Hierarchy
Cert Contents

# *Reference*

- Conklin (Principles of Computer Security)
  - Chapter 6