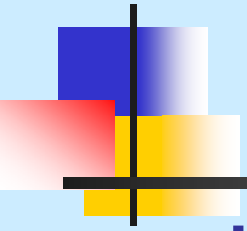
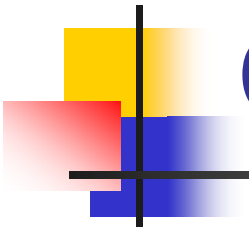
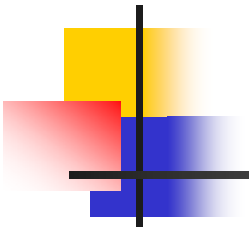


CHAPTER 5:
KEY MANAGEMENT & PKI



OBJECTIVES

- Key management issues in Private Key and Public key encryption systems.
- Public Key Infrastructure basics

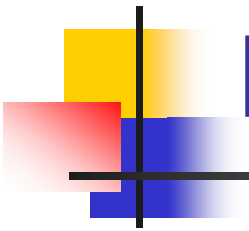


Encryption

- The problems with establishing and managing a secure messaging system are to ensure that :
  - Encryption techniques and secret keys are sufficiently complex so that unauthorized people cannot decrypt messages
  - Keys are accessible to people who are authorized to use them, and kept away from people who are not authorized to use them

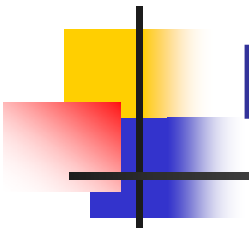
Private Key Cryptography

- The problem with symmetrical key encryption is key distribution: ensuring that the keys to the message senders and recipients do not get into the hands of unauthorized persons.
- As the number of users of the secure messaging system increases, the problem of generating, distributing, safeguarding, and accounting for the secret keys increases at a geometric rate. (order of n^2)



Public Key Cryptography

- Public key cryptography uses two keys that are mathematically linked; one key can be used only to encrypt a message, and the other key can be used only to decrypt the message.
- The key (public key) that is used to encrypt a message can be freely distributed (or placed in an accessible directory), and the recipient keeps the key (private Key) used to decrypt the message.

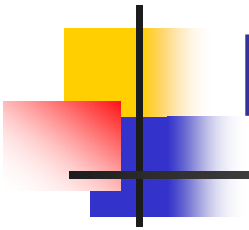


Public Key Cryptography (cont.)

- Can greatly improve cryptography security, convenience, and flexibility
- Public keys can be distributed freely
- Users cannot deny they have sent a message if they have previously encrypted the message with their private keys
- Primary disadvantage is that it is computing-intensive

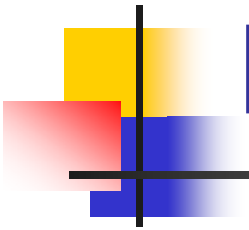
Public / Private Key Pair

- Public Key Cryptography provides the basis for:
  - Digital Envelopes : anyone can encrypt data with the public key; only the holder of the private key can decrypt.
  - Digital Signatures : the holder of the private key can encrypt (sign); anyone can verify that the owner of the private key did the encryption (signature)
- The Private key must be kept secret by its owner
- The Public Key is freely distributed for others to use.



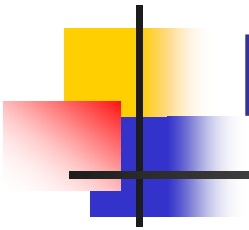
Public Key ?

- How to find out someone's Public Key?
  - Send with message
  - Lookup from database
- How to trust the result?
  - Digital Certificates
  - Trusted Certification Authority
- Signed message that proves "Bob's key is N"



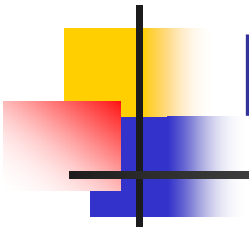
Encryption Keys

- It doesn’t matter how strong your encryption algorithm is
- Or how secure your protocol is
- If the opponents can get hold of your keys, your security is gone
- Proper use of keys is crucial to security in computing systems



Properties of Keys

- Length
- Randomness
- Lifetime



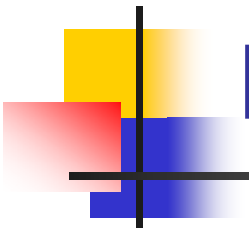
Key Length

- If your cryptographic algorithm is otherwise perfect, its strength depends on key length
- Since the only attack is a brute force attempt to discover the key
- The longer the key, the more brute force required
- Good recommendations are to use at least 80 bit keys for private key encryption and at least 1024-bit keys for RSA and Diffie-Hellman, 160-bit ECC keys are also thought to be secure

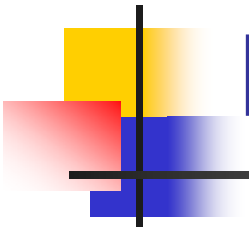
Relative Strengths of Different Key Lengths

Private key encryption(DES,RC5)	Public key Encryption (RSA, Diffie-Hellman)	Elliptic Curve
40 bits		
56 bits	400 bits	
64 bits	512 bits	
80 bits	768 bits	
90 bits	1024 bits	160 bits
120 bits	2048 bits	210 bits
128 bits	2304 bits	256 bits

Are There Real Costs for Key Length?

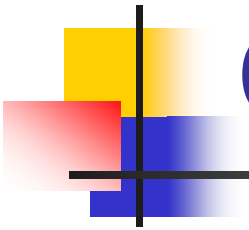


- Clearly, more bits is more secure
- Why not a whole lot of key bits, then?
- Much encryption done in hardware
  - More bits in hardware costs more
- Software encryption slows down as you add more bits, too
  - Public key cryptography costs are highly dependent on key length



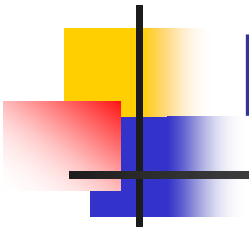
Key Randomness

- Brute force attacks assume your key at random
- If the attacker can get any knowledge about your mechanism of choosing a key, he can substantially reduce brute force costs
- The closer the method chosen approaches true randomness, the better



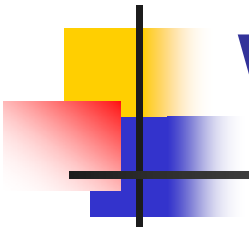
Cryptographic Methods

- Start with a random number
- Use a cryptographic hash on it
- If the cryptographic hash is a good one, the new number looks pretty random
- Produce new keys by hashing old ones
- Depends on strength of hash algorithm



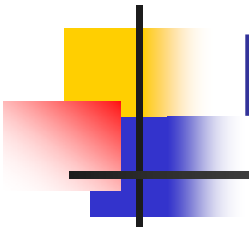
Key Lifetime

- If a good key's so hard to find,
  - Why ever change it/
- How long should one keep using a given key?



Why Change Keys

- Long-lived keys are more likely to be compromised
- The longer a key lives, the more data is exposed if compromised
- The longer a key lives, the more resources opponents can (and will) devote to breaking it
- The more a key is used, the easier the cryptanalysis on it
- A secret that cannot be readily changed should be regarded as a vulnerability

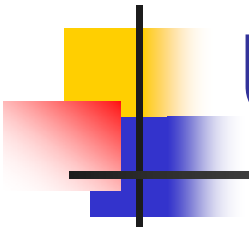


Practicalities of Key lifetimes

- In some cases, changing keys is inconvenient
  - E.g. encryption of data files
- Keys used for specific communications sessions should be changed often
  - E.g. new key for each phone call
- Keys used for key distribution can’t be changed too often

Desirable Properties in a Key Management System

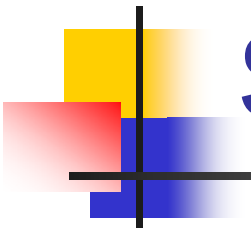
- Secure
- Fast
- Low overhead for users
- Scaleable
- Adaptable
  - Encryption algorithms
  - Applications
  - Key lengths



Users and Keys

- Where are user's keys kept?
- Permanently on the user's machine?
  - What happens if the machine is cracked?
- But people can't remember random(ish) keys
  - Hash keys from passwords/passphrases?
- Keep keys on smart cards?
- Get them from key servers?

Security measures for Key Servers

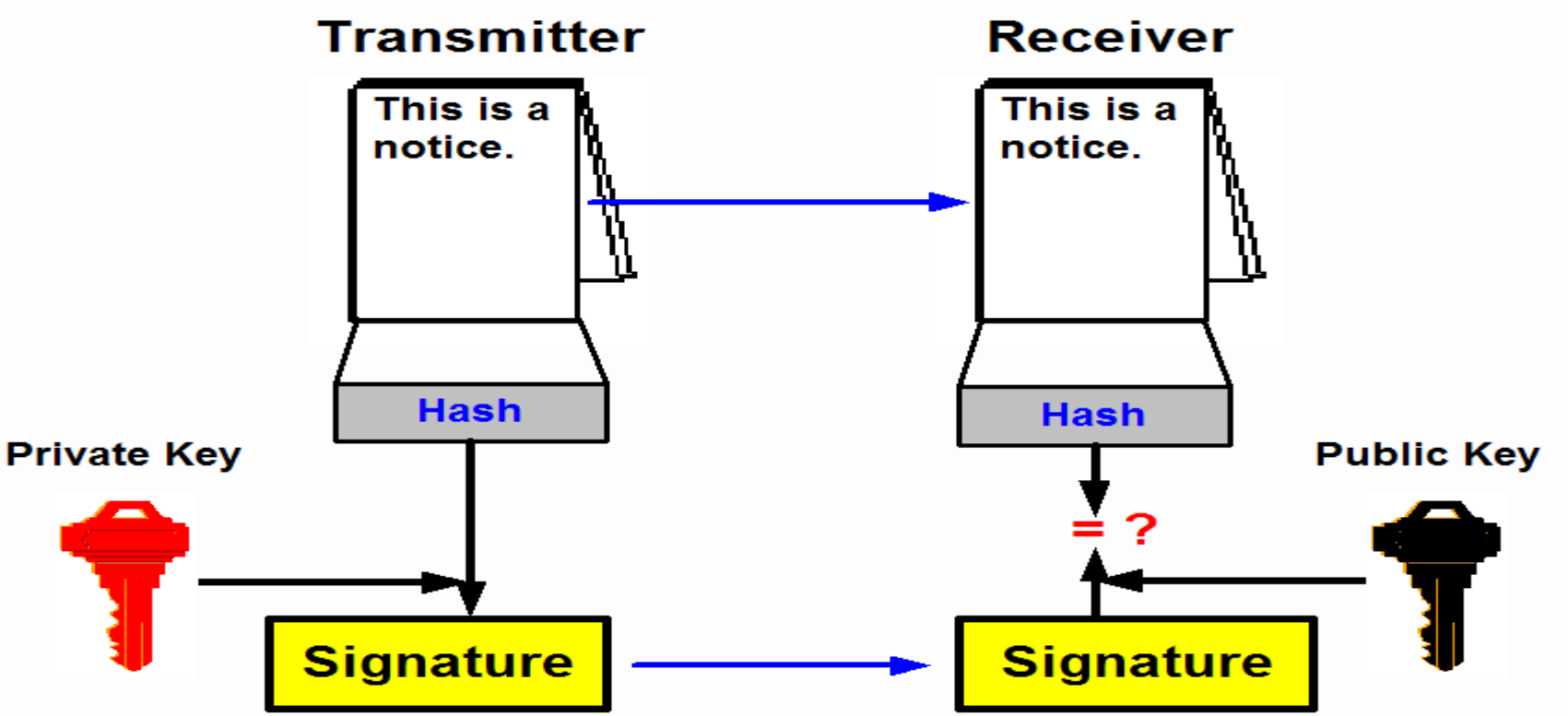


- If a cracker breaks the key server, everything goes with it
- The key server should be dedicated to just that service
- Extraordinary care should be used in setting it up and administering it
- Use a key server that stores as few keys permanently as possible
- Use a key server that handles revocation and security problems well

Digital Signatures

- Electronic signatures are data attached to other data for authentication purposes
- Digital signatures are electronic signatures linked to the signed data in a way that tampering is noticed and that the sender can be identified unequivocally.
- A digital signature helps to prove that:
  - The person sending the message with a public key is who they claim to be
  - The message was not altered
  - It cannot be denied the message was sent

Digital Signatures (cont.)



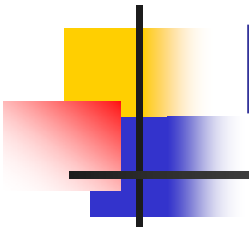
Digital Signatures (cont.)

- To create a digital signature, the signing transmitter creates a Manipulation Detection Code (hash) of the message and then uses an exclusively transmitter-owned private key to encrypt the hash. This is the digital signature and it is attached to the real message (message expanding).

- The private key has a matching public key that the receiver can use to verify the signature. The receiver uses the same hash function to create a hash of the real message, and then takes the public key to the transmitter, decrypts the digital signature, and compares hashes.

Digital Signatures (cont.)

- A trustworthy institution (i.e., a Trust Center or a Certificate Authority) assigns this pair of keys to a particular person. The following factors form the basis for using digital signatures:
  - Secure software, which supports digital signature functionality (e.g.. email-clients or plug ins)
  - Secure infrastructure, which supports key exchange (PKI—a Trust Center is a special PKI with more security)
  - Choice of hash functions and public key algorithms



Digital Certificates

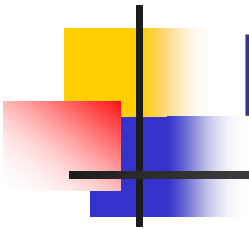
- Digital documents that associate an individual with its specific public key
- Data structure containing a public key, details about the key owner, and other optional information that is all digitally signed by a trusted third party

Digital Certificates (cont.)

- Digital certificates are virtual fingerprints that authenticate absolutely the identity of a person or thing.
- The certificate itself is simply a collection of information to which a digital signature is attached.
- A third-party authority that the community of certificate users trusts attaches the digital signature.

Public Key Infrastructure PKI

- The need for universal systems to support e-commerce, secure transactions, and information privacy is one aspect of the issues being addressed with PKI
- PKI is two-key asymmetric system with four key components : CA, RA, RSA, and digital certificates.
  - CA : Certification Authority
  - RA : Registration Authority
- Messages are encrypted with a public key and decrypted with a private key



Example

- You want to send an encrypted message to Gigi,so you request her public key
- Gigi responds by sending you that key
- You use the public key she sent you to encrypt the message
- You send the message to Gigi
- Gigi uses her private key to decrypt the message

The Main Goal of PKI

- To define an infrastructure that should work across multiple vendors, systems, and networks.
- It is important to note that *PKI is a framework* and not a specific technology
- Implementations of PKI are dependent on the perspective of the software vendors that implement it
- This has been one of the major difficulties with PKI…incompatibilities across vendors.

What is the Purpose of PKI ?

- PKI arrangements enable computer users to be authenticated to each other, and to use the information in identity certificates (i.e., each other's public keys) to encrypt and decrypt messages traveling to and fro.
- In general, a PKI consists of client software, server software such as a certificate authority, hardware (e.g., smart cards) and operational procedures.
- A user may digitally sign messages using his private key, and another user can check that signature (using the public key contained in that user's certificate issued by a certificate authority within the PKI).
- This enables two (or more) communicating parties to establish confidentiality, message integrity and user authentication without having to exchange any secret information in advance.

Typical use of PKI

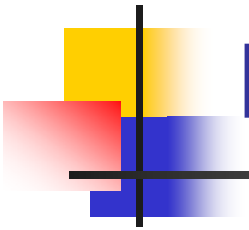
- Most enterprise-scale PKI systems rely on certificate chains to establish a party's identity, as a certificate may have been issued by a certificate authority computer whose 'legitimacy' is established for such purposes by a certificate issued by a higher-level certificate authority, and so on.
- This produces a certificate hierarchy composed of, at a minimum, several computers, often more than one organization, and often assorted interoperating software packages from several sources.
- Standards are critical to PKI operation, and public standards are critical to PKIs intended for extensive operation.

Typical use of PKI (cont.)

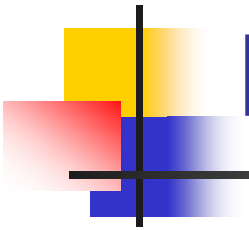
- Enterprise PKI systems are often closely tied to an enterprise's directory scheme, in which each employee's public key is often stored (embedded in a certificate), together with other personal details (phone number, email address, location, department, ...).

- Today's leading directory technology is LDAP and in fact, the most common certificate format (X.509) stems from its use in LDAP's predecessor, the X.500 directory schema.

Lightweight Directory Access Protocols LDAP



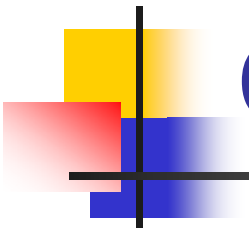
- Is a standard directory access protocol that allows queries to be made of directories (specifically. Pared-down X.500-based directories)
- If a directory supports LDAP, you can query that directory with an LDAP client
- LDAP is the main access protocol used by Active Directory AD (by Microsoft)



Public Key Certificates

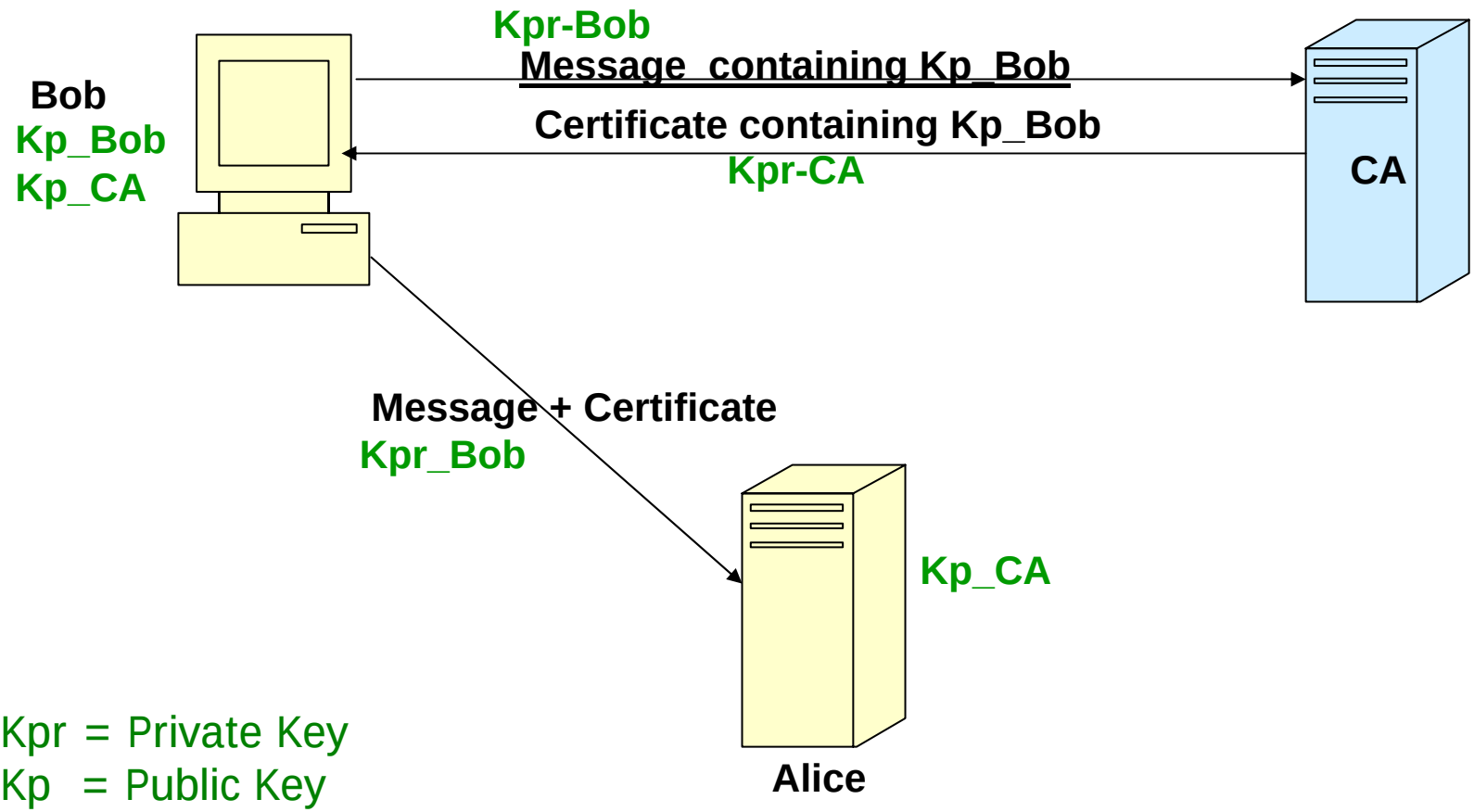
- The most common kind of certificate
- Essentially, a copy of your public key signed by a trusted authority
- Presentation of the certificate alone serves as authentication of your public key

Implementation of Public Key Certificates

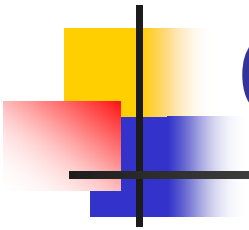


- Set up a universally trusted authority
- Every user presents his public key to the authority
- The authority returns a certificate
  - Containing the user's public key signed by the authority's private key.

PKI Operation



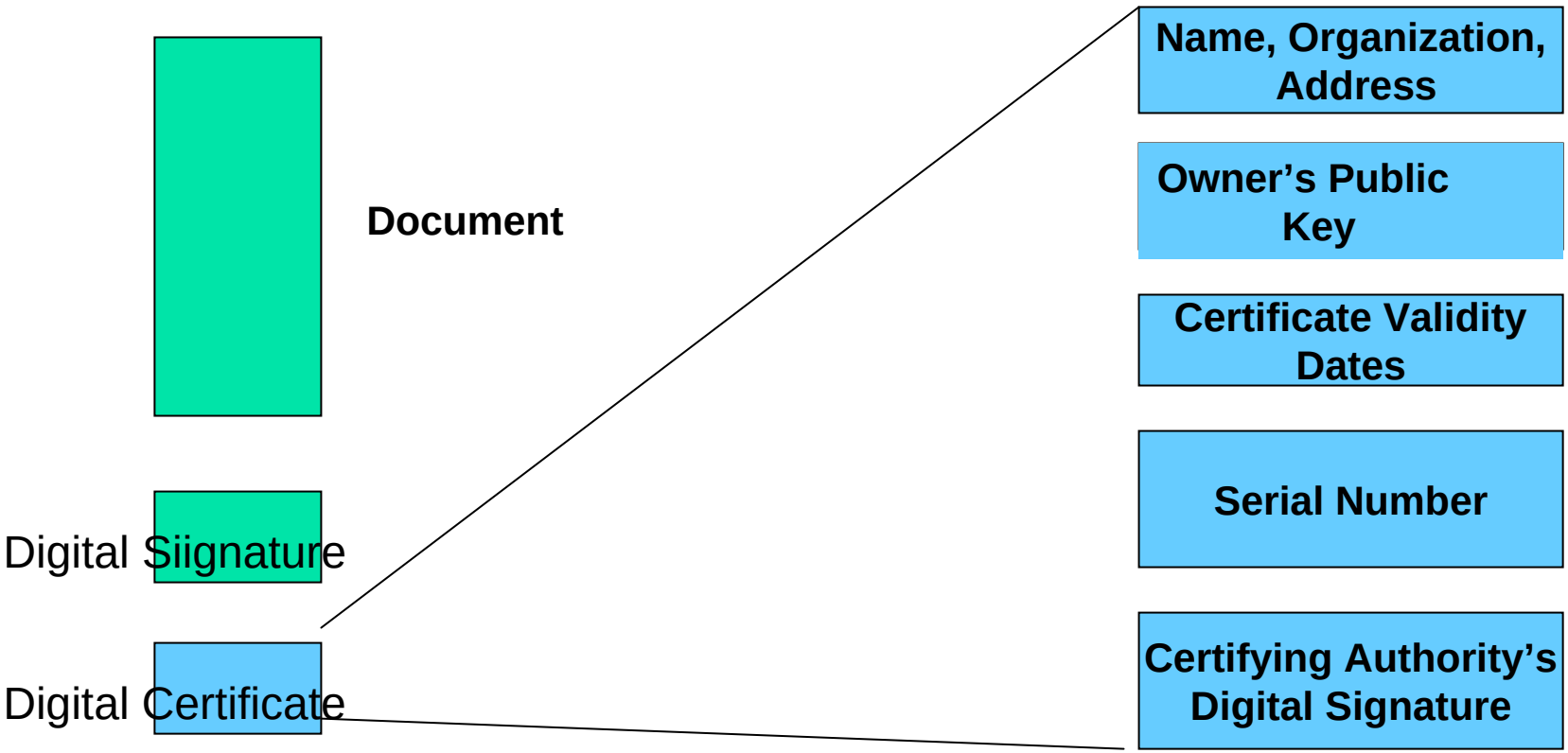
Kpr = Private Key
Kp = Public Key



Checking a Certificate

- Every user keeps a copy of the authority's public key
- When a new user wants to talk to you, he gives you his certificate
- Decrypt the certificate using the authority's public key
- You now have an authenticated public key for the new user
- Authority need not be checked on-line

Digital Certificate



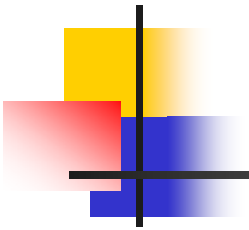
Digital Certificate (ITU Standard X.509)

- X.509 is an international standard defined by the International Telecommunication Union (ITU) that defines the format for the digital certificate
- Most widely used certificate format for PKI
- X.509 is used by Secure Socket Layers (SSL)/Transport Layer Security (TLS), IP Security (IPSec), and Secure/Multipurpose Internet Mail Extensions (S/MIME)

Digital Certificate Fields (ITU Standard X.509) (cont.)

Table □ X.509 certificate

Field Name	Explanation
Certificate version number	0=Version 1, 1=Version 2, 2=Version 3
Serial number	Unique serial number of certificate
Issuer signature algorithm ID	"Issuer" is Certificate Authority
Issuer X.500 name	Certificate Authority name
Validity period	Start date/time and expiration date/time
Subject X.500 name	Private key owner
Subject public key information	Algorithm ID and public key value
Issuer unique ID	Optional; added with Version 2
Subject unique ID	Optional; added with Version 2
Extensions	Optional; added with Version 3
Signature	Issue's digital signature



Certificate Policies

- A CERTIFICATE POLICY defines what the certificate can be used for.
- A CA can potentially issue a number of different certificates : say, one for e-mail, one for e-commerce, and one for financial transactions.
- The policy might indicate that it is not to be used for signing contracts or for purchasing equipment.
- Certificate policies affect how a certificate is issued and how it is used

Certificate Policies (cont.)

- CERTIFICATE POLICY is a set of rules that indicates the applicability of a certificate.
- A CA would have policies regarding the interoperability or certification of another CA site
- The process of requiring interoperability is called *cross-certification*
- The organizations using the certificates also have the right to decide which types of certificates are used and for what purposes
- The receiving organization can use this policy to determine whether the certificate has come from a legitimate source
- The policy indicates which certificates will be accepted in a given application

Policy Framework

- When Alice sends an encrypted message to Bob, she must be certain that the public key she uses to encrypt the message is actually Bob's public key and not the public key of someone who is impersonating Bob.
- A digital certificate is used to prove that a public key actually belongs to a particular person or thing (the "subject" of the certificate).
- A CA issues the digital certificate. The degree to which a certificate user (i.e., Alice) can trust the certificate issued to the subject (i.e., Bob) depends upon the "Certificate Policies" and "Certification Practices" the CA uses to verify the subject's identity when issuing the certificate.

Policy Framework (cont.)

- To meet the needs of users, CAs may provide a range of digital certificates that vary according to "a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements."
- These rules are found in a Certificate Policy ("CP") and may be used by a certificate user to determine if a certificate "is sufficiently trustworthy for a particular purpose."

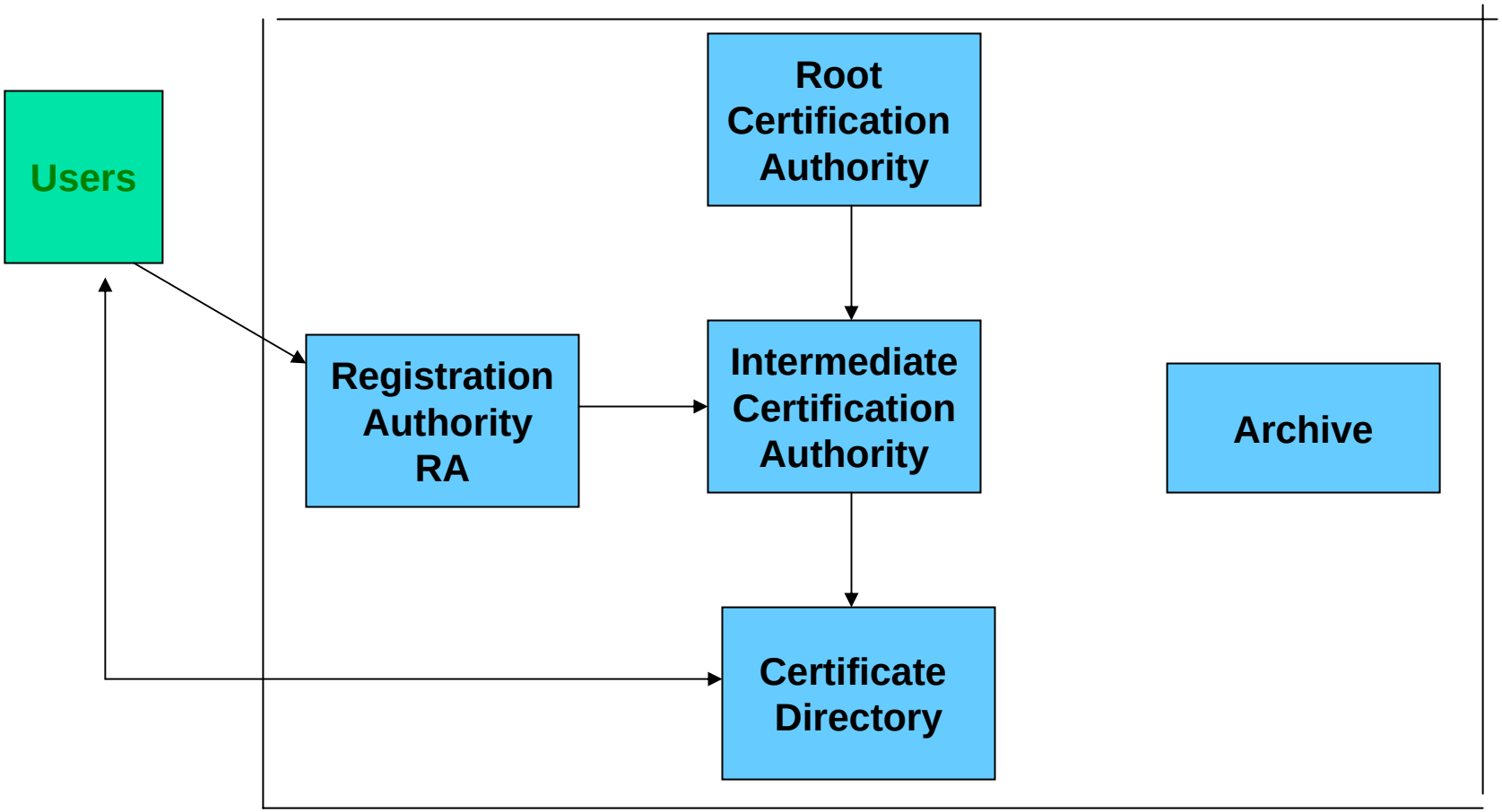
Policy Framework (cont.)

- A Certification Practice Statement ("CPS") is a detailed description of the practices followed by a CA in issuing and managing certificates.
- While a CP is typically eight to 10 pages and a CPS is typically 40 to 80 pages, an average end user does not care about the detailed cryptographic methods or security procedures that are used. The user's primary concern is the validity of the certificate and his or her liability (or exposure) for relying on the certificate.
- A PKI Disclosure Statement ("PDS") is a simplified document that is designed to assist users in making informed trust decisions.

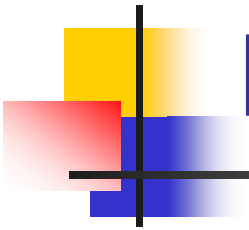
Key Management—PKI

- Weaknesses associated with asymmetric cryptography led to the development of PKI
- A CA is an important trusted party who can sign and issue certificates for users
- Some of its tasks can also be performed by a subordinate function, the Registration Authority RA
- Updated certificates and Certificate Revocation Lists CRLs are kept in a Certificate Repository CR for users to refer to
- The use of PKI enables a secure exchange of digital signatures, encrypted documents, authentication and authorization, and other functions in open networks where many communication partners are involved.

Public Key Infrastructure



Registration Authorities and LRAs



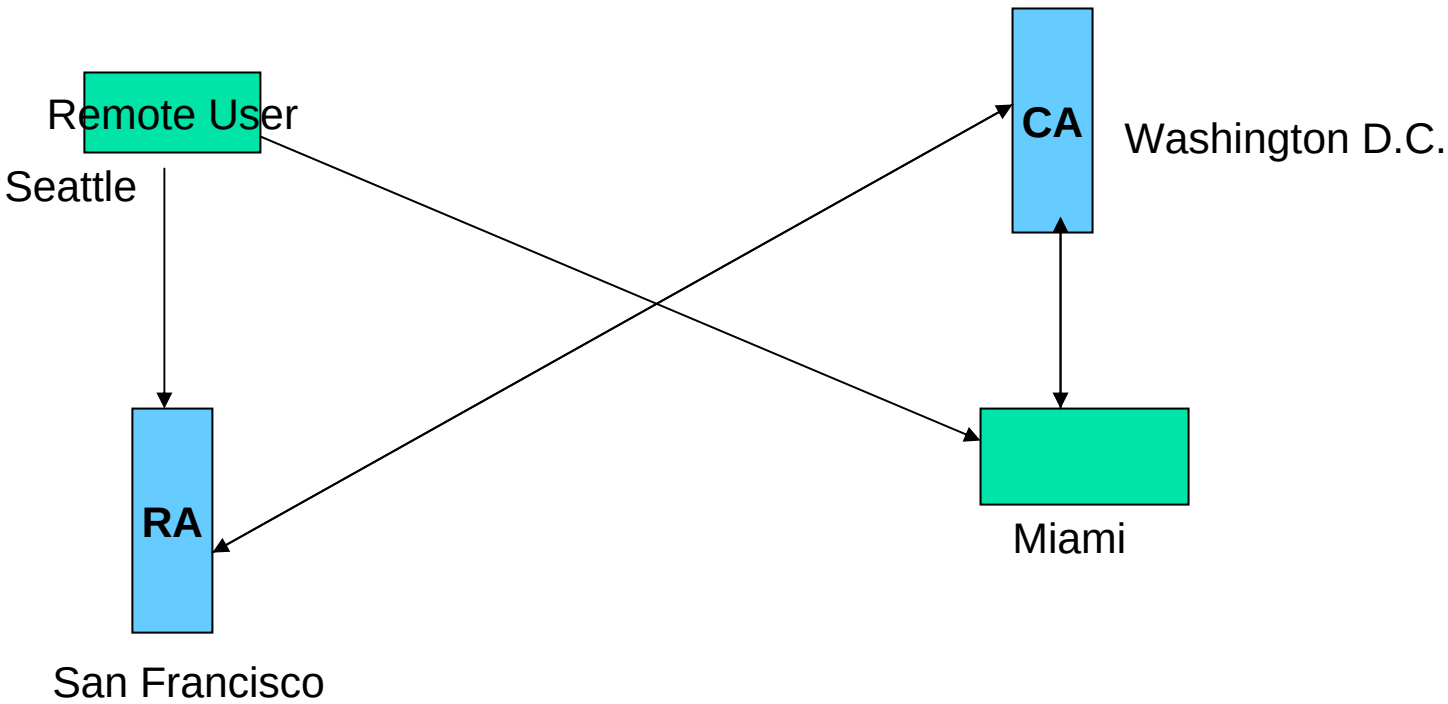
- A RA offloads some of the work from a CA
- A RA acts as a middleman in the process
- It can distribute keys, accept registration for the CA, and validate identities
- The RA does not issue certificates....That responsibility remains with the CA

Registration Authority (RA)

- The Registration Authority (RA) is responsible for recording and verifying all information the CA needs. In particular, the RA must check the user's identity to initiate issuing the certificate at the CA. This functionality is neither a network entity nor is it acting online. The RAs will be where users must go to apply for a certificate. Verification of the user identity will be done for example by checking the user's identity card.
- A RA has two main functions:
  - Verify the identity and the statements of the claimant
  - Issue and handle the certificate for the claimant

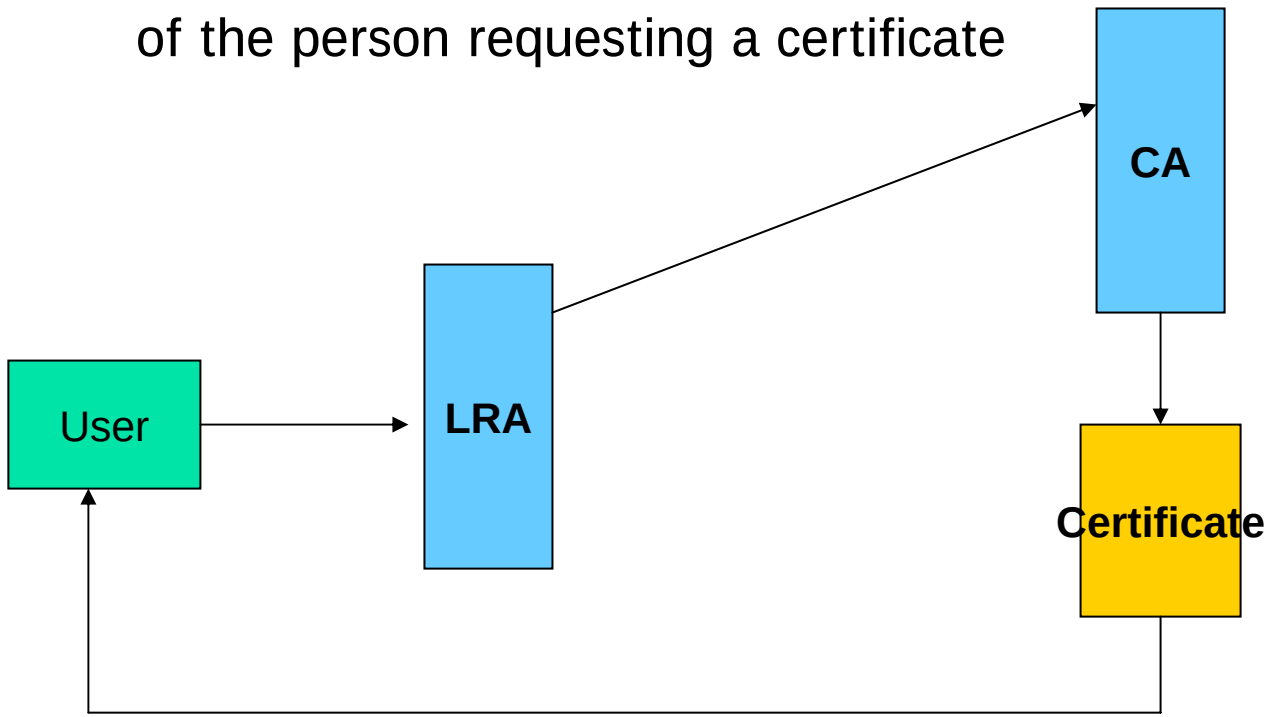
RA Relieving Work From a CA

- A user in Seattle wants to send a message to a user in Miami



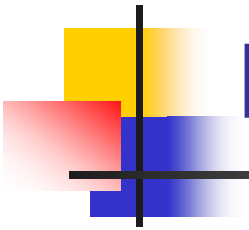
The LRA Verifying Identity for the CA

- User presents credentials to an LRA to get a certificate from a CA
- The LRA involves the physical identification of the person requesting a certificate



Description of PKI

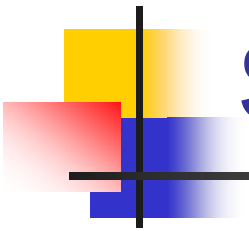
- Manages keys and identity information required for asymmetric cryptography, integrating digital certificates, public key cryptography, and CAs
- For a typical enterprise:
  - Provides end-user enrollment software
  - Integrates corporate certificate directories
  - Manages, renews, and revokes certificates
  - Provides related network services and security
- Typically consists of one or more CA servers and digital certificates that automate several tasks



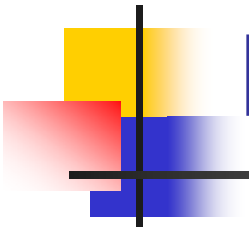
PKI Standards and Protocols

- A number of standards have been proposed for PKI
  - Public Key Cryptography Standards (PKCS)
  - X.509 certificate standards

Public Key Cryptography Standards (PKCS)



- Numbered set of standards that have been defined by the RSA Corporation since 1991
- Composed of 15 standards .

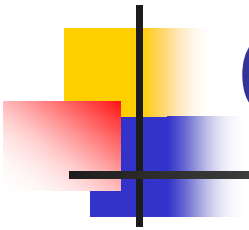


Directory Service

- The directory service has two main functions:
  - Publish certificates
  - Publish a Certificate Revocation List or to make an online certificate available via the Online Certificate Status Protocol (OCSP)

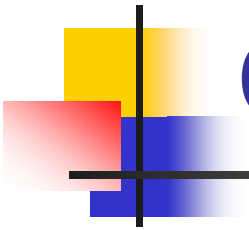
Certification Authority - Delegation

- CA can delegate its responsibilities to :
  - Registration authority (RA)
  - Repository (certificate directory)
  - Archive



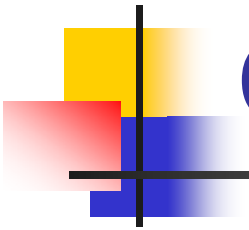
Certification Authority CA

- What is a CA ?
- CA functions
- Responsibilities
- Delegation



Certificate Authority

- A fundamental component of PKI.
- The Certificate Authority (CA) is the entity responsible for issuing and administering the digital certificates. The CA acts as the agent of trust in the PKI.
- Collection of Software, Hardware and people managing it.



Certificate Authority Functions

- A CA performs the following main functions:
  - Issues users with keys/Packet Switching Exchanges (PSEs) (though sometimes users may generate their own key pair)
  - Protect its private key
  - Verify subject information in CSR
  - Certifies users' public keys
  - Publishes users' certificates
  - Issues and publishes certificate revocation lists (CRLs)
  - Maintains archive of expired and revoked certificates

Certificate Authority Functions (cont.)

- Can provide the information in a publicly accessible directory, called a Certificate Repository (CR)
- Some organizations set up a Registration Authority (RA) to handle some CA, tasks such as processing certificate requests and authenticating users

Certificate Authority CA (cont.)

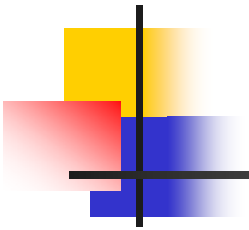
- The foundation upon which a PKI is built is trust—in other words the user community must trust the CA to distribute, revoke, and manage keys and certificates in such a way as to prevent any security breaches. As long as users trust the CA and its business processes, they can trust certificates the CA issues.
- The CA's signature in a certificate ensures that any changes to its contents will be detected. Such certificates can be distributed publicly and users retrieving a public key from a certificate can be assured of the validity that the key:
  - Belongs to the entity specified in the certificate
  - Can be used safely in the manner for which the CA certified it

Certificate Authority (cont.)

- A CA has the following tasks:
  - Generate the certificate based on a public key. Typically a Trust Center generates the pair of keys on a smart card or a USB token.
  - Guarantees the uniqueness of the pair of keys and links the certificate to a particular user
  - Manages published certificates
  - Is part of cross certification with other CAs

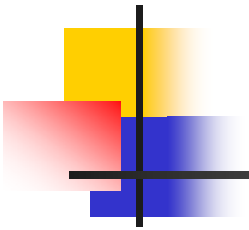
Issue Certificates – Root CA

- Root CA :
  - Issue certificates to other CA's
    - Level 1 CA
  - Authorizes Level 1 CA to issue certificates
  - Level 1 CA can issue other CA certificates if authorized
  - A CA chain is created in this manner



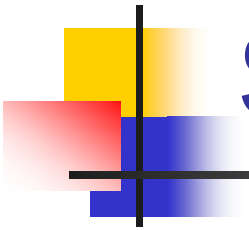
Issue Certificates

- Issue certificates to users
- Information in the certificate is binding to the entity
  - Name , Organization, Address
  - Public Key
  - Validity Dates
  - Serial Number
  - Certifying authority’s digital signature



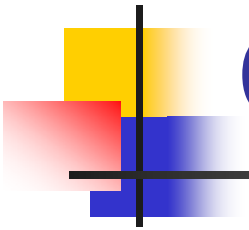
Issue Certificates - Process

- User submits a Certificate Signing Request (CSR) – with name, public key and other information
- CA follows known Policies and Procedures to validate request as defined in the Certificate Practice Statement (CPS)
- CA attaches extra information :
  - Validity dates, Key Usage, Account ID, etc.
- CA signs Certificate



Scaling Issues of Certificates

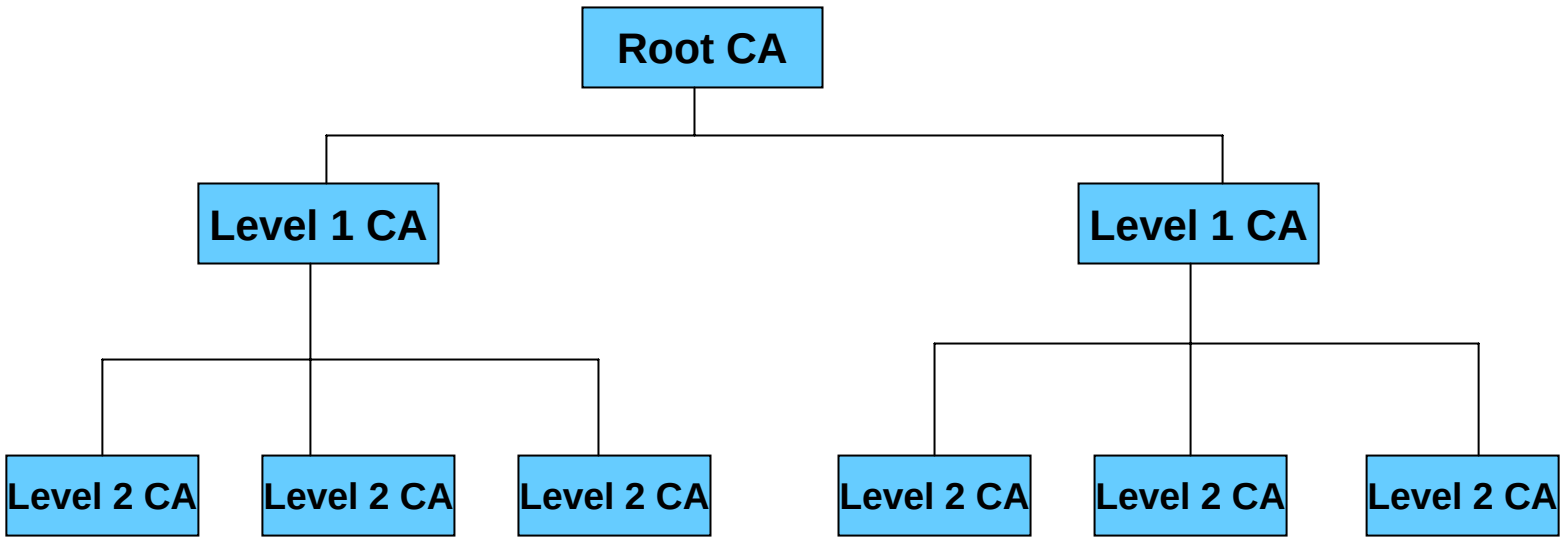
- If there are ~800 million Internet users needing certificates, can one authority serve them all?
- Probably not
- So we need multiple authorities
- Does that mean everyone needs to store the public keys of all authorities?

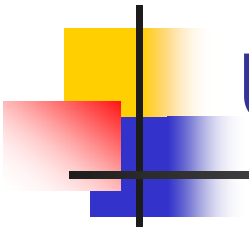


Certification Hierarchies

- Arrange certification authorities hierarchically
- The single authority at the top produces certificates for the next layer down
- And so on, recursively

PKI Certification Hierarchy

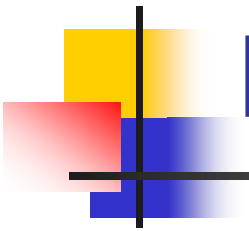




Using Certificates From Hierarchies

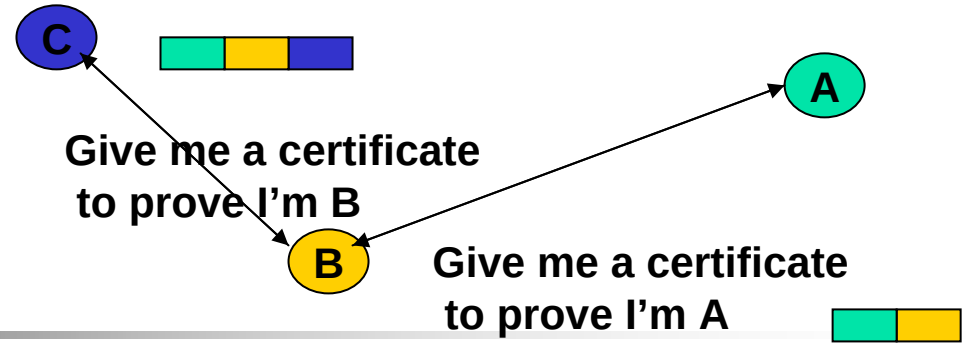
- I get a new certificate
- I don’t know the signing authority
- But the certificate also contains that authorities certificate
- Perhaps I know the authority who signed this authority’s certificate

Extracting the Authentication



- Using the public key of the higher level authority, extract the public key of the signing authority from the certificate
- Now I know his public key, and it is authenticated
- I can now extract the user's key and authenticate it

Example



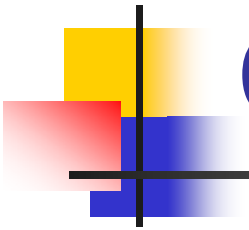
Alice gets a message with a certificate

Should Alice believe that he's really A ?

Then she uses B to check A

Alice has never heard of B but she has heard of C

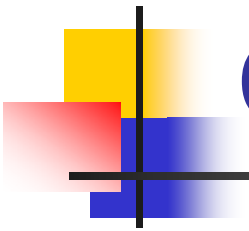
So she uses C to check B



Certificates and Trust

- Ultimately, the point of a certificate is to determine if something is trusted
  - Do I trust the request to perform some financial transaction?
  - So, Trustysign.com signed this certificate
  - How much confidence should I have in the certificate?

Potential Problems in the Certification Process

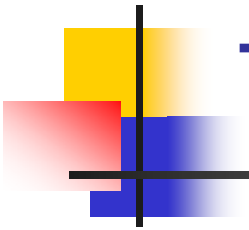


- What measures did Trustysign.com use before issuing the certificate?
- Is the certificate still valid?
- Is Trusysign.com's signature/certificate still valid?
- Who is trustworthy enough to be at the top of the hierarchy?

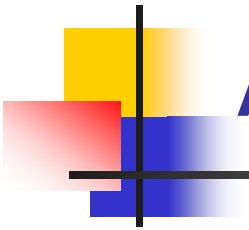
Trustworthiness of Certificate Authority

- How did Trustysign.com issue the certificate?
- Did it get an in-person sworn affidavit from the certificate's owner?
- Did it phone up the owner to verify it was him?
- Did it just accept the word of the requester that he was who he claimed to be ?

What Does a Certificate Really Tell Me?

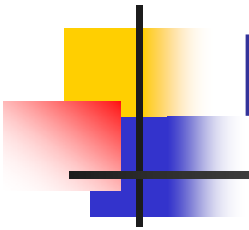


- That the CA tied a public/private key pair to identification information
- Generally does not tell why the CA thought the binding was proper
- I may have different standards than that Ca



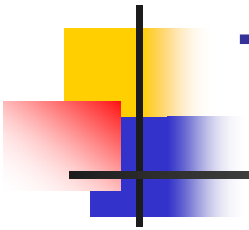
Another Problem

- Things change, and as a result what used to be trusted is not any more
- If there is trust-related information out in the network, what happens when things change?



Revocation

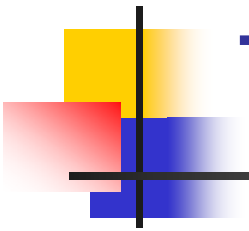
- A general problem for keys, certificates, access control lists, etc.
- How does the system revoke something related to trust?
- In a network environment
- Safely, efficiently, etc.



The Web of Trust Model

- Public keys are still passed around signed by others
- But your trust in others is based on your personal trust of them
  - Not on a formal certification hierarchy
  - “ I work in the office next to Bob, so I trust Bob’s certifications”

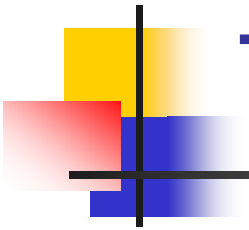
Certificates in the Web of Trust



- Any user can sign any other user's public key
- When a new user presents me his public key, he gives me one or more certificates signed by others
- If I trust any of those others, I trust the new user's public key

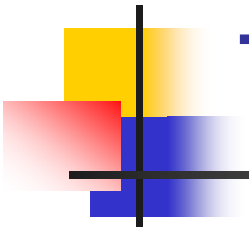
Advantages and Disadvantages of Web of Trust Model

- Scales very well
- No central authority
- Very flexible
  - May be hard to assign degrees of trust
  - Revocation may be difficult
  - May be hard to tell who you will and won't trust



Trust Models

- Refers to the type of relationship that can exist between people or organizations
- In the direct trust, a personal relationship exists between two individuals
- Third-party trust refers to a situation in which two individuals trust each other only because each individually trusts a third party



Types of Trust Models

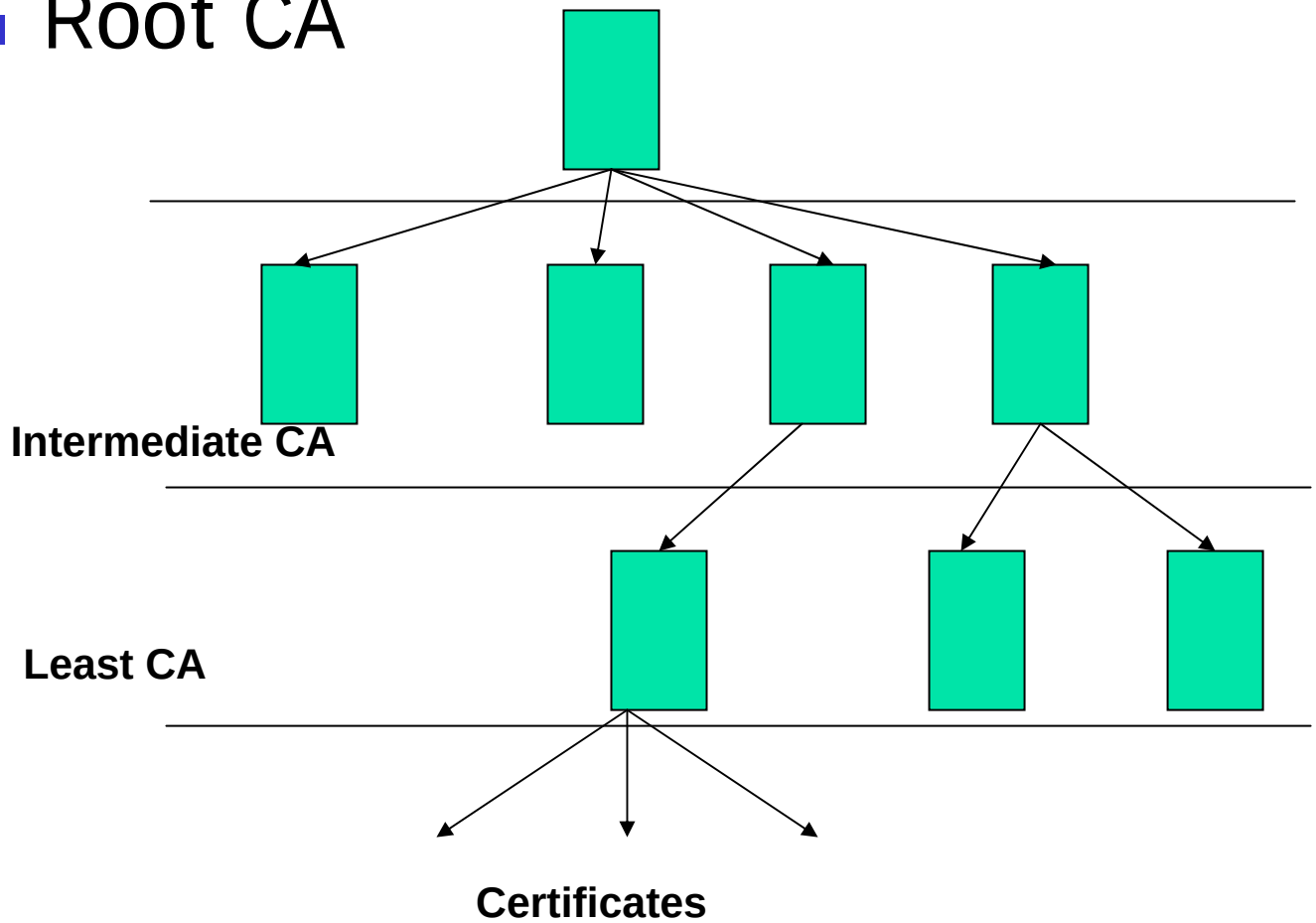
- The Four different PKI trust models are based on direct and third-party trust:
  - Hierarchical
  - Bridge
  - Mesh (Web) Peer-to-peer
  - Hybrid
- PKI was designed to allow all of these trust models to be created

Trust Models (cont.)

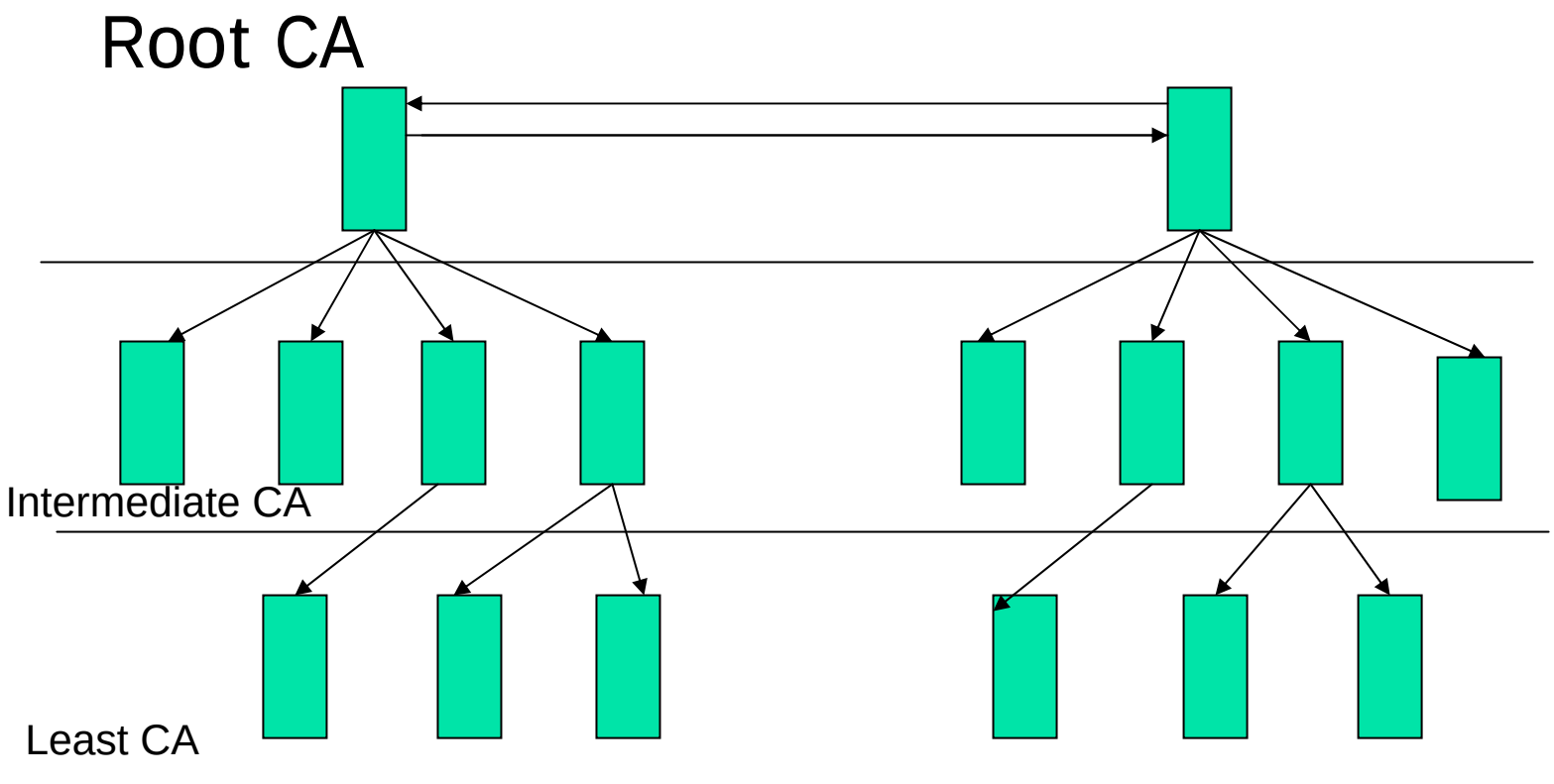
- **Hierarchical Trust Model :**
  - The most common implementation in a large organization that wants to extend its certificate-processing capabilities
  - Allows tight control over certificate-based activities
- **Bridge Trust Model :**
  - Flexibility and interoperability between organizations are the primary advantage
  - Lack of trustworthiness of the root Cas can be a major disadvantage
  - This model may be useful if you're dealing with a large, geographically dispersed organization
- **Mesh Trust Model :**
  - This model might be useful in a situation where several organizations must cross-certify certificates
  - The advantage is that you have more flexibility
  - The major disadvantage is that each root CA must be trustworthy in order to maintain security

Hierarchical Trust Model (Tree)

- Root CA

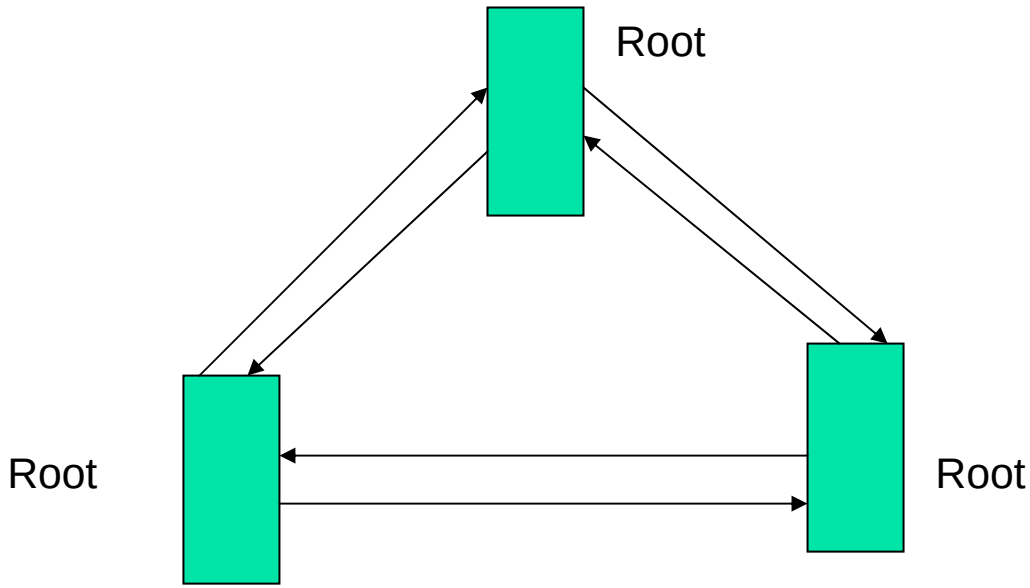


Bridge Trust Model



Mesh Model of Trust (Web of Trust)

- Each of the root CAs can communicate with the intermediate CAs in their respective hierarchies



Trust Models (continued)

- The web of trust model is based on direct trust
- Single-point trust model is based on third-party trust
  - A CA directly issues and signs certificates
- In an hierarchical trust model, the primary or root certificate authority issues and signs the certificates for CAs below it

Managing Digital Certificates

- After a user decides to trust a CA, they can download the digital certificate and public key from the CA and store them on their local computer
- CA certificates are issued by a CA directly to individuals
- Typically used to secure e-mail transmissions through S/MIME and SSL/TLS

Managing Digital Certificates (continued)

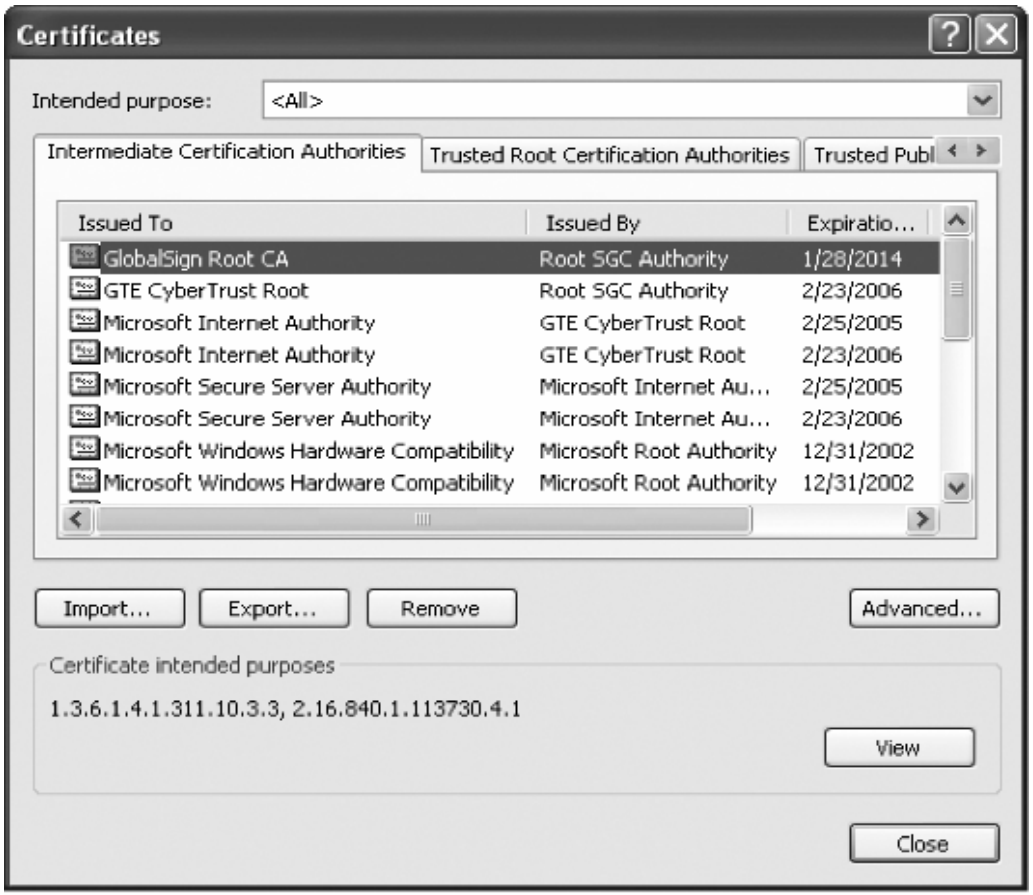
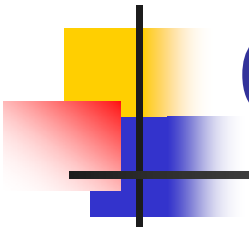


Figure Default CAs in Web browser

Managing Digital Certificates (continued)

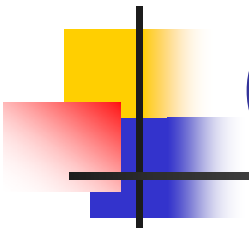
- Server certificates can be issued from a Web server, FTP server, or mail server to ensure a secure transmission
- Software publisher certificates are provided by software publishers to verify their programs are secure



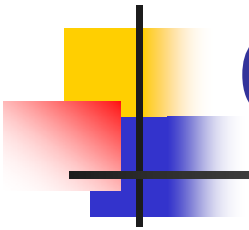
Certificate Policy (CP)

- Published set of rules that govern operation of a PKI
- Begins with an opening statement outlining its scope
- Published by the user of the CA

Certificate Practice Statement (CPS)

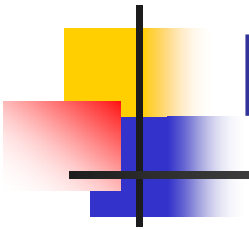


- More technical document compared to a CP
- Describes in detail how the CA uses and manages certificates



Certificate Life Cycle

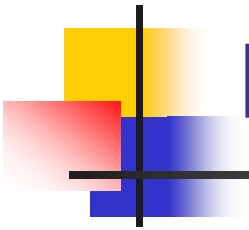
- Typically divided into four parts:
  - Creation
  - Revocation
  - Expiration
  - Suspension



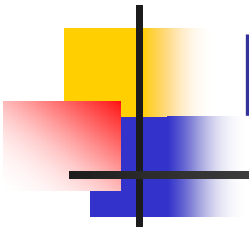
Exploring Key Management

- Because keys form the very foundation of the algorithms in asymmetric and PKI systems, it is vital that they be carefully managed

Centralized and Decentralized Management

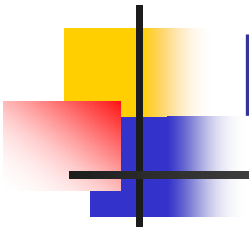


- Key management can either be centralized or decentralized
- An example of a decentralized key management system is the PKI web of trust model
- Centralized key management is the foundation for single-point trust models and hierarchical trust models, with keys being distributed by the CA



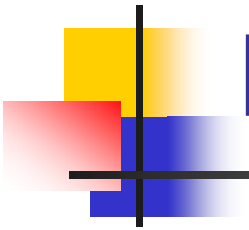
Key Storage

- It is possible to store public keys by embedding them within digital certificates
- This is a form of software-based storage and doesn’t involve any cryptography hardware
- Another form of software-based storage involves storing private keys on the user’s local computer



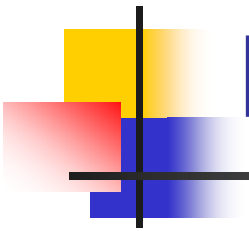
Key Storage (continued)

- Storing keys in hardware is an alternative to software-based keys
- Whether private keys are stored in hardware or software, it is important that they be adequately protected



Key Usage

- If you desire more security than a single set of public and private (single-dual) keys can offer, you can choose to use multiple pairs of dual keys
- One pair of keys may be used to encrypt information and the public key could be backed up to another location
- The second pair would be used only for digital signatures and the public key in that pair would never be backed up



Key Handling Procedures

- Certain procedures can help ensure that keys are properly handled:
  - Escrow
  - Expiration
  - Renewal
  - Revocation
  - Recovery
  - Suspension
  - Destruction