Topic 7: Public Key Infrastructure (PKI)

Understand the PKI technologies for secure distribution of public keys

Source: Stalling's book, chapter 14; also lots of docs on this subject on the Internet.

Overview

- ❑ Part 1
  - ➢ Public Key Infrastructures (PKI) Overview
- ❑ Part 2
  - ➢ Digital Certificates
- ❑ Part 3
  - ➢ Certificate Revocation Lists (CRLs)
- ❑ Part 4
  - ➢ Certificate Hierarchies
  - ➢ Conclusions

PKI Overview

- ❑ PKI
  - ❍ provides functions, technologies, policies and services that enable practical deployment and wide-scale applications of public-key cryptography (PKC).
  - ❍ includes the management and control of public and private keys.

- ❑ Security properties/services offered by PKC include:
  - ➢ Certificate-based user/entity authentication.
  - ➢ Digital signing of electronic documents, emails, software for authentication (integrity) and non-repudiation protections.
  - ➢ Encryption, typically for symmetric key distributions.

PKI Overview

❑ Applications of PKI around us:

- ❍ Web browsers, servers and services, e.g. SSL (secure socket layer).
- ❍ Virtual Private Networks (VPNs), e.g. IPSec.
- ❍ Secure email services, e.g. S/MIME, PGP (Pretty Good Privacy).
- ❍ Secure file storage services, e.g. PGP.
- ❍ Secure electronic transactions, e.g. SET.
- ❍ Visa/Master smartcards.
- ❍ Copyright protection (DRM – Digital Right Management).
- ❍ …

PKI Overview

- When using public-key cryptography, two major issues should be considered:
  - **Issue 1:** How to ensure the security (secrecy and strength) of the private key.
    - The key size should be large enough.
    - The lifetime of the key should guard against brute-force attacks.
    - The key should be kept secret; they should be generated, transported, stored and destroyed (at the end of its lifetime) securely.

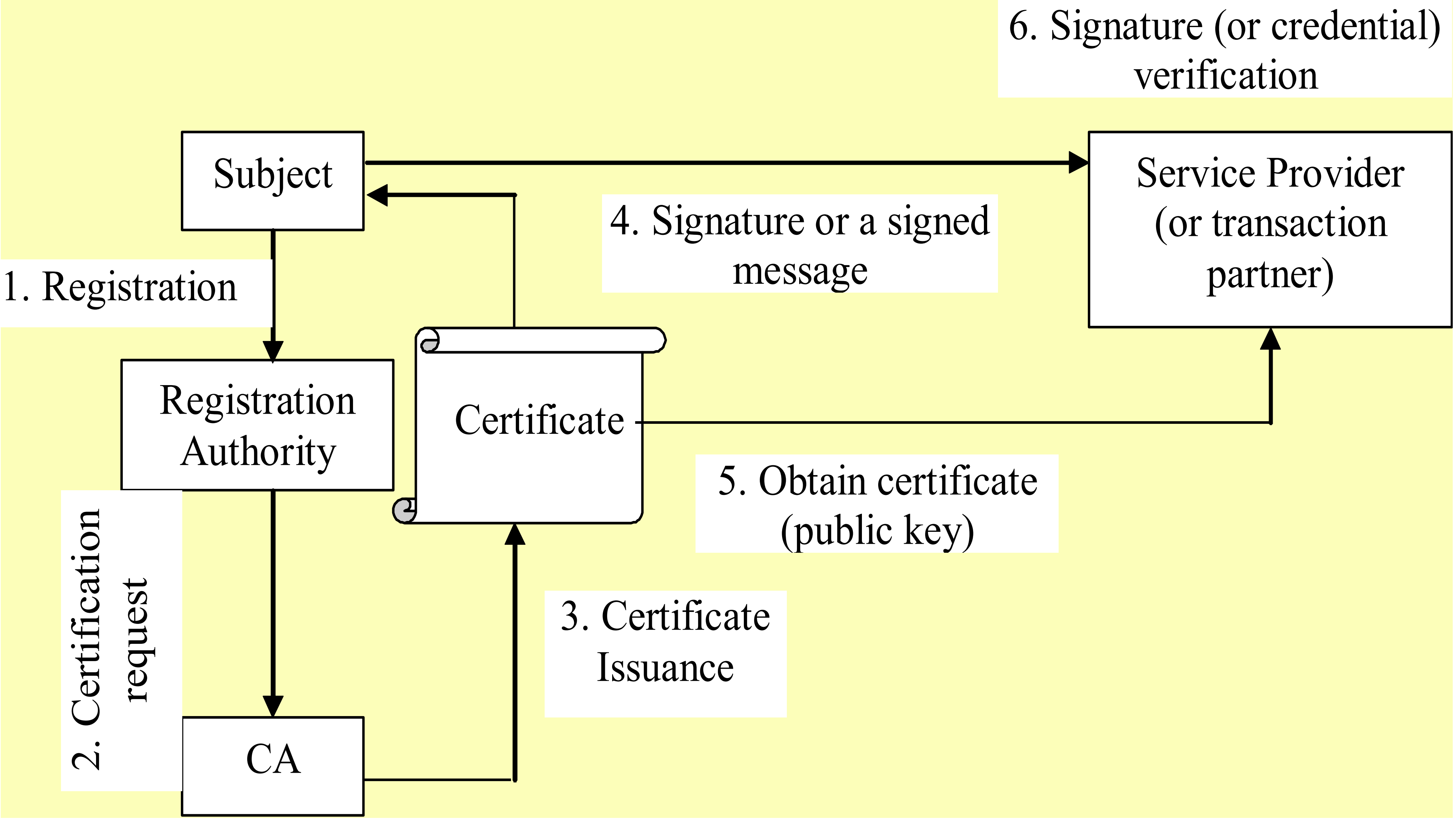
PKI Overview

- ❍ **Issue 2**: How to ensure that a public key is trustworthy, i.e. how could we trust that a given public key indeed belongs to a claimed entity.
  - ➢ The solution is to have some trusted entity or authority to sign one's public key ➔ **digital certificate**.
  - ➢ Otherwise, communications are vulnerable to man-in-the-middle attack.
- ❑ A digital/PKI certificate is **a statement**:
  - ❍ certifying that this public key indeed belongs to this identity
  - ❍ the owner of this identity possesses the corresponding private key.
- ❑ When one uses a digital certificate, s/he must demonstrate that s/he knows the corresponding private key.
- ❑ Digital/PKI Credential = PKI certificate + the matching private key

PKI Overview - Main PKI entities

- ❑ **Registration Authority (RA)**: verifying the identity of a user requesting for a certificate.
- ❑ **Certificate Authority (CA)**: issuing and managing digital credentials
  - ❍ credential=private key + certificate
  - ❍ Key pair can be generated by a CA or by the requester
- ❑ **Data Repository**: typically a LDAP directory, is where certificates and revocation status are *officially* stored.

PKI Overview – A simplified view of acquiring a Cert for signature purpose



PKI Overview – A simplified view of acquiring a Cert for signature purpose

- ❑ Assumptions used:
  - ❍‘You’ (Subject) and ‘the Service Provider (SP)’ do not trust (or donot know) each other and You want to send a signed message to SP.
  - ❍You have already got a pair of private and public keys and need to get your public key certified (a certificate for the public key).
  - ❍CA is trusted by both You and the SP.

- ❑ Pls note: SP should ALSO have the CA’s certificate (why?).

PKI – Main Functions

- SystemSetup: a **credential** service provider (usually CA) should get the policy, procedures and services ready, including key generation/update, **certificates** issuance, distribution and revocation, possibly key recovery, and potential interaction with other providers, e.g. with a registration authority (RA) and other CAs.

PKI Overview – Main Functions

- **SubjectRegistration**: during this process, a subject makes her/himself known to a RA/CA:
  - **Enrollment:** An applicant, e.g. *Alice,* may need to provide the following information (*depending on classes of certificates*):
    - Proof of *Alice*' s identity (email address, driving license, birth certificate, fingerprints, passport, NI number, etc).
    - Alice' s public key, KU_{Alice}
  - **Authenticate applications**
    - share information with a third-party database.
    - personal appearance (use of Local Registration Authority).

PKI Overview – Main Functions

- ❑ KeyGeneration: a pair of crypto keys are generated **either by the subject or by the CA**, and the CA will certify the public key of the pair.
- ❑ CertificateIssuance (Certification): the CA issues a certificate for a subject' s public key.
- ❑ CertificateVerification (proving the possession of credential): this is performed when a certificate is used to access a service or to perform a transaction.

PKI Overview – Main Functions

- ❑ CertificateRevocation: if the private key associated to the public key certified in the certificate is compromised or suspected of being compromised, then the certificate should be revoked.
- ❑ Cross-certification: is an operation to allow a pair of CAs to establish a trust relationship through the signing of each other' s public keys in a certificate.

Part 2 Overview

❑ Digital Certificates

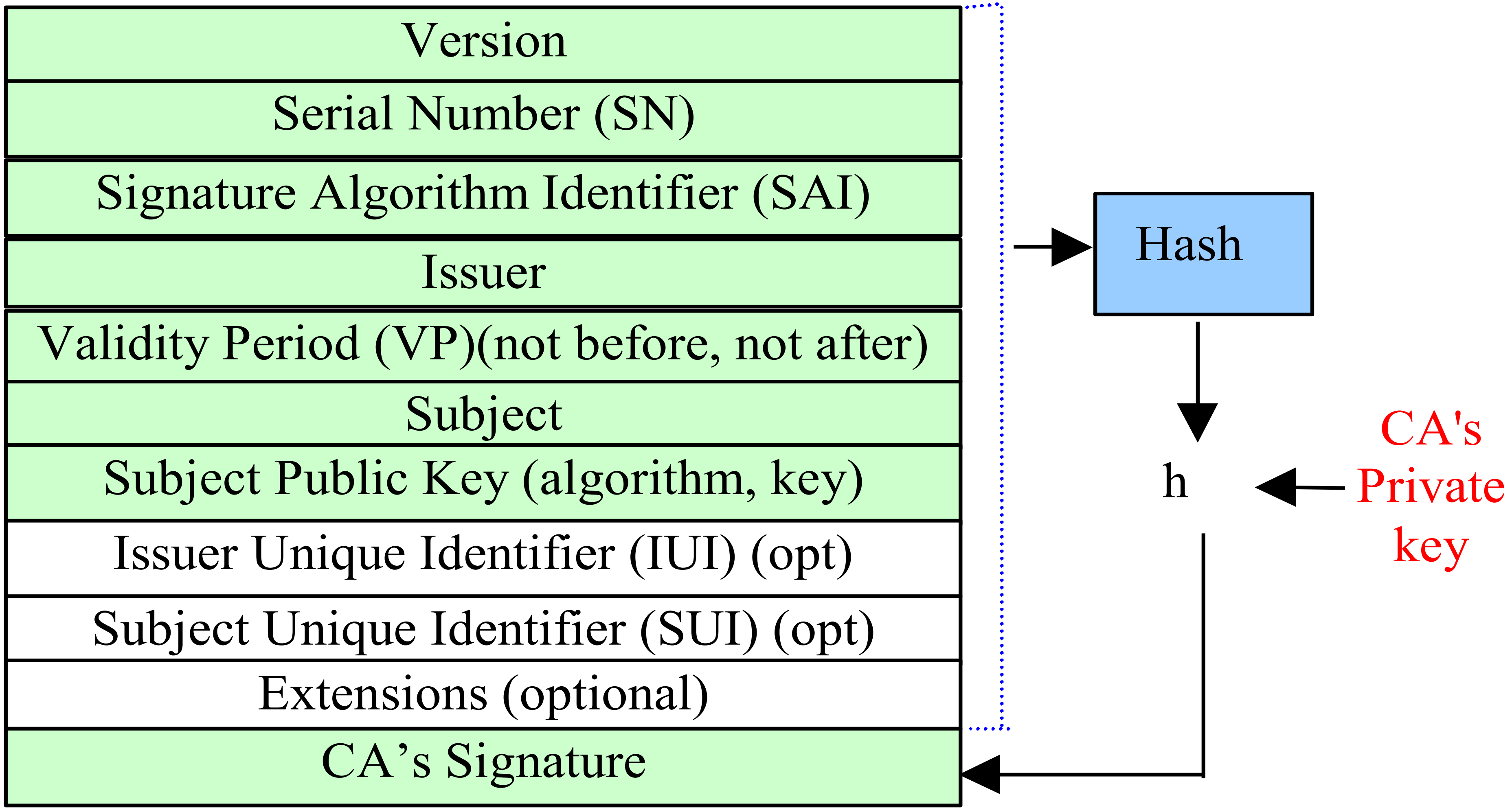
Digital Certificates

❑ Certification is a secure and scalable way of distributing public keys.

❑ A digital certificate (or *public-key certificate*, *digital ID*, *certificate*)

- ❍binds an entity's public key (+ one/more attributes) to its identity (the entity = person, hardware device, software process).
- ❍is digitally signed by the CA so you need CA's public key to verify the certificate.
- ❍its contents are application dependent, e.g. a certificate for secure email contains the entity's email address, a certificate for financial purpose may contain credit card number and credit limit, etc.

Digital Certificates - the X.509 v3 certificate format



Version

Serial Number (SN)

Signature Algorithm Identifier (SAI)

Issuer

Validity Period (VP)(not before, not after)

Subject

Subject Public Key (algorithm, key)

Issuer Unique Identifier (IUI) (opt)

Subject Unique Identifier (SUI) (opt)

Extensions (optional)

CA’s Signature

Digital Certificates - the X.509 v3 certificate format

- Version: current values are v1, v2, v3.
- SN: unique identifier for each certificate generated by issuer (CA).
- SAI: identifying the algorithm, such as RSA or DSA, used by the CA to sign the certificate.
- Issuer: the issuer's name (X.500 ‘distinguished name’).
- VP: a range of time when the certificate is valid.
- Subject: the subject's name (X.500 ‘distinguished name’).
- SPK: the subject's public key and parameters, and the identifier of the algorithm with which the key is used.
- IUI: to allow the reuse of issuer names over time.
- SUI: to allow the reuse of subject names over time.
- Ext: provide a way to associate additional information for subjects, public keys, managing the certification hierarchy

Digital Certificates - An example

Version: 3

Serial Number (SN): 02:41:00:00:01

Signature Algorithm Identifier (SAI):
MD5 digest with RSA encryption

Issuer: C=US, O=RSA Data Security, Inc.,
OU=Secure Server Certification Authority

Validity Period (VP):
---Not Before Date: 16/5/96 12:00:00 AM
---Not After Date: 17/5/96 11:59:59 PM

Subject: C=GB, O=Manchester Univ,
OU=Computer Science

Subject Public Key (SPK):
Public key algorithm: RSA Encryption
Public key: Modulus: 00:92:.....(typically 200 digits)
Exponent: 65537

CA's Signature: 88:d1:.....

Part 3 Overview

❑ Certificate Revocation Lists (CRLs)

CRLs – What is it and why do we need it?

- ❑ CRL is a mechanism to let the world know that certificates are no longer valid. It is a black list of revoked certificates (i.e. prematurely terminated certificates).

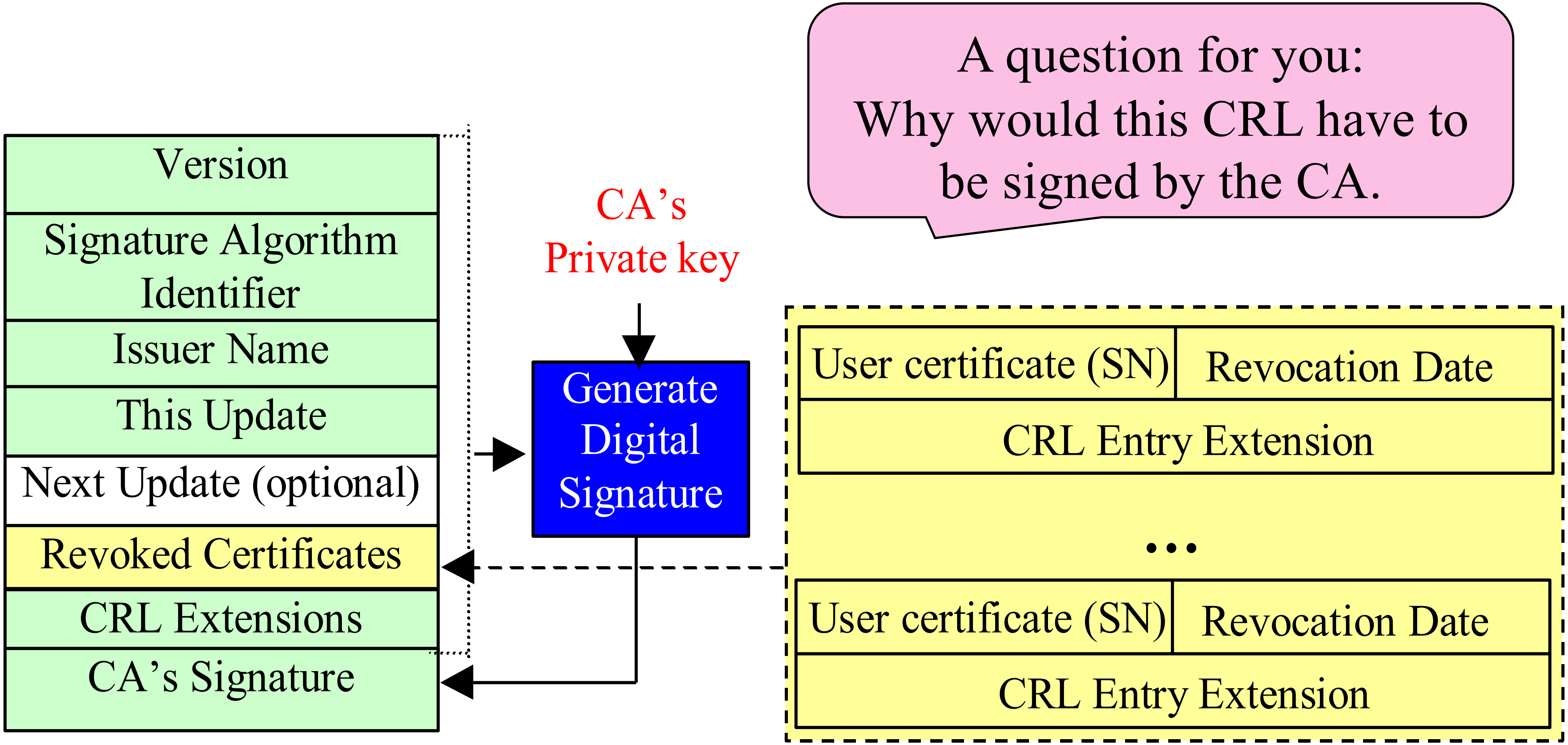
- ❑ Reasons for revocation include:
    - ❍ The corresponding private key has been compromised.
    - ❍ CA may have been compromised.
    - ❍ Subject’s affiliation has changed.
    - ❍ Key/certificate no longer needed.
    - ❍ …

- ❑ Required to reduce
    - ❍ risk of impersonation attacks.
    - ❍ risk of repudiation attacks.

CRLs – How to revoke it?

- ❑ A CRL is a data structure, digitally signed by the issuing CA, containing:
  - ❍date and time of the CRL publications.
  - ❍name of the issuing CA.
  - ❍serial numbers of all the revoked certificates.

CRLs - X.509v2 CRL format



Version

Signature Algorithm Identifier

Issuer Name

This Update

Next Update (optional)

Revoked Certificates

CRL Extensions

CA's Signature

CA's Private key

Generate Digital Signature

User certificate (SN) | Revocation Date

CRL Entry Extension

…

User certificate (SN) | Revocation Date

CRL Entry Extension

A question for you:
Why would this CRL have to be signed by the CA.

CRLs - X.509v2 CRL format

- Version: v2 should be used if any extension field are present. Otherwise, it can be omitted.
- Issuer Name: the entity that issued and signed the CRL.
- This Update: the date/time of issue of this CRL.
- Next Update: the date/time of issue of next CRL. The next CRL could be issued prior to, but not after, the indicated date.
- User Certificate SN: certificate serial number of a revoked certificate.
- Revocation Date: the effective date of a revocation.
- Extension: X.509 v2 CRL Entry Extension fields have the same sub-fields as X.509 v3 certificates.

CRLs – Deployment issues

- ❑ Using CRL is not that straightforward
  - ❍ The issuing CA needs to keep the CRL up-to-date.
  - ❍ A certificate-using application should obtain the most recent CRL and ensure that the certificate serial number is not on the CRL list; in other words, a certificate is said to be valid *iif* the following verifications are positive:
    - ➢ It has a valid CA signature,
    - ➢ It has not expired, and
    - ➢ It is not listed in the CA's most recent CRL.
  - ❍ There are some scalability issues.

Part 4 Overview

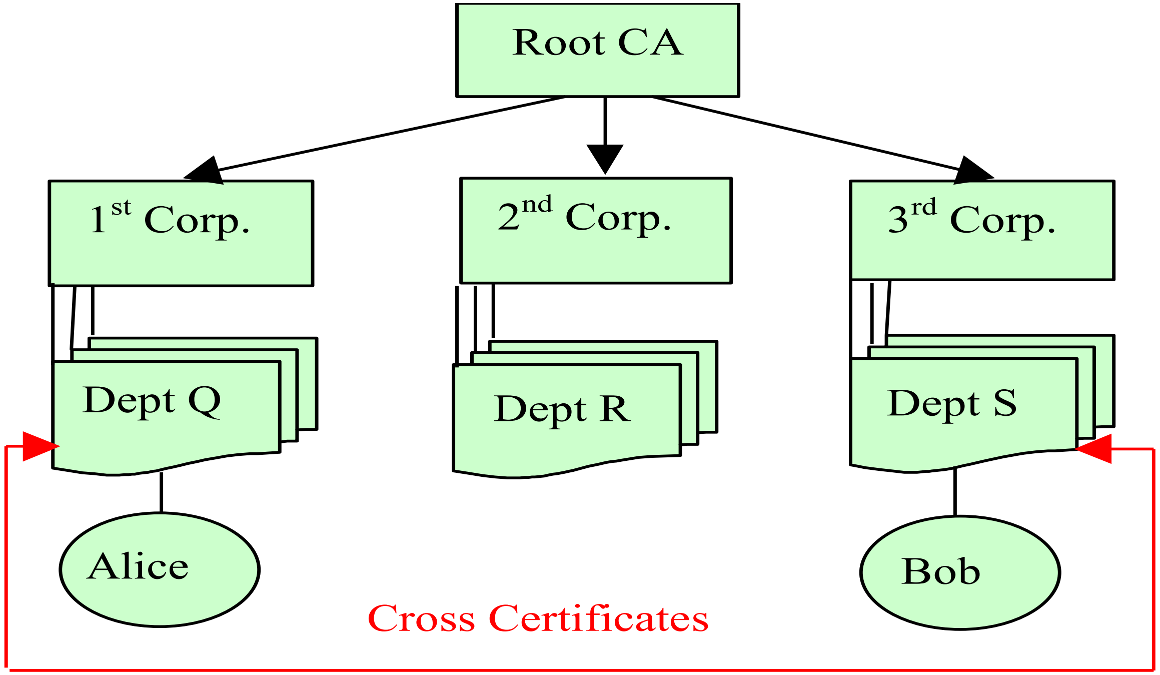
- Certificate Hierarchies
  - About how multiple CAs are organised/used
  - Two trust models
    - Top-down Certificate Hierarchy (Hierarchical Trust Model); used in X.509 PKI
    - Bottom-up Certificate Hierarchy (Peer-to-Peer Trust Model or web-of-trust); used in PGP
- Conclusions

Top-down Certificate Hierarchy

- ❑ In most cases, we use more than one CAs, as using one root key to sign certificates
  - ❍ is too risky if that one key is compromised.
  - ❍ is not scalable when user base is large.
- ❑ In some cases, certificate managements may resemble the management structure of an organisation, as depicted in the next slide.
- ❑ Certificate hierarchy
  - ❍ Start with a root CA (trust anchor) with a root cert/key pair (root-public-key, root-private-key).
  - ❍ **Delegate** signing power to subordinate CAs (create more key pairs, sign their public keys with root-private-key, ...)
- ❑ The fact that one authority, CA_U, signs on another authority CA_W's cert, $Sign_U(Cert_W)$, signifies that CA_U trusts CA_W.

Top-down Certificate Hierarchy

- ❑ RootCA generates certificates for intermediate CAs.
- ❑ Intermediate CAs generate certificates for the leaf CAs.
- ❑ Leaf CAs generate certificates for the end-entities (users, devices, and applications).



- ❑ Alice’s Certificate Chain: $\{CERT_{Alice}\}S_{DeptQ}+\{CERT_{DeptQ}\}S_{1stCorp}+\{CERT_{1stCorp}\}S_{RootCA}$
- ❑ If Bob wishes to authenticate a message signed by Alice, he can proceed ‘up’ the certificate chain until he finds a certificate he can trust.

Top-down Certificate Hierarchy

- Validating a cert possibly involves validating a chain of certs (called chain of trust).
  - verify all the digital certificates, including the signatures signed by all subordinate CAs in a bottom-up manner until you reach the root CA's signature, or until you reach a subordinate CA that you can trust.

Top-down Certificate Hierarchy - Cross certification

- In this example, the 3rd Corp' s Dept S has certified the 1st Corp' s Dept Q.
- So, Alice' s Certification Chain with **cross certification** is:

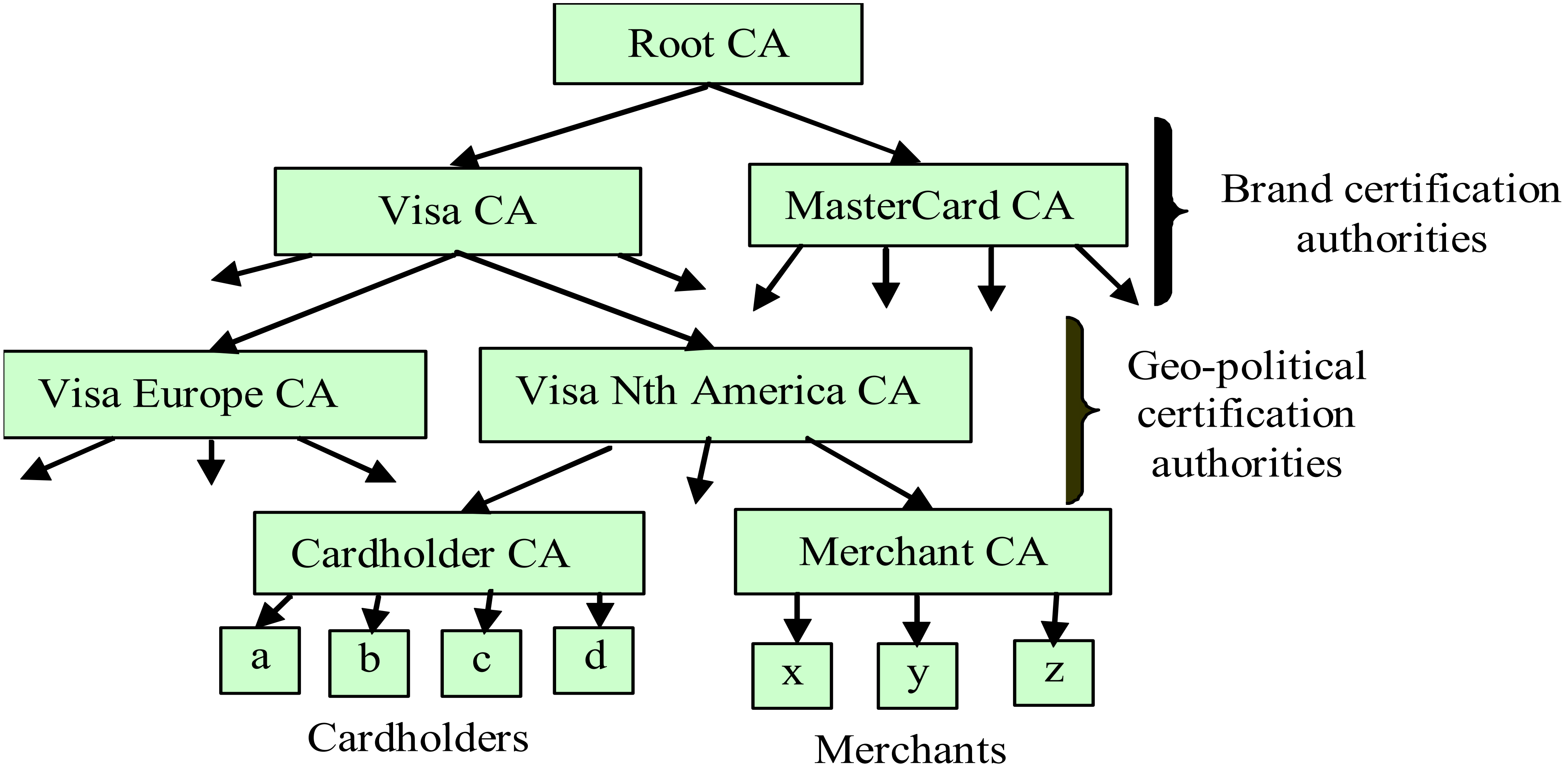
$$\{CERT_{Alice}\}S_{DeptQ}+\{CERT_{DeptQ}\}S_{1stCorp}+\{CERT_{DeptQ}\}S_{DeptS}+\{CERT_{1stCorp}\}S_{RootCA}+\{CERT_{DeptS}\}S_{3rdCorp}+\{CERT_{3rdCorp}\}S_{RootCA}$$

- Now Bob only has to go up Alice' s Certificate Chain to find his dept' s certificate.
- Cross certification provides efficient certificate verification.

Top-down Certificate Hierarchy

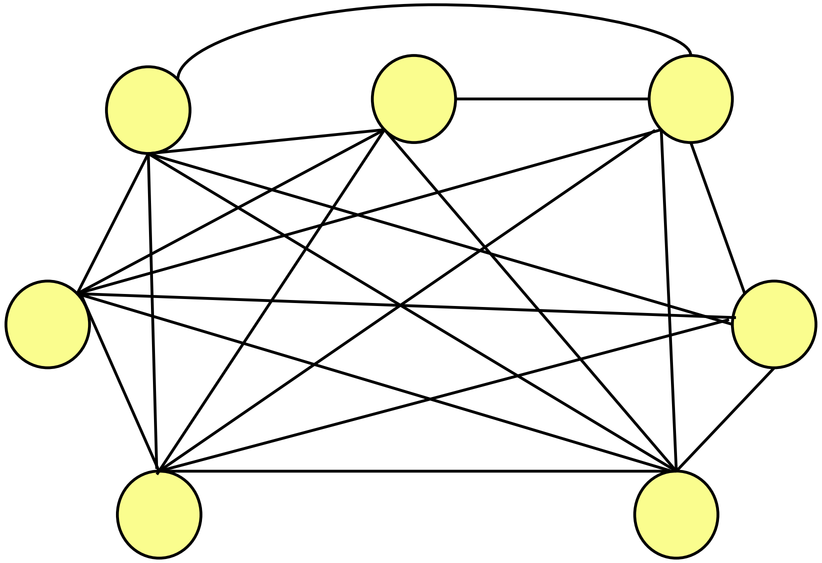
- ❑ Certificate types:
  - ❍ CA certificates: self-signed (a standalone or root CA), or issued by a superior CA within a hierarchy.
  - ❍ End-entity certificates: issued by a CA to subjects.
  - ❍ Cross-certification certificates: signed by a peer CA (independent CAs sign each other’ s certificates to establish peer-to-peer trust relationships).

Top-down Certificate Hierarchy - An Example (SET)



Bottom-up Certificate Hierarchy

- ❑ There is no trusted anchor among the CAs.
- ❑ Usually end-entities sign certificates for (other) entities they know (serve as CAs).
- ❑ Need a mechanism to assess the trust level of each CA/certificate.
- ❑ Used in the PGP (Pretty Good Privacy) solution.
- ❑ Potentially a fully meshed structure - not scalable.



Exercise Question – E7.1

(a) Investigate an on-line CA and find out what process or procedures that are necessary for you to acquire a public key certificate, how many classes of certificates and what each class can be used for.

(b) X.509 is a top-down approach to public key management. Investigate and describe a bottom-up approach to public key management.

Exercise Question – E7.2

Assuming that Alice has sent a signed message to Bob.
(i) Highlight the steps for verifying a digital certificate.
(ii) Highlight the steps Bob takes to verify the authenticity of the message from Alice.

Conclusions

- ❑ Digital certificates allows us to bind a public key to its rightful owner.
- ❑ This binding of key with identity allows us to solve the problem of how to distribution authentic public keys.
- ❑ Various PKI systems have been proposed - X509 works in a top-down manner.
- ❑ A *CA* is primarily responsible for issuing certificates and ensuring the validity of the certificates issued.