Topic 7

Public Key Infrastructure (PKI)

Understand the main components of PKI technology

Overview

- Public Key Infrastructures (PKI) Overview
- Digital Certificates
- Certificate Revocation Lists (CRLs)
- Top-down Certificate Hierarchy
- Conclusions
- *source: chapter 14 of Cryptography and Network Security*

Lots of docs on this subject on the Internet!

PKI Overview

- PKI provides functions, technologies, policies and services that enable practical deployment and wide-scale application of public-key cryptography.

- Security services supported:
  - 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).
  - ….. etc.

PKI Overview

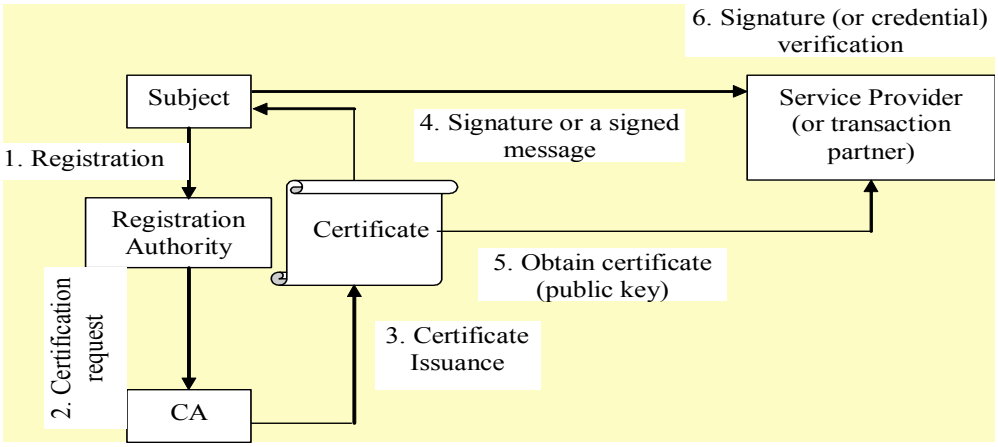
- The major problem with wide-scale application of public-key cryptography is:
  - How could we trust that a given public key belongs to the claimed entity, i.e. secure public key distribution.
- The solution is to have some one or some authority to sign one's public key ➔ **digital certificate**.
- A digital certificate is **a statement**:
  - certifying that this public key belongs to this identity, and
  - the owner with this identity possesses the corresponding private key.

PKI Overview – Two trust models

- SPKI (Simple PKI): a bottom-up approach
  - Uses a Web-of-trust model.
  - Public keys are signed/certified by friends or friends' friends.
  - You are supposed to trust some of the friends.
  - Used in the PGP (Pretty Good Privacy) solution.
- X509 PKI: a top-down approach
  - Public keys are signed/certified by trusted authorities, Certification Authorities (CAs)
  - A CA or CA hierarchy digitally sign keys in a top-down manner.

PKI Overview

- Typical PKI entities:



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.
- SubjectRegistration: during this process, a subject makes itself known to a CA.
- 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.

PKI Overview – Main functions

- CredentialIssuance (Certification): the CA issues a certificate for a subject's public key.
- CredentialVerification (proving the possession of credentials): this is performed when a credential is used to access a service or to perform a transaction.
- CredentialRevocation: if the private key associated to the public key certified in the certificate is compromised or suspected 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 called **cross-certification**.

PKI Overview – Main functions

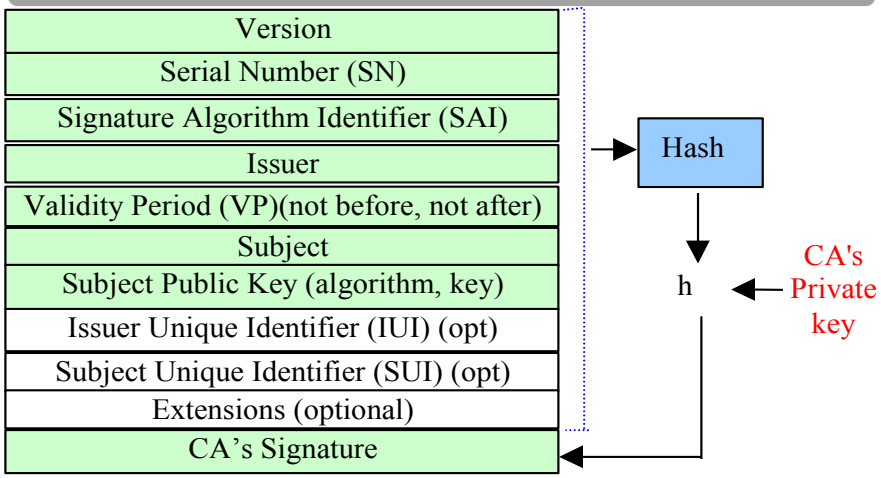
- **SubjectRegistration:**
  - **Enrollment:** An applicant, e.g. *Alice,* may need to provide the following information (*depending on classes of certificates*):
    - Proof of *Alice*'s identity (email address, driving license, birth certificate, fingerprints, passport, NI number, etc).
    - Alice's public key, KU_{Alice}
  - **Authenticates applications**
    - share information with a third-party database.
    - personal appearance (use of Local Registration Authority).
- Data Repository, typically a LDAP directory, is where certificates and revocation status are *officially* stored.

Homework: try an on-line CA and find out what procedures are necessary for you to acquire a certificate for your public key, and how much does each issue cost!

Digital Certificates - Overview

- 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



Digital Certificates - the X.509 v3 certificate format

- Version: current values are v1, v2, v3.
- SN: a number unique to the issuer (CA).
- SAI: identifies the algorithm, such as RSA or DSA, used by the CA to sign the certificate.
- Issuer: the issuer's name.
- VP: a range of time when the certificate is valid.
- Subject: the subject's 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 and certification revocation lists.

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

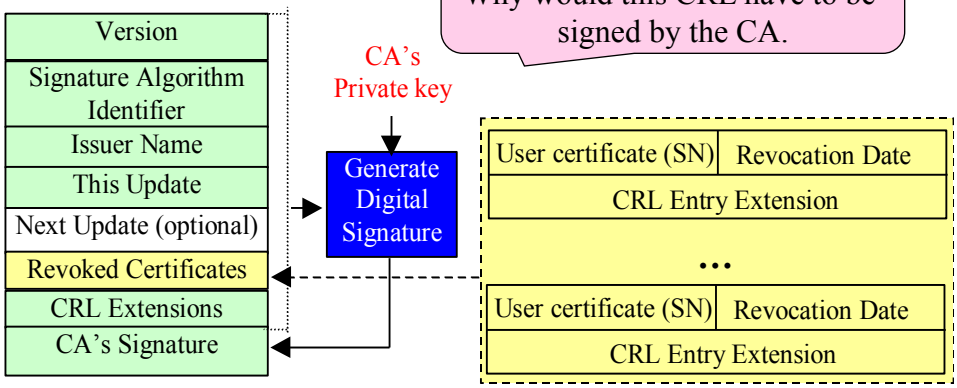
CA's Signature: 88:d1:.....

CRLs - Why do we need it?

- A certificate has a validity period, {*notBeforeDate, notAfterDate*}.
  - But what if a private key corresponding to a public key certified in a certificate is compromised before the expiration date?
  - Vulnerable to repudiation attacks.
- Certificate Revocation Lists (CRLs)
  - CRL is a mechanism to let the world know that a certificate is no longer valid. It is a black list of revoked certificates (i.e. prematurely terminated certificates).
  - 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

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.
- That is why short expiration policies are important.

Homework: check your browser and see whether this facility has been enabled!

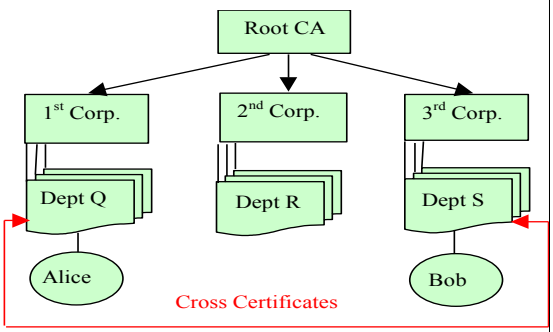
Top-down Certificate Hierarchy

- 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 with a root cert/key.
  - Create more keys, sign them with root key, and delegate them to subordinate CAs.
- Validating a cert possibly involves validating a chain of certs (called chain of trust).

Top-down Certificate Hierarchy

- Certificate chain validation: verify all the digital 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!



- 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.

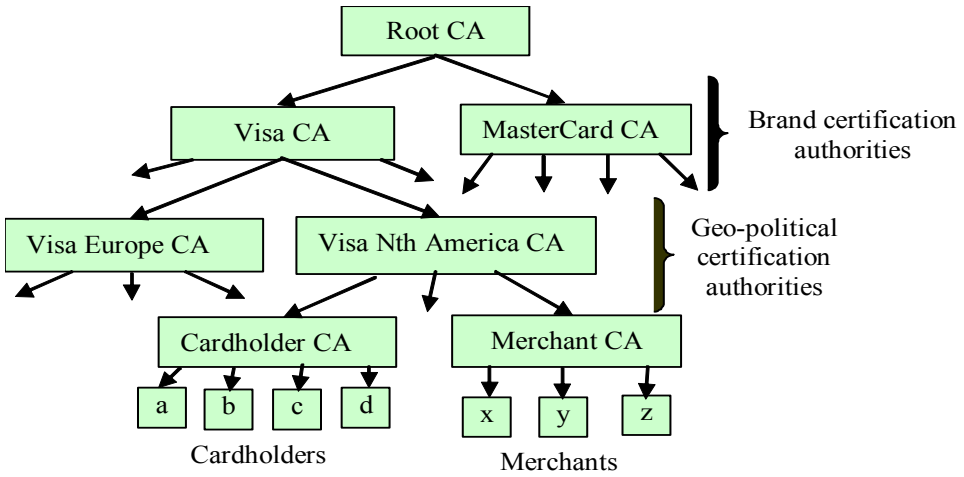
Top-down Certificate Hierarchy - Cross certification

- 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 - An Example (SET)



Exercise 7(a)

Suppose that KU_A and KR_A are the public and private keys of a party A respectively, that KU_B and KR_B are those of a party B, and that each of A and B can use any cryptosystems.

(i)If A wants to send a very long message to B, suggest an encryption method by which only B can decrypt the message and the encryption/decryption processes are the most efficient.

(ii)Can A encrypt a message so that anyone receiving the message will be assured that the message came only from A (i.e. authenticity protection)? If yes, give your method; and if not, explain why not.

(iii) Suggest an *efficient* method by which both confidentiality and authenticity protections are provided.

Exercise 7(b)

- **PKCS** refers to a set of public-key cryptography standards, defined and published by RSA Security; visit http://en.wikipedia.org/wiki/PKCS and **read about** PKCS#12 and public-key certificates.
- **Investigate procedures/protocols for PKI certificate acquisition.**

- Use the *PKI* module in CrypTool to generate and import an RSA key pair, and use the keys to do encryption and signature generation and verification.
- Again you can find this module via Menu: "Digital Signatures/PKI" \"Generate/Import Keys".

Conclusion

- 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 responsible for issuing certificates and periodical publishing *CRL* - revocation notification of compromised corresponding private keys.