KF School of Computing and Information Sciences
Florida International University

CNT 4403
Computing and Network Security

Key Management – Public Key Infrastructure

Dr. Kemal Akkaya

E-mail: *kakkaya@fiu.edu*

Public Key Infrastructure

- **A system (or infrastructure) to securely distribute & manage public keys**
  - Public keys are stored in Certificates
- **Important for wide-area trust management (e.g., for Internet transactions)**
- **Ideally consists of**
  - a certification authority (CA)
  - certificate repositories
  - a certificate revocation mechanism (CRLs, etc.)
- **Many models possible: monopoly, oligarchy, anarchy, etc.**
  - Trust issues

Monopoly Model

- **Single organization is the CA for everyone**
  - Everybody gets the certificate directly from the CA
- **Shortcomings:**
  - no such universally-trusted organization
  - requires everyone to authenticate physically with the same CA – not scalable
  - once established, CA can abuse its position (excessive pricing, etc.)
  - requires perfect security at CA

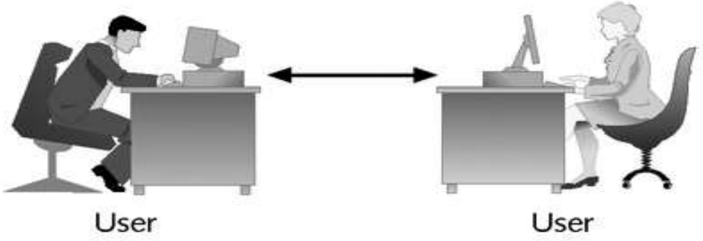
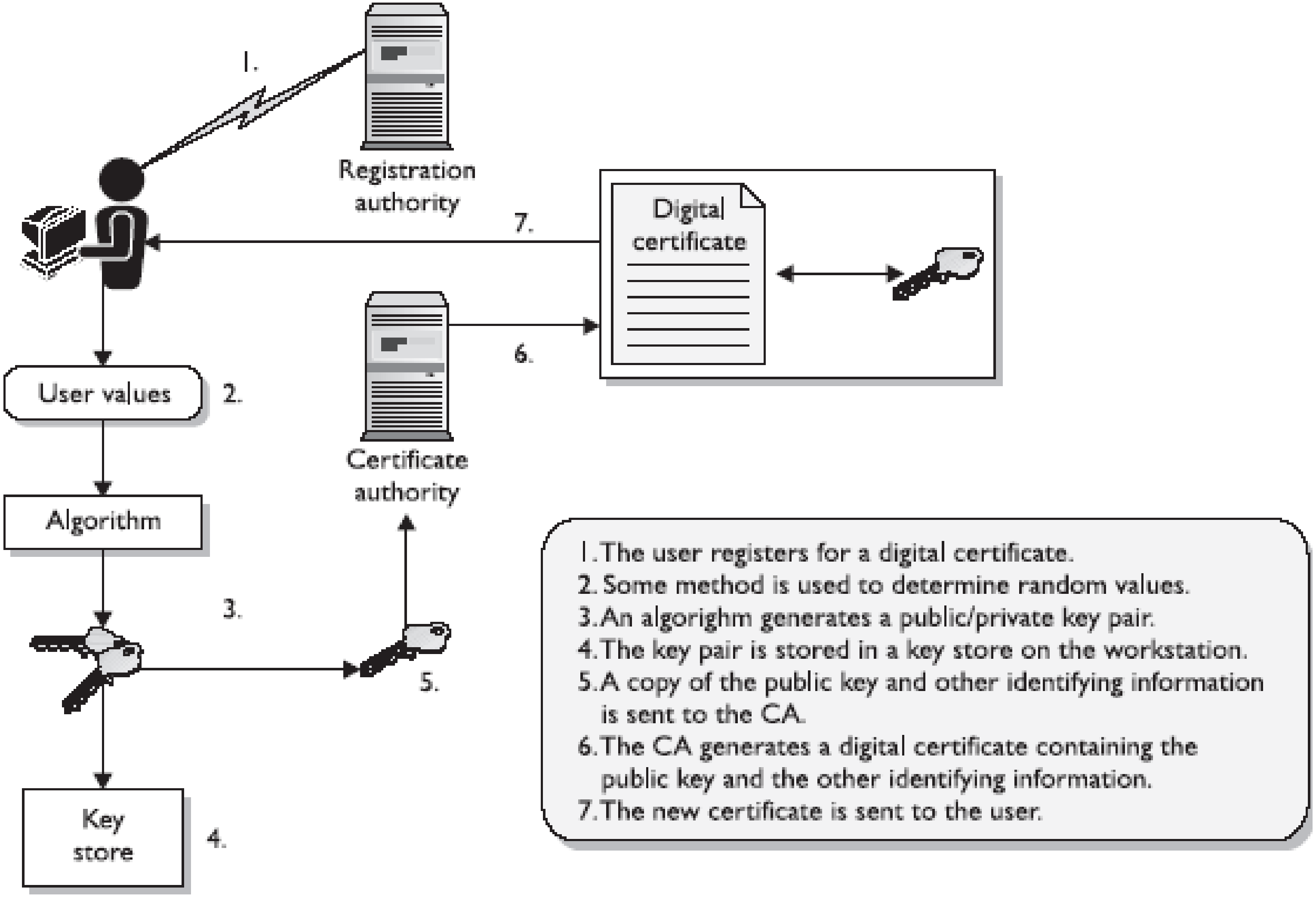


Figure 14-5 Direct trust

Monopoly with Registration Authorities

- CA trusts other organizations called Registration Authorities (RAs) to check identities, do the initial authentication
- RAs support all or some of:
  - Identification
  - User key generation/distribution
  - Interface to CA
  - Key/certificate management
- Solves the problem of physically meeting the CA
  - Other problems remain

Steps for obtaining a digital certificate via RAs



1. The user registers for a digital certificate.
2. Some method is used to determine random values.
3. An algorighm generates a public/private key pair.
4. The key pair is stored in a key store on the workstation.
5. A copy of the public key and other identifying information is sent to the CA.
6. The CA generates a digital certificate containing the public key and the other identifying information.
7. The new certificate is sent to the user.

Delegated CAs

- Root CA certifies lower-level CAs to certify others
- All verifiers trust the root CA & verify certificate chains beginning at the root (i.e., the root CA is the *trust anchor* of all verifiers)
- E.g., a national PKI, where a root CA certifies institutions, ISPs, universities who in turn certify their members
- Limitations are similar to monopoly with RAs

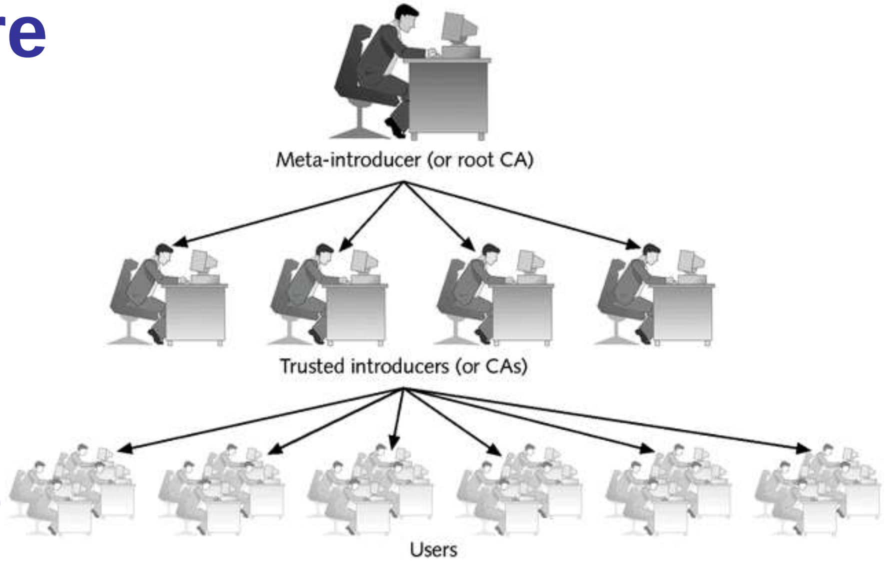


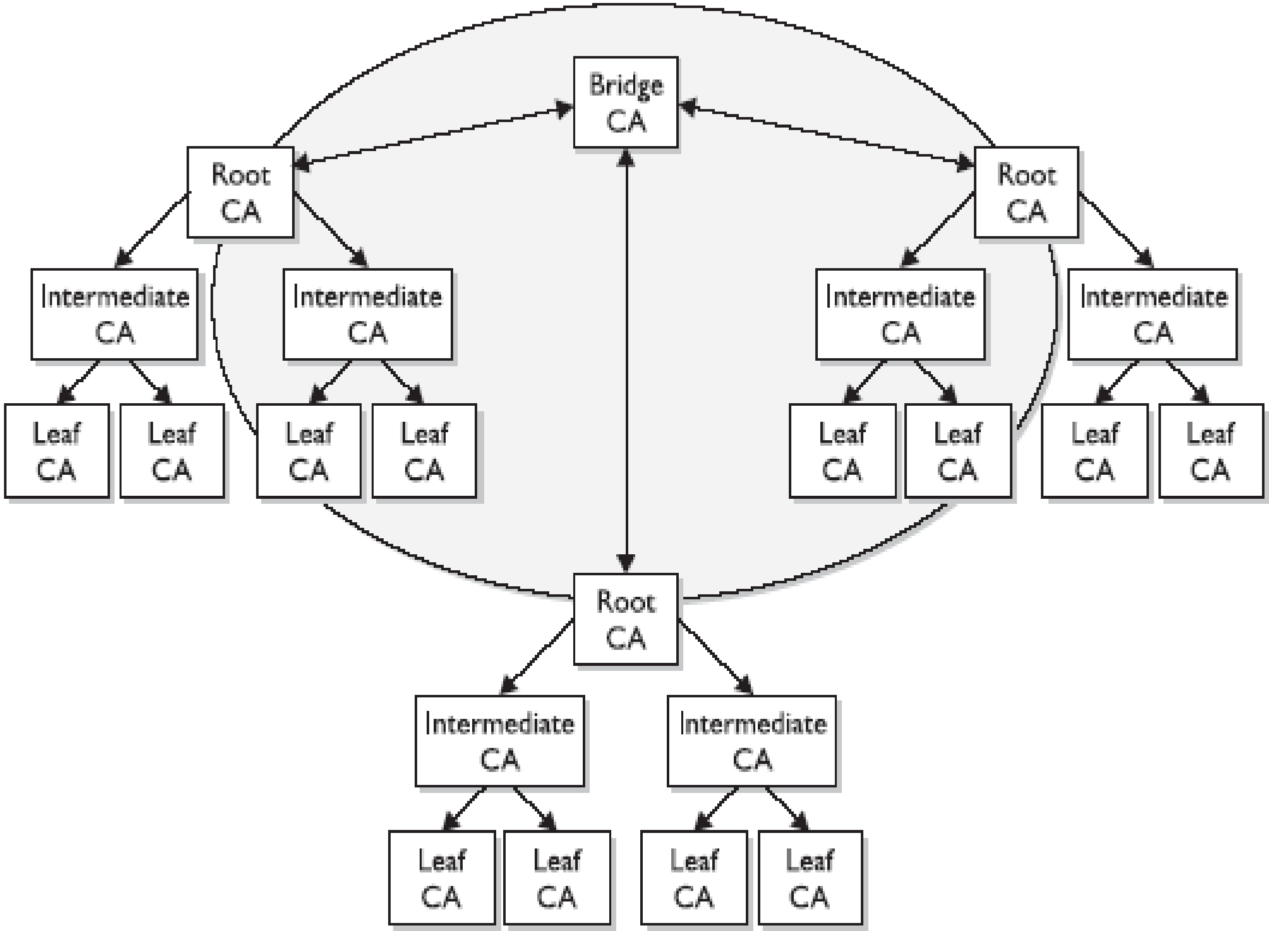
Figure 14-6 Hierarchical trust

Oligarchy

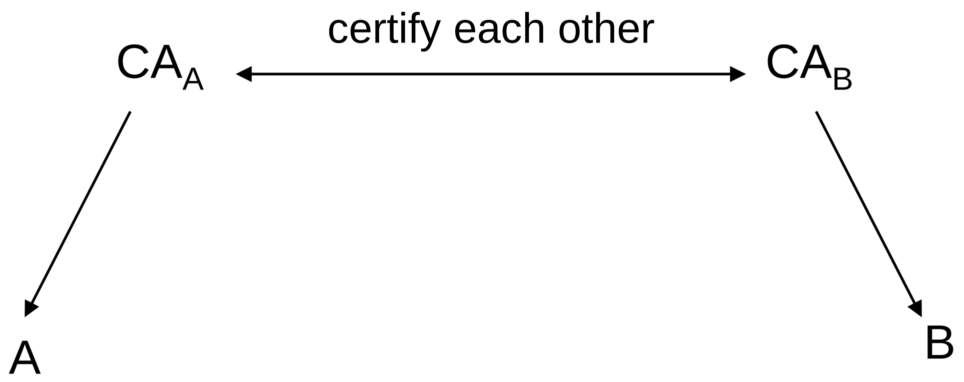
- ❑ **Many root CAs exists trusted by verifiers**
  - ➢ Verisign, Equifax, Entrust, CyberTrust, Identrus, …
  - ➢ Root CAs are unrelated (no cross-certification)
- ❑ **The model of web security**
  - ➢ Each root CA's public key exists in browsers
- ❑ **Advantages:**
  - ➢ Solves the problems of single authority (e.g., excessive pricing)
    - ✓ *n* security-sensitive sites instead of one.
- ❑ **Disadvantages:**
  - ➢ Compromise of any one compromises the whole system
  - ➢ users can easily be tricked into trusting fake CAs. (depending on implementation)

Cross-Trust on Oligarchy

- **How do we establish trust between different CAs?**
  - Bridge CAs can be used but they are rare



Example



- A, to authenticate the public key of B
  - verifies B’s certificate issued by CA_B,
  - verifies CA_B’s cert. issued by CA_A,
- B does vice versa to authenticate A’s public key

Anarchy

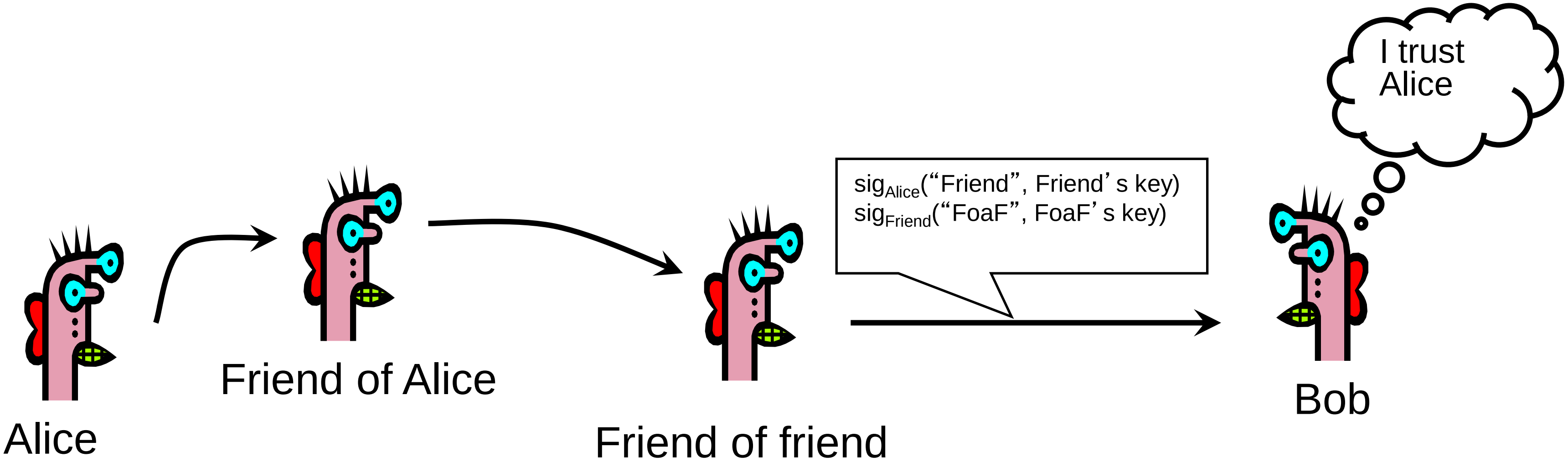
❑ Each user decides whom to trust & how to authenticate their public keys

❑ Certificates issued by arbitrary parties can be stored in public databases, which can be searched to find a path of trust to a desired party

❑ Works well for informal, non-sensitive applications (e.g., PGP)

Example: Pretty Good Privacy

- **Instead of a single root certificate authority, each person has a set of keys they “trust”**
  - If public-key certificate is signed by one of the “trusted” keys, the public key contained in it will be deemed valid
- **Trust can be transitive**
  - Can use certified keys for further certification



I trust Alice

sig$_{Alice}$(“Friend”, Friend’s key)
sig$_{Friend}$(“FoaF”, FoaF’s key)

Alice

Friend of Alice

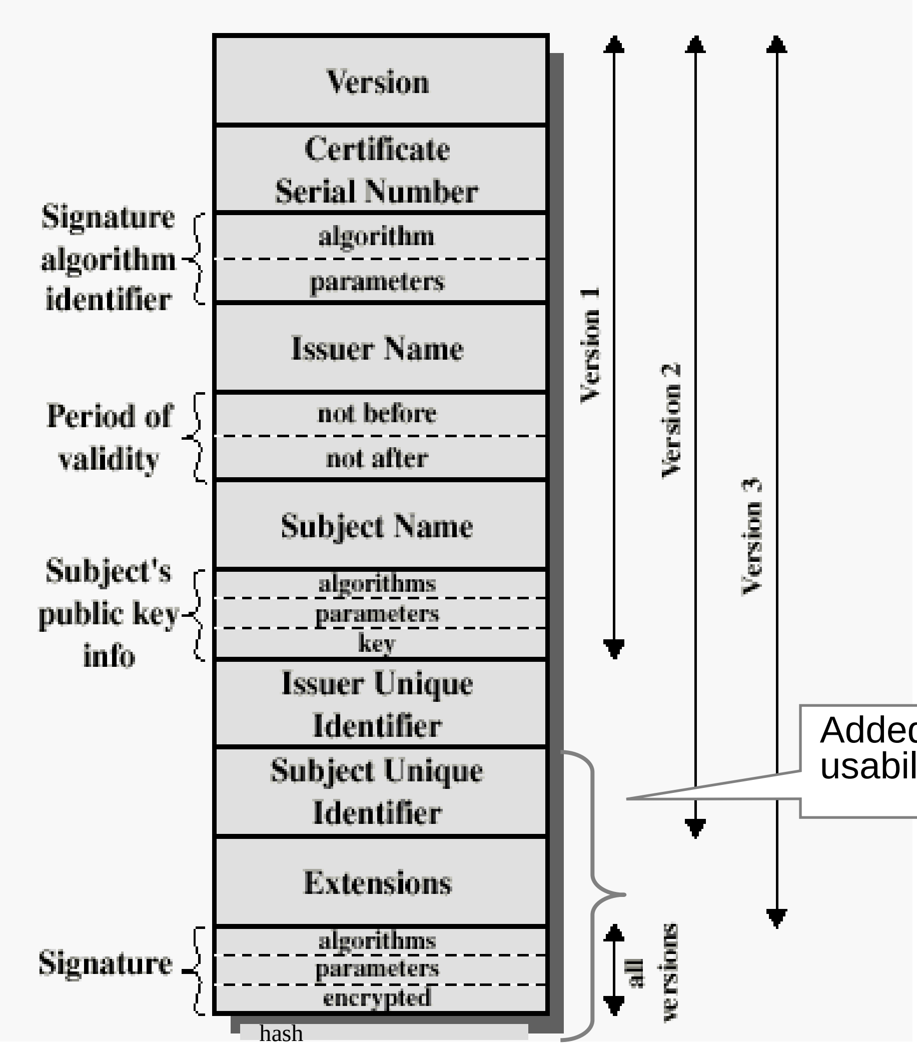
Friend of friend

Bob

X.509 Certificates

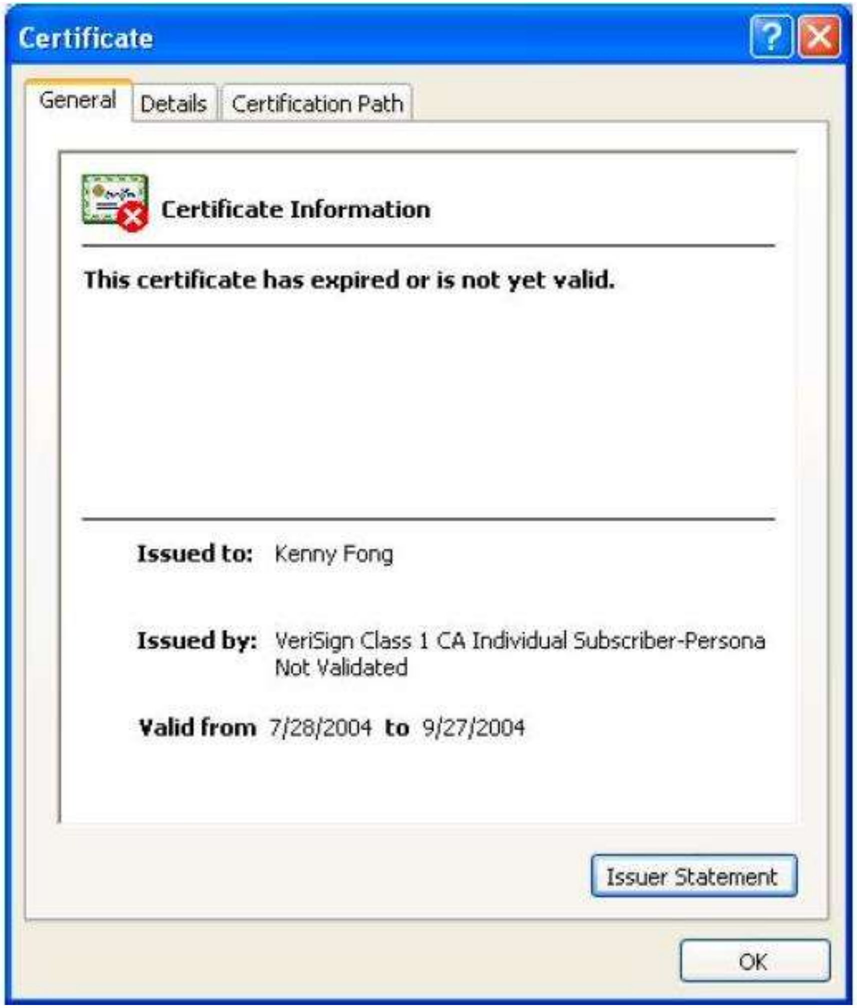
- **Common standard for certificate format**
- **PKIX: Internet standard for X.509-based PKI**
- **Fields (X.509 v3):**
  - version
  - serial number
  - signature algorithm identifier
  - issuer
  - validity period
  - subject
  - subject public key information
  - signature
  - standard extensions (key usage limitation, etc.)
  - other extensions (application & CA specific)

X.509 Certificate



Added in X.509 versions 2 and 3 to address usability and security problems

Sample Personal Certificate



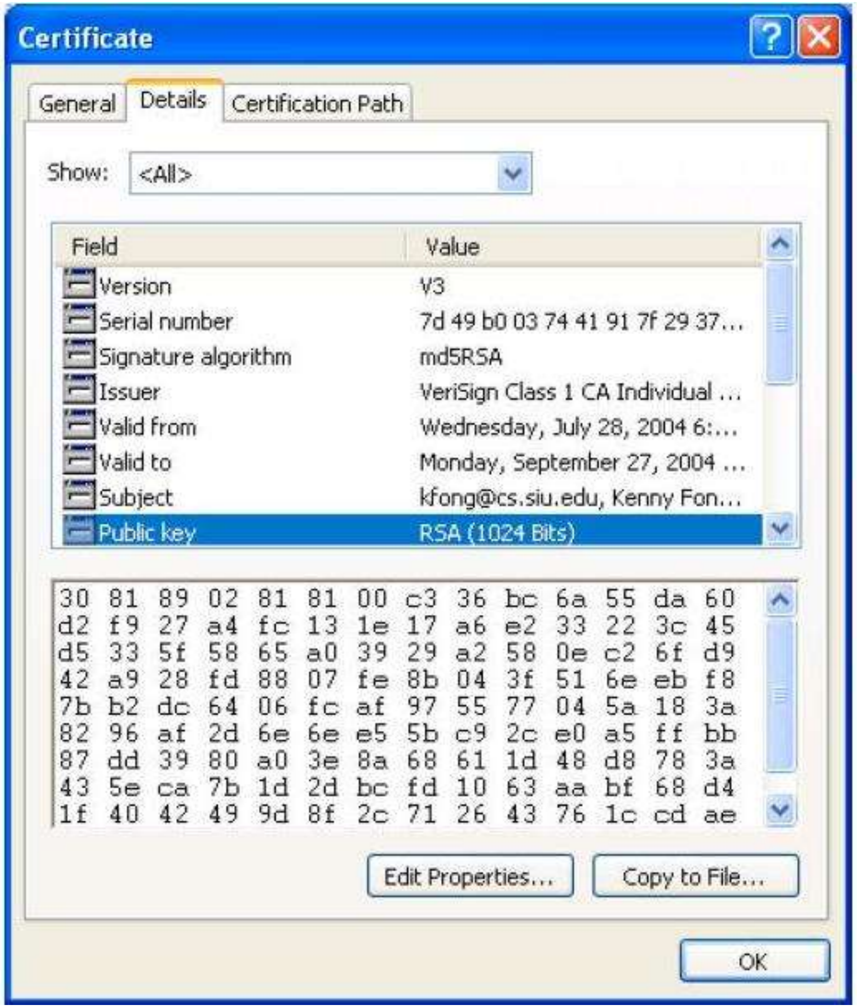
Certificate Information

This certificate has expired or is not yet valid.

Issued to: Kenny Fong

Issued by: VeriSign Class 1 CA Individual Subscriber-Persona Not Validated

Valid from 7/28/2004 **to** 9/27/2004



Show: <All>

Field	Value
Version	V3
Serial number	7d 49 b0 03 74 41 91 7f 29 37...
Signature algorithm	md5RSA
Issuer	VeriSign Class 1 CA Individual ...
Valid from	Wednesday, July 28, 2004 6:...
Valid to	Monday, September 27, 2004 ...
Subject	kfong@cs.siu.edu, Kenny Fon...
Public key	RSA (1024 Bits)

```
30 81 89 02 81 81 00 c3 36 bc 6a 55 da 60
d2 f9 27 a4 fc 13 1e 17 a6 e2 33 22 3c 45
d5 33 5f 58 65 a0 39 29 a2 58 0e c2 6f d9
42 a9 28 fd 88 07 fe 8b 04 3f 51 6e eb f8
7b b2 dc 64 06 fc af 97 55 77 04 5a 18 3a
82 96 af 2d 6e 6e e5 5b c9 2c e0 a5 ff bb
87 dd 39 80 a0 3e 8a 68 61 1d 48 d8 78 3a
43 5e ca 7b 1d 2d bc fd 10 63 aa bf 68 d4
1f 40 42 49 9d 8f 2c 71 26 43 76 1c cd ae
```

Certificate Classes

- ❑ **Different type of certs available, the higher the class the more id required**
- ❑ **Class 1:**
  - ➢ Usage: Encrypting and digitally signing email messages
  - ➢ Identity Checks: Automated enrollment (e.g., entering your name and email in a web form).
- ❑ **Class 2:**
  - ➢ Usage: Software Signing
  - ➢ Identity Checks: address, company information
- ❑ **Class 3:**
  - ➢ Usage: Setting up new CAs
  - ➢ Identity Checks: Face-to-face meeting

Certificate Repositories

- ❑ **Once the certificate is registered, identity proven, and a key pair generated, they are placed in a public repository.**
- ❑ **All of the certificates can be in one, large distributed database (LDAP)**
- ❑ **Each CA can maintain its own repository and have a means of querying the other repositories for information for its users**
- ❑ **Business communities and governments are starting the process of creating their CAs**
  - ➢ They are linking them by signing or cross-certifying and publishing all of their information in business-class repositories.

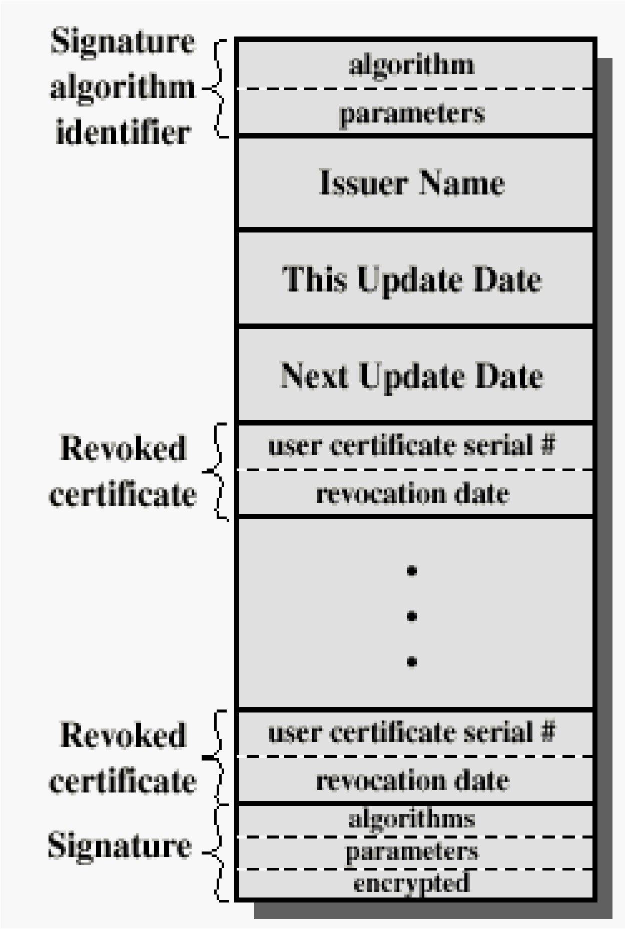
Certificate Revocation

- Revocation is very important
- Many valid reasons to revoke a certificate
  - Private key corresponding to the certified public key has been compromised
  - User stopped paying his certification fee to this CA and CA no longer wishes to certify him
  - CA’s certificate has been compromised!
- Expiration is a form of revocation, too
  - But it is not considered a reason to revoke the certificate
    - ✓ Certificate becomes invalid when it expires
    - ✓ Carries no threat

Certificate Revocation Mechanisms

- **Online Certificate Status Protocol (OCSP)**
  - When a certificate is presented, recipient goes to a special online service (OCSP Server of the CA) to verify whether it is still valid
    - ✓ Like a merchant dialing up the credit card processor
- **Certificate revocation list (CRL)**
  - CA periodically issues a signed list of revoked certificates
    - ✓ Credit card companies used to issue thick books of canceled credit card numbers
  - Can issue a "delta CRL" containing only updates
  - Or local cached CRLs
- **Does revocation protect against forged certs?**
  - If the certificate is known to be forged, yes

X.509 Certificate Revocation List



Because certificate serial numbers must be unique within each CA, this is enough to identify the certificate