Lecture 12

Public Key Certification and Revocation

1

1

Certificates

- Kohnfelder (BS Thesis, MIT, 1978) proposed "certificates" as yet another public-key distribution method
- Certificate = **explicit *binding* between a public key and its owner's (unique) name**
- Must be issued (and signed) by a recognized trusted Certificate Authority (CA)
- Issuance is done **off-line**

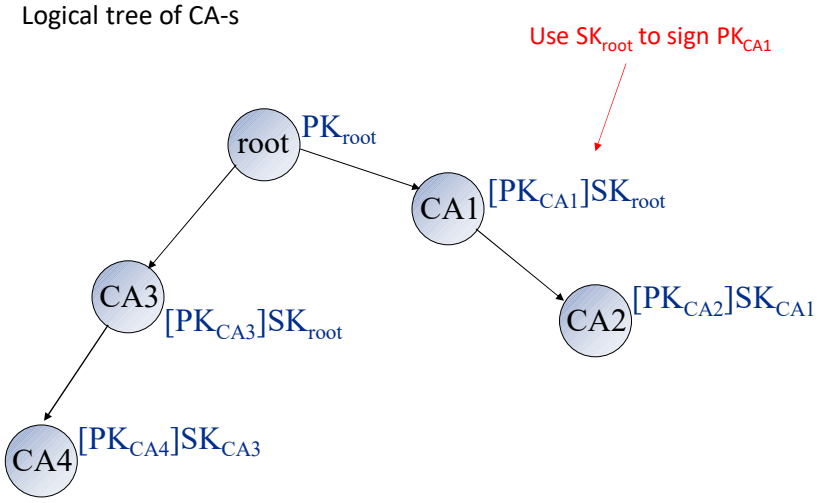
Cannot let root CA to sign all the certifcates in the world --- not scalable

2

2

Certification Tree / Hierarchy

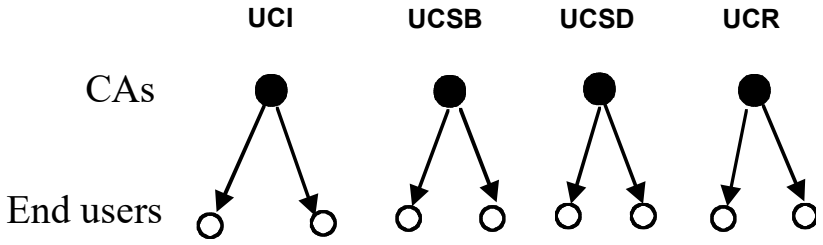
Logical tree of CA-s



3

3

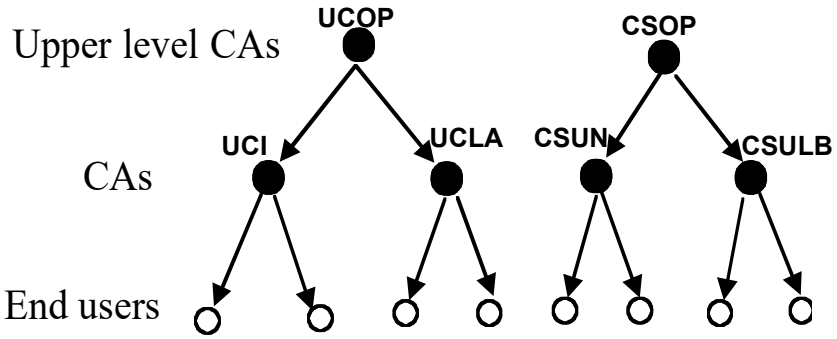
Hierarchical Public Key Infrastructure (PKI) Example



4

4

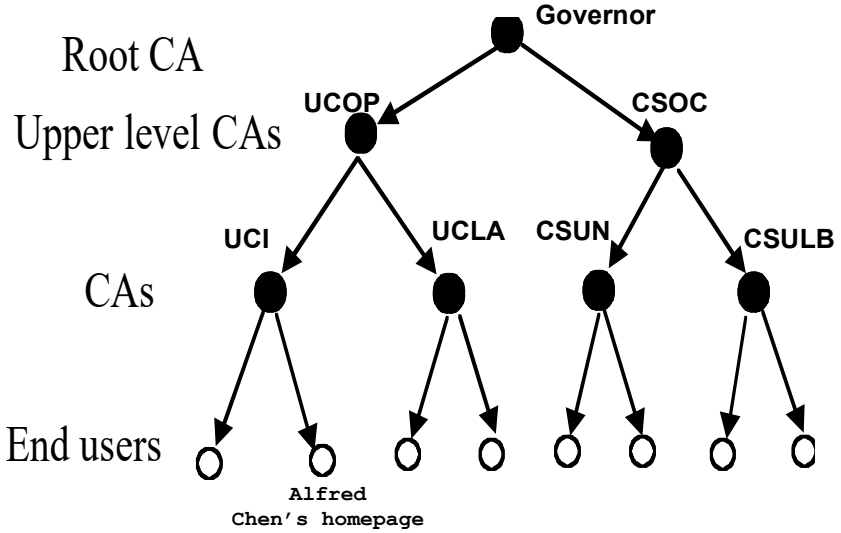
Hierarchical PKI Example



5

5

Hierarchical PKI Example

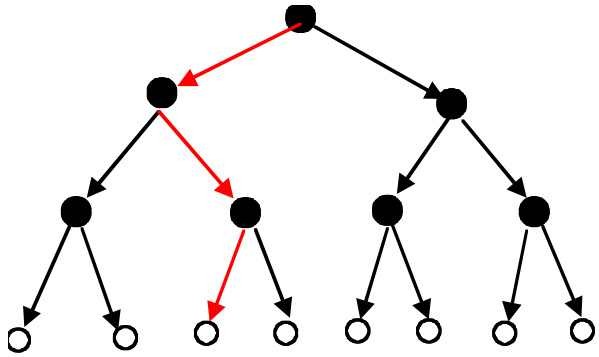


6

6

Certificate Paths

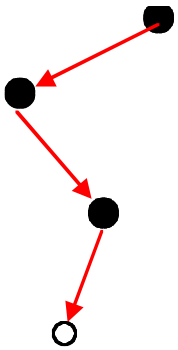
Derived from PKI



7

7

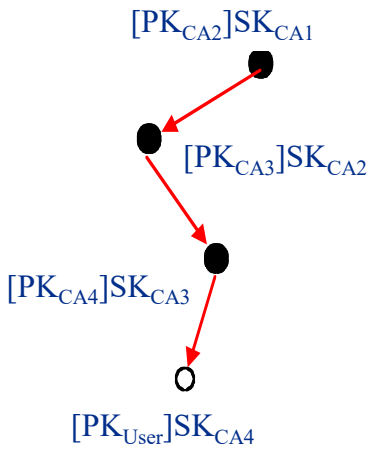
Certificate Paths



8

8

Certificate Paths



- Verifier must know public key of the first CA (usually the root CA)
- Other public keys are 'discovered' one by one
- All CAs on the path must be (implicitly) trusted by the verifier

9

9

X.509 Standard

- X.509v3 is the current version
  - ITU standard
  - ISO 9495-2 is the equivalent ISO standard
- Defines certificate format, not PKI
- **End users cannot be CAs**

10

10

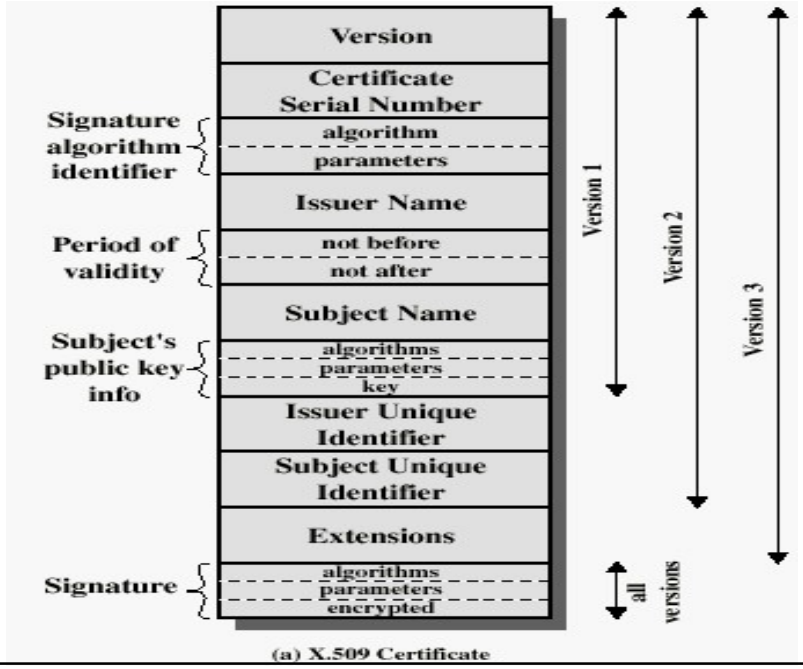
X.509 Certificate Format

- version
- serial number
- signature algorithm ID
- issuer name(X.500 Distinguished Name)
- validity period
- subject(user) name (X.500 Distinguished Name)
- subject public key information
- issuer unique identifier (version 2 and 3 only)
- subject unique identifier (version 2 and 3 only)
- extensions (version 3 only), e.g., revocation info
- signature on the above fields

11

11

X.509 Certificate Format



(a) X.509 Certificate

12

12

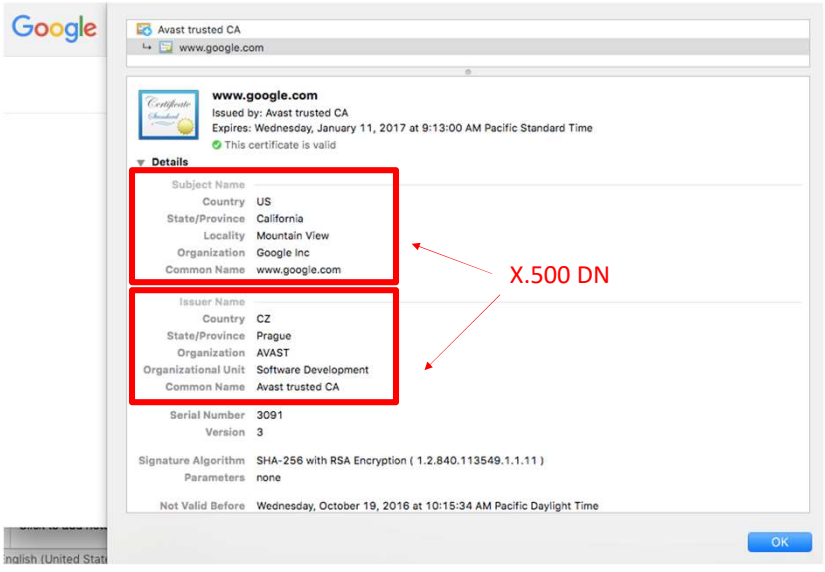
A Sample X.509v3 Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            10:e6:fc:62:b7:41:8a:d5:00:5e:45:b6
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Organization Validation CA - SHA256 - G2
        Validity
            Not Before: Nov 21 08:00:00 2016 GMT
            Not After : Nov 22 07:59:59 2017 GMT
        Subject: C=US, ST=California, L=San Francisco, O=Wikimedia Foundation, Inc., CN=*.wikipedia.org
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
            pub:
                    00:c9:22:69:31:8a:d6:6c:ea:da:c3:7f:2c:ac:a5:
                    af:c0:02:ea:81:cb:65:b9:fd:0c:6d:46:5b:c9:1e:
                    9d:3b:ef
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Agreement
            Authority Information Access:
                CA Issuers - URI:http://secure.globalsign.com/cacert/gsorganizationvalsha2g2r1.crt
                OCSP - URI:http://ocsp2.globalsign.com/gsorganizationvalsha2g2
            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.4146.1.20
                  CPS: https://www.globalsign.com/repository/
                Policy: 2.23.140.1.2.2
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:http://crl.globalsign.com/gs/gsorganizationvalsha2g2.crl
            X509v3 Subject Alternative Name:
                DNS:*.wikipedia.org, DNS:*.m.mediawiki.org, DNS:*.m.wikibooks.org, DNS:*.m.wikidata.org, DNS:*.m.wikimedia.org, DNS:*.m.wikimediafoundation.org, DNS:*.m.wikinews.org,
DNS:*.m.wikipedia.org, DNS:*.m.wikiquote.org, DNS:*.m.wikisource.org, DNS:*.m.wikiversity.org, DNS:*.m.wikivoyage.org, DNS:*.m.wiktionary.org, DNS:*.mediawiki.org,
DNS:*.planet.wikimedia.org, DNS:*.wikibooks.org, DNS:*.wikidata.org, DNS:*.wikimedia.org, DNS:*.wikimediafoundation.org, DNS:*.wikinews.org, DNS:*.wikiquote.org, DNS:*.wikisource.org,
DNS:*.wikiversity.org, DNS:*.wikivoyage.org, DNS:*.wiktionary.org, DNS:*.wmfusercontent.org, DNS:*.zero.wikipedia.org, DNS:mediawiki.org, DNS:w.wiki, DNS:wikibooks.org,
DNS:wikidata.org, DNS:wikimedia.org, DNS:wikimediafoundation.org, DNS:wikinews.org, DNS:wikiquote.org, DNS:wikisource.org, DNS:wikiversity.org, DNS:wikivoyage.org, DNS:wiktionary.org,
DNS:wmfusercontent.org, DNS:wikipedia.org
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Subject Key Identifier:
                28:2A:26:2A:57:8B:3B:CE:B4:D6:AB:54:EF:D7:38:21:2C:49:5C:36
            X509v3 Authority Key Identifier:
                keyid:96:DE:61:F1:BD:1C:16:29:53:1C:C0:CC:7D:3B:83:00:40:E6:1A:7C

    Signature Algorithm: sha256WithRSAEncryption
         8b:c3:ed:d1:9d:39:6f:af:40:72:bd:1e:18:5e:30:54:23:35:
         ...
```

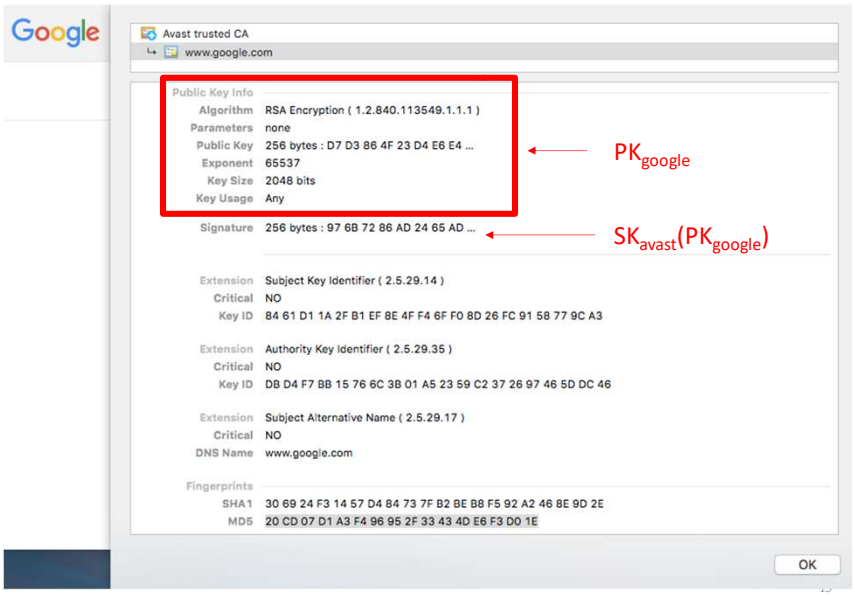
13

A Sample Certificates in Practice (1/3)



14

A Sample Certificates in Practice (2/3)



15

A Sample Certificates in Practice (3/3)

```
-----BEGIN CERTIFICATE-----
MIIDTzCCAvmgAwIBAgIBATANBgkqhkiG9w0BAQQFADBcMSEwHwYDVQQKExhFdXJv
cGVhbiBJQ0UtVEVMIHByb2plY3QxIzAhBgNVBAsTGlYzLUNlcnRpZmljYXRpb24g
QXV0aG9yaXR5MRIwEAYDVQQHEwlEYXJtc3RhZHQwHhcNOTcwNDAyMTczNTU5WhcN
OTgwNDAyMTczNTU5WjBrMSEwHwYDVQQKExhFdXJvcGVhbiBJQ0UtVEVMIHByb2pl
Y3QxIzAhBgNVBAsTGlYzLUNlcnRpZmljYXRpb24gQXV0aG9yaXR5MRIwEAYDVQQH
EwlEYXJtc3RhZHQxDTALBgNVBAMTBFVTRVIwWTAKBgRVCAEBAgICAANLADBIAkEA
qKhTY0kbk8PDC2yIEVXefmri+VKg3GklxMi/VeExqM7kqSmFmYoVmt72L+G0UF9e
BHWm9HbcPA453Dq+PqRhiwIDAQABo4IBmDCCAZQwHwYDVR0jBBgwFoAUfnLy+DqG
nEKINDRmdcPU/NGiETMwHQYDVR0OBBYEFJfc4B8gjSoRmLUx4Sq/ucIYiMrPMA4G
A1UdDwEB/wQEAwIB8DAcBgNVHSABAf8EEjAQMAYGBCoDBAUwBgYECQgHBjBDBgNV
HREEPDA6gRV1c2VyQGRhcm1zdGFkdC5nbWQuZGWGIWh0dHA6Ly93d3cuZGFybXN0
YWR0LmdtZC5kZS9+dXNlcjCBsQYDVR0SBIGpMIGmgQxnbWRjYUBnbWQuZGWGEWh0
dHA6Ly93d3cuZ21kLmRlghdzYXR1cm4uZGFybXN0YWR0LmdtZC5kZaRcMSEwHwYD
VQQKExhFdXJvcGVhbiBJQ0UtVEVMIHByb2plY3QxIzAhBgNVBAsTGlYzLUNlcnRp
ZmljYXRpb24gQXV0aG9yaXR5MRIwEAYDVQQHEwlEYXJtc3RhZHSHDDE0MS4xMi42
Mi4yNjAMBgNVHRMBAf8EAjAAMB0GA1UdHwQWMBQwEqAQoA6BDGdtZGNhQGdtZC5k
ZTANBgkqhkiG9w0BAQQFAANBAGkM4ben8tj76GnAE803rSEGIk3oxtvxBAu34LPW
DIEDzsNqPsfnJCSkkmTCg4MGQlMObwkehJr3b2OblJmD1qQ=
-----END CERTIFICATE-----
```

16

16

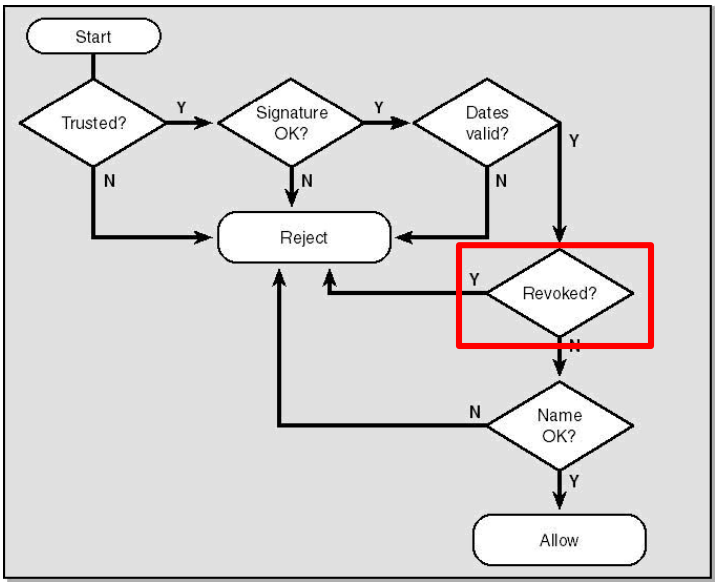
Certificates in Practice

- X.509 certificate format is defined in Abstract Syntax Notation 1 (ASN.1)
- ASN.1 structure is encoded using the Distinguished Encoding Rules (DER)
- A DER-encoded binary string is typically base-64 encoded to get an ASCII representation (previous slide)

17

17

Verifying a Certificate (assuming Common CA)



18

18

Certificate Revocation Scenarios

After a certificate is issued, lots of things can happen before it expires
What if:

- Bob's CA goes out of control?
- Bob left the company?
- Bob forgets his private key?
- Someone steals Bob's private key?
- Bob willingly discloses his private key?

Eve can decrypt/sign while Bob's certificate is still valid!

- Bob reports key loss to CA (or CA finds out somehow)
- CA issues a **Certificate Revocation List (CRL)**
  - Distributed/published in public announcements/databases
- When verifying Bob's signature or encrypting a message for Bob, Alice first checks if Bob's certificate is still valid!
- IMPORTANT: what about signatures Eve generated before Bob realized his key is lost?

19

19

Certificate is a capability

- Certificate revocation needs to occur when:
  - certificate holder key compromise/loss
  - CA key compromise
  - end of contract (e.g., certificates for employees)
- Certificate Revocation List (CRL) lists certificates that are not yet naturally expired but revoked
- CRL should be reissued periodically, even there is no new revocation activity! WHY?

20

20

Requirements for Revocation

- Timeliness
  - Before using a certificate, must check most recent revocation status
- Efficiency
  - Computation
  - Bandwidth and Storage
  - Availability
- Security

21

21

Types of Revocation

- Explicit
  - Only revoked certificates are periodically announced
  - Alice's certificate is not listed among the revoked ➔ Alice is not revoked
  - Need to distribute/publish revocation info
- Implicit
  - Each certificate is frequently/periodically re-issued
  - Alice has a current valid certificate ➔ Alice is not revoked
  - No need to distribute/publish revocation info

22

22

Revocation Methods

Explicit:

- CRL - Certificate Revocation List
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- OCSP – On-line Certificate Status Protocol

Implicit:

- CRS - Certificate Revocation System

23

23

Revocation Methods

Explicit:

- **CRL - Certificate Revocation List**
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- OCSP – On-line Certificate Status Protocol

Implicit:

- CRS - Certificate Revocation System

24

24

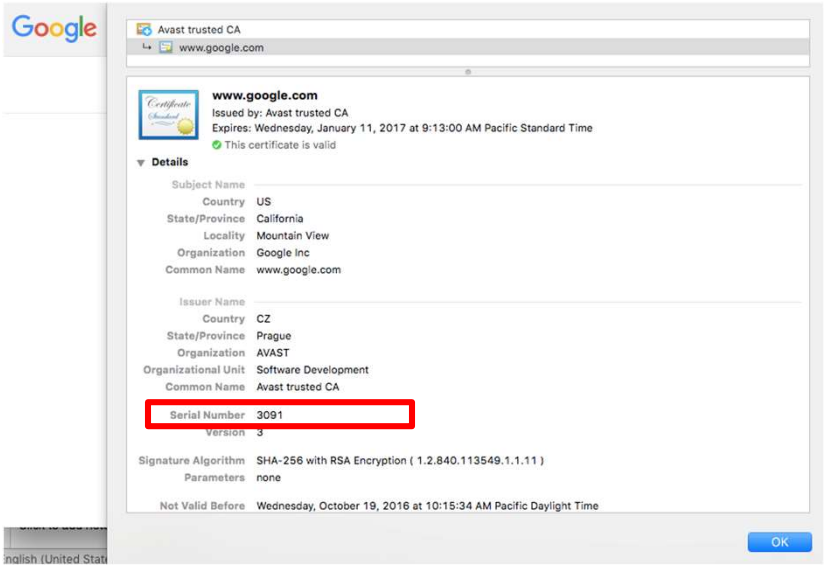
Certificate Revocation List (CRL)

- Off-line mechanism
- CRL = list of revoked certificates (e.g., SNs) signed by a revocation authority (RA)

25

25

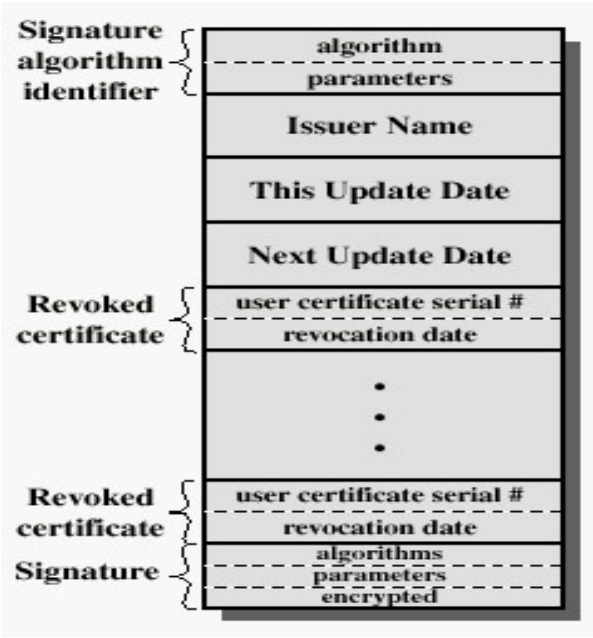
A Sample Certificates in Practice (1/3)



26

26

X.509 CRL Format



27

27

Certificate Revocation List (CRL)

- Off-line mechanism
- CRL = list of revoked certificates (e.g., SNs) signed by a revocation authority (RA)
- RA not always CA that issued the revoked PKC
- Periodically issued: daily, weekly, monthly, etc.

28

28

Pros & Cons of CRLs

- Pros
  - Simple
  - Does not need secure channels for CRL distribution
- Cons
  - Timeliness: "window of vulnerability"
  - CRLs grow and can become huge
  - How to distribute CRLs reliably?

29

29

Pros & Cons of CRLs

- Pros
  - Simple
  - Does not need secure channels for CRL distribution
- Cons
  - Timeliness: "window of vulnerability"
  - **CRLs grow and can become huge**
  - How to distribute CRLs reliably?

30

30

Certificate Revocation Tree (CRT)

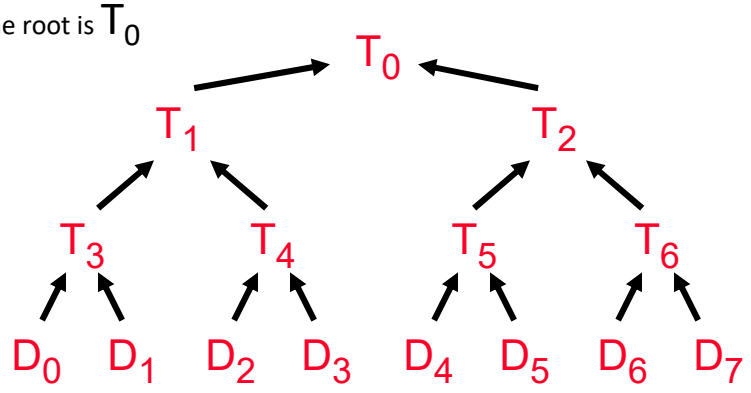
- Proposed in 1998 by P. Kocher
- Based on so-called hash trees
  - Hash trees first proposed by R. Merkle in another context in 1979 (one-time signatures)

31

31

Merkle Hash Tree Example

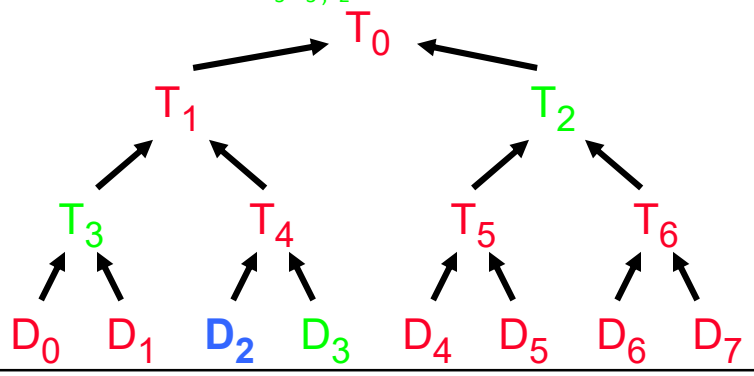
- Need to authenticate a sequence of values: $S = \{ D_0 , D_1 , \ldots, D_N \}$
- Construct binary tree over data values
- Arrows represent hashing, e.g., $T_4 = H (D_2 \,||\, D_3)$
- The root is T_0



32

Merkle Hash Trees: II

- Verifier knows T_0
- How can verifier authenticate that a tree leaf D_i is in the sequence ***s***?
- Solution: re-compute T_0 using D_i
  - Due to **strong collision resistance**, as long as one can construct T_0 with D_i in a sequence ***s'*** = {…, D_i, …}, this sequence ***s'*** must be the original sequence ***s***
- Example: to authenticate that D_2 is in the sequence
  - Naïve idea: send {D_0 , D_1 , D_2, …, D_7} → someone can re-compute root, and verify if equals T_0
  - Better idea: send D_2 + **co-path**=[D_3 ,T_3 , T_2] → verify if T_0 == H(H(T_3 || H(D_2 || D_3)) || T_2)



33

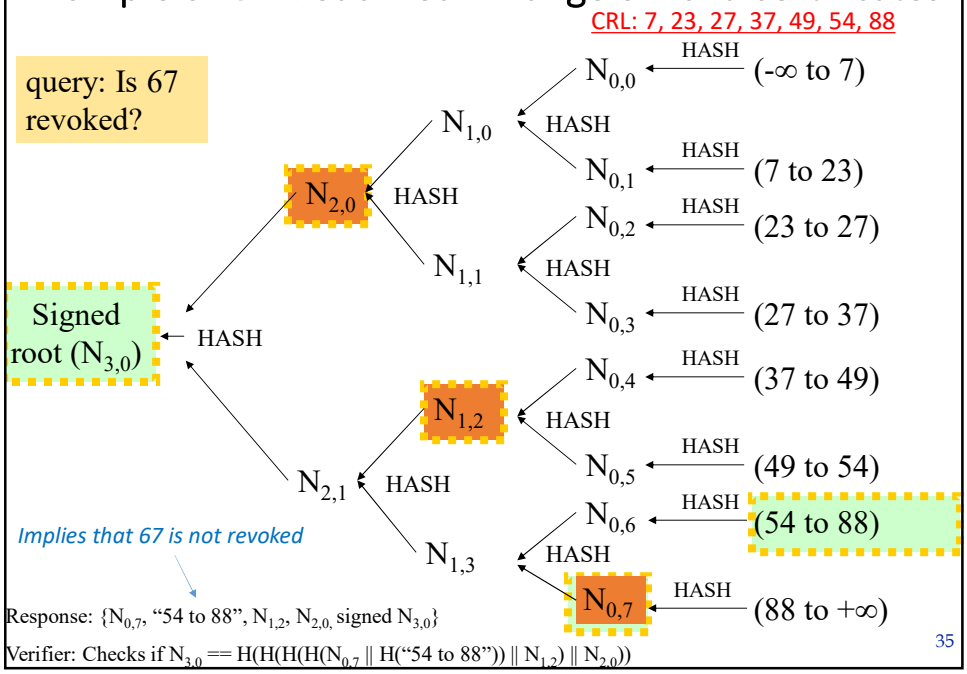
CRT Contd.

- Express ranges of SN of PKC's as tree leaf labels:
  - E.g., (5—12) means: 5 and 12 are revoked, those larger than 5 and less than 12 are okay
  - Place the hash of the range in the leaf
- Response includes (1) the corresponding tree leaf, (2) the necessary hash values along the path to the root (called **co-path**), (3) the signed root
- The CA periodically updates the structure and distributes to untrusted servers called Confirmation Issuers

34

34

Example of CRT: each leaf = range of valid certificates



35

Characteristics of CRT

- Each response (leaf + co-path) represents a proof
- Length of proof: O(log n)
  - Much shorter than CRL which is O(n)
    - In the example last page: CRL – length: 7, CRT – length: log(7)
  - Where n is # of revoked certificates
- Only one "real" signature for the whole tree – over the root

36

36

Revocation Methods

Explicit:

- CRL - Certificate Revocation List
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- **OCSP – On-line Certificate Status Protocol**

Implicit:

- CRS - Certificate Revocation System

37

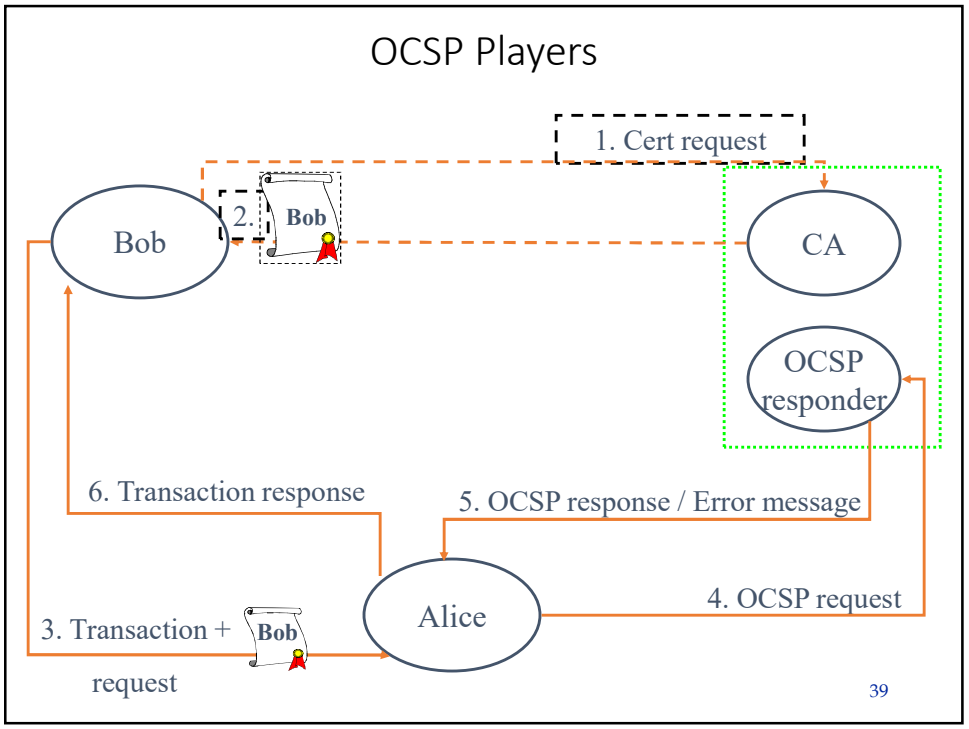
37

Explicit Revocation: OCSP

- OCSP = On-line Certificate Status Protocol (RFC 2560) - June 1999
- Conveys instantaneous status of a PKC

38

38



39

Explicit Revocation: OCSP

- OCSP = On-line Certificate Status Protocol (RFC 2560) - June 1999
- Conveys instantaneous status of a PKC
- Used in place of or, as a supplement to, checking CRLs
- Especially suitable for sensitive, volatile settings, e.g., stock trades, electronic funds transfer, military

40

40

Revocation Methods

Explicit:

- CRL - Certificate Revocation List
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- OCSP – On-line Certificate Status Protocol

Implicit:

- CRS - Certificate Revocation System

41

41

Revocation Methods

Explicit:

- CRL - Certificate Revocation List
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- OCSP – On-line Certificate Status Protocol

Implicit:

- CRS - Certificate Revocation System

42

42

Implicit Revocation: Certificate Revocation System (CRS)

- Proposed by Micali (1996)
- Aims to improve CRL communication costs
- Basic idea: CA periodically refreshes valid certificates
- Uses off-line/on-line signature scheme to reduce update cost

43

One-Way Hash Chains

- Versatile cryptographic primitive
- Construction:
  1. Pick random number $\mathbf{Y_N}$ and a public hash function **H()**
  2. Compute N values $\mathbf{Y_{N-1}}$,…,$\mathbf{Y_0}$ such that $\mathbf{Y_{i-1} = H(Y_i)}$
  3. Secret **ROOT=$\mathbf{Y_N}$** , public **ANCHOR=$\mathbf{Y_0}$**



- Properties:
  - Use in reverse order of construction: $\mathbf{Y_0}$, $\mathbf{Y_1}$, …, $\mathbf{Y_N}$
  - Hard to compute $\mathbf{Y_i}$ from $\mathbf{Y_j}$ (if j<i), easy to compute $\mathbf{Y_j}$ from $\mathbf{Y_i}$
  - For example: easy to compute $\mathbf{Y_1}$ from $\mathbf{Y_2}$ since $\mathbf{Y_1=H(Y_2)}$
  - But, Infeasible to compute $\mathbf{Y_2}$ from $\mathbf{Y_1}$
- Verifier can efficiently authenticate $\mathbf{Y_j}$ knowing $\mathbf{Y_i}$ (j<i) by verifying whether: $\mathbf{Y_j} = H^{i-j}(\mathbf{Y_i}) = H(H(\ldots H(\mathbf{Y_i})\ldots))$
- This method is robust to missing values

44

CRS: Creation of a Certificate

- Two new parameters included in each PKC: $\mathbf{Y_0}$ and $\mathbf{N_0}$

$$\mathbf{Y_0} = H^{MAX}(\mathbf{Y_{MAX}})$$

CHAIN ANCHOR

CHAIN ROOT

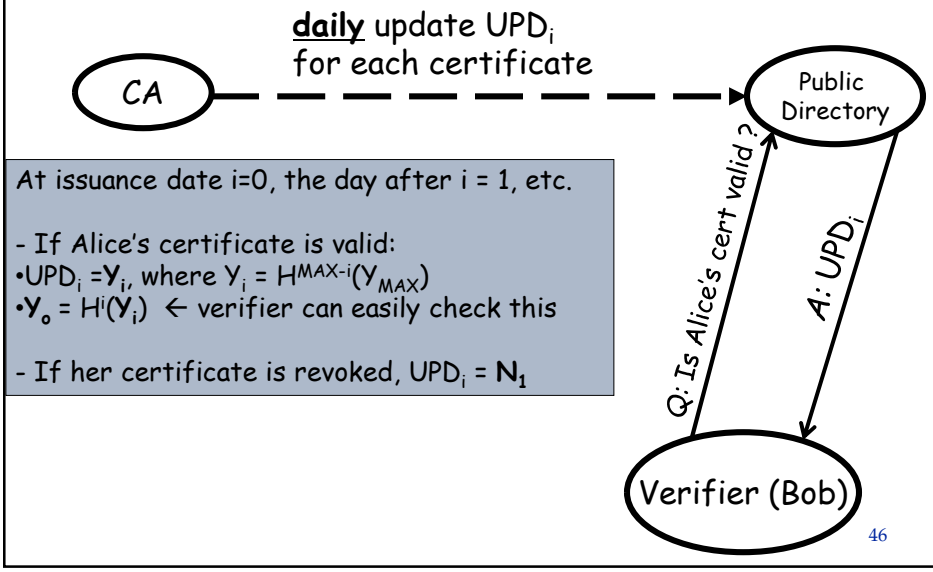
$$\mathbf{N_0} = H(\mathbf{N_1})$$

- $[\mathbf{Y_{MAX}}, \mathbf{N_1}]$ -- per-PKC secrets stored by CA
- H() -- public one-way function, e.g., SHA-2

45

45

CRS Example: Certificate issued for a year, refreshed daily



46

46

Revocation Methods

Explicit:

- CRL - Certificate Revocation List
  - Sources: CRL-DP, indirect CRL, dynamic CRL-DP
  - Delta-CRL, windowed CRL, etc.
  - Certificate Revocation Tree (CRT) and other Authenticated Data Structures
- OCSP – On-line Certificate Status Protocol

Implicit:

- CRS - Certificate Revocation System

47