

# PKI

## Part 4: Certificate Revocation and Short-lived Certificates

Kjell Jørgen Hole  
NoWires Research Group  
Department of Informatics  
University of Bergen

last changed October 6, 2008

## Outline

- Introduction to certificate revocation
- Certificate Revocation Lists (**CRLs**)
  - problems with CRLs
- On-line query mechanisms
  - Online Certificate Status Protocol (**OCSP**)
- Short-lived certificates

# Certificate revocation

- There are reasons why a certificate should no longer be considered valid, even when it has not yet expired
  - (suspected) private-key compromise
  - change in job status
  - change of (last) name
  - person gets fired

3

# CRLs

- A CRL is a signed data structure containing serial numbers of revoked certificates
- The CA is responsible for maintaining and signing the CRL
- End entities download a CRL to determine whether a certificate is revoked
- A CRL can be cached to enhance performance
  - possible to verify certificates while off-line

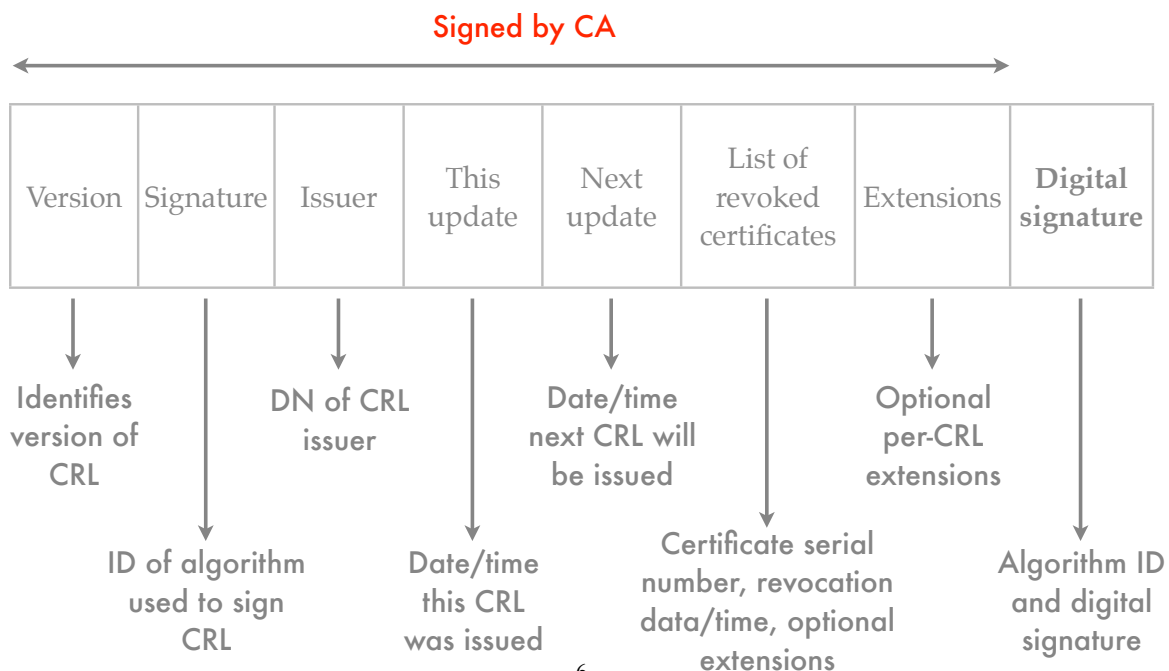
4

# CRL versions

- Two versions of a CRL is defined:
- Version 1 was defined in the original X.509 specifications
  - Version 1 is flawed and should not be used
  - Version 2 is depicted on the next slide

5

## Version 2 CRL structure



6

# Periodic CRL updating

- A CRL is a periodic publication mechanism
- The frequency with which the CA updates the CRL is an important consideration:
  - the acceptable delay between discovering that a certificate should be revoked and updating the CRL vary for different systems
  - the acceptable delay should be defined in the Certificate Policy

7

# Complete CRL

- A CRL that covers the entire certificate population of a CA is called a **complete** or **full** CRL
- The directory location of a full CRL is implied by the CA's distinguished name
- A complete CRL is easy to implement
  - it works well for a limited number of end entities

8

# The problem with full CRLs

- Full CRLs do not scale well because the lists tend to become large as the number of users grow
- a large CRL may require too much bandwidth when many end entities try to download the list, especially in wireless systems
- storage of large CRLs may also become a problem on thin clients like mobile phones

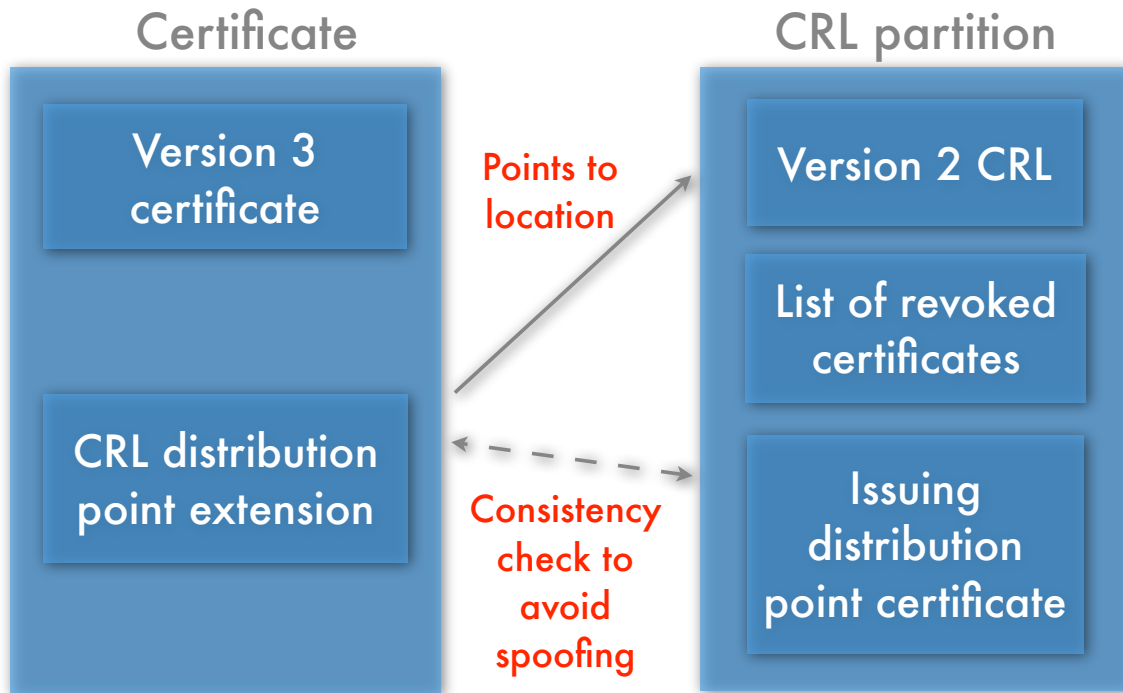
9

# CRL distribution points

- **CRL distribution points** (or partitioned CRLs) post revocation information for a single CA in multiple CRLs
- a certificate contains a CRL distribution point extension that identifies a specific server (DNS name or IP address) for a CRL partition
- the partitioning makes it possible to limit the maximum size of a CRL

10

# Illustration



11

## Distribution point problems

- While a CRL distribution point can identify a specific server for a CRL partition, it cannot change the CRL partitions once they are introduced
- large populations of certificate users may require a more dynamic partitioning scheme

12

# Delta CRLs

- A **delta CRL** only contains changes from a given base CRL
- The combination of the base and delta CRLs constitutes all known revocation information
- It is possible to post **multiple** delta CRLs against the same base CRL
  - each new delta CRL contains the complete list of the revoked certificates from the previous delta CRL

13

# More on delta CRLs

- It is only necessary to retrieve the latest delta CRL
- The delta CRL reduces the size of the revocation information an end entity must retrieve to determine if a certificate is revoked
  - a delta CRL reduces bandwidth requirements
- Necessary to post new base CRL when the delta CRL becomes large

14

# Example

- Enterprise CRL policy:
  - a new base CRL is issued once a week
  - delta CRLs are issued every eight hours
- The larger base CRL only needs to be downloaded and cached once a week

15

# Problems with delta CRLs

- CA must provide both base CRL and delta CRLs
- Certificate validation becomes more complex than in the full CRL case
- A base CRL may be too large for a thin client such as a smart phone

16

# Other CRL schemes

- There exist other more complicated CRL schemes
  - Certification Revocation Trees (CRTs)
- However, simple and straightforward CRL checking is more likely to be implemented than complex, multitiered CRL schemes

17

# On-line query mechanisms

- CRLs were created during a time when clients had intermittent connections to the CA, necessitating off-line processing of CRLs
- We now assume that the end entities have on-line connections to the CA whenever the status of a certificate must be determined
- These mechanisms may be better suited for thin clients in wireless systems

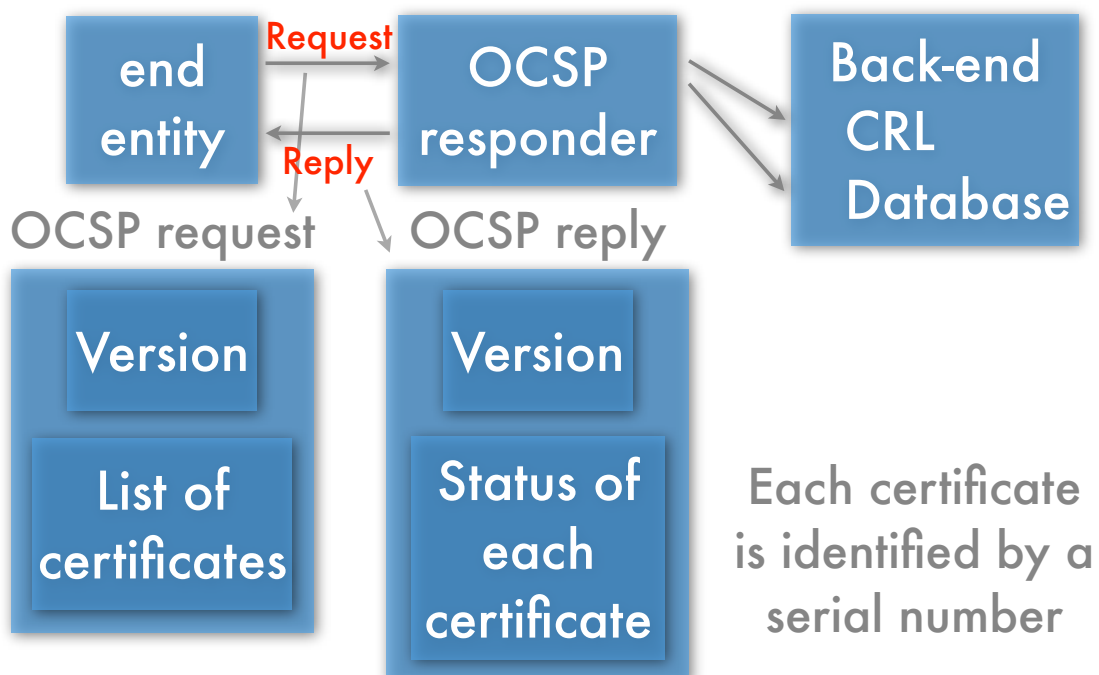
18

# OCSP

- The Online Certificate Status Protocol (**OCSP**) is defined in RFC 2560
- Request-response protocol that obtains revocation information from a trusted server called the **OCSP responder**
- **OCSP request**: list of certificate identifiers
- **OCSP reply**: certificate status is **good**, **revoked**, or **unknown**

19

## OCSP illustration



# More on the OCSP reply

- An OCSP reply must be digitally signed to provide evidence that the reply
  - originated from the OCSP responder
  - has not been altered in transit
- The end entity must obtain a copy of the OCSP responder's public-key certificate to verify a reply

21

# Revocation strategies

- Numerous revocation strategies can be implemented behind the OCSP responder
- the “freshness” of the replies depends on the latency involved in updating the revocation information at the back end

22

# OCSP performance

- The digital signing of the replies is likely to have a negative effect on the performance of the OCSP responder

23

**Case study: PKI with short-lived certificates and thin clients**

24

# Long-lived certificates

- The use of long-lived client certificates is responsible for much of the complexity and cost of PKIs
- in particular, certificate revocation requires costly administration and maintenance
- Long-lived server certificates do not introduce the same problems in systems where the CA is in a well-protected environment, e.g., an Intranet

25

# Advantages of short-lived client certificates

- Short-lived client certificates introduce several advantages:
  - no need for certificate revocation
  - a shorter key can be used (?), resulting in better performance

26

# Useful environments

- Short-lived client certificates are useful in client-server systems where the server side:
  - controls the RA and CA
  - makes the client software
  - has a central data base with customer information
- Examples are enterprise Intranets and Internet banking systems with thin clients

27

# Lifetimes

- The lifetime of a client certificate should be determined by its intended use:
  - certificates for 9-to-5 employees should be valid for 8 to 10 hours
  - end-entity certificates for Internet banking should only be valid for a single session

28

# Client smart cards

- The thin clients can use **smart cards** to store private keys and to generate new key pairs
- An end user may receive a smart card after he or she has provided authentication information to an RA
- **Initially**, the card contains the information:
  - a key pair (but no public-key certificate)
  - root CA's public-key certificate

29

# SSL (1)

- A client and the server exchange public-key certificates during the initialization of an SSL/TLS connection
- When SSL is used together with short-lived client certificates, an end entity must obtain a new certificate before the SSL connection can be initiated

30

## SSL (2)

- The enterprise (or bank) CA has a long-lived certificate signed by a third party CA, e.g. VeriSign
- A client accepts the CA certificate as long as it can be verified using the public key in the root certificate on the smart card
- if the CA is compromised then it is the bank's responsibility to change the CA's key pair

31

## Compromised CA

- CA must:
  1. generate a new key pair
  2. obtain a new public-key certificate from third party CA, e.g., VeriSign
- Clients must ask for new certificates because old certificates are signed with the compromised CA's private key

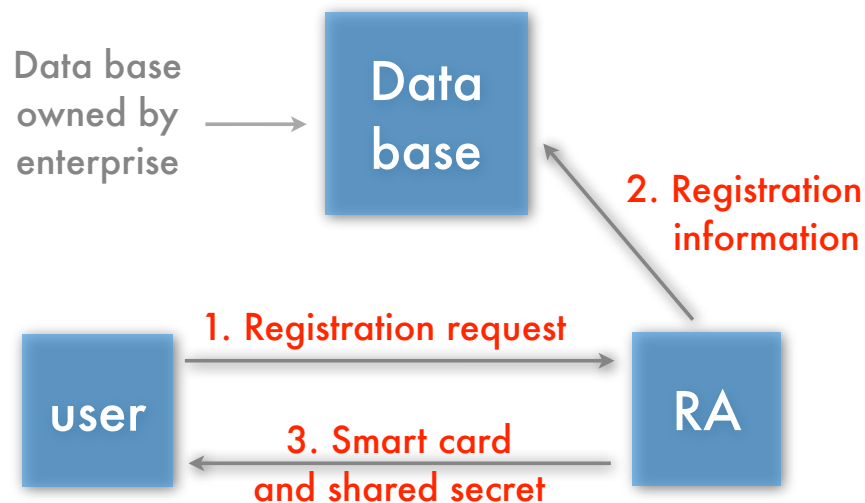
32

# PKI/SSL framework

- To better understand how to use short-lived certificates, we introduce a PKI framework
- The framework consists of:
  1. Thin clients with smart cards
  2. RA (Registration Authority)
  3. CA (Certification Authority)
  4. Central data base containing certificate credentials

33

## User registration



- A new user must register to obtain a certificate

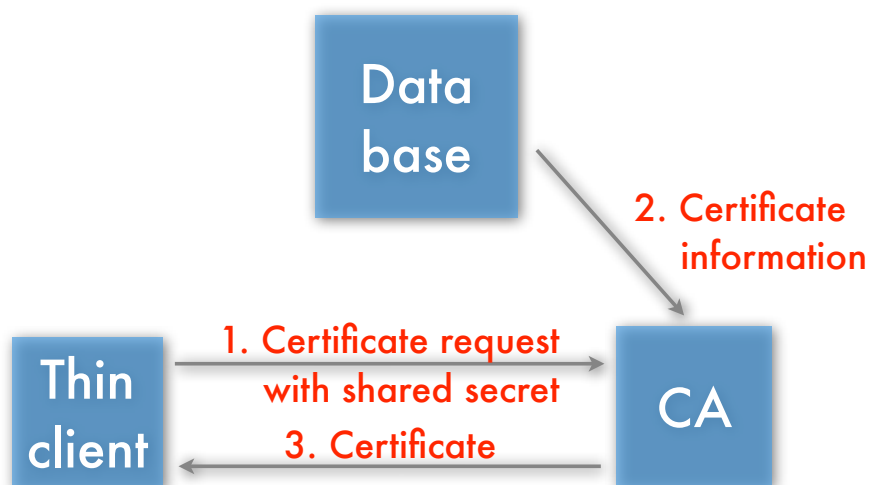
34

# RA

- The user provides the RA with multiple picture IDs to authenticate himself
- The RA then
  1. verifies the registration information
  2. stores DN, public key, and other pertinent information in the central data base

35

## Certification procedure



- The client provides the shared secret to obtain a certificate

36

# CA

1. CA obtains certificate information from the data base, including public key
2. CA verifies that the end entity has the corresponding private key
3. CA generates and sends short-lived client certificate

37

# Comments

- The use of a shared secret to obtain the first certificate limits the degree of obtainable non-repudiation
- It may be necessary to use multiple CAs for this framework to handle a large number of end entities
  - all CAs access the same central data base

38

# Client certificate renewal

- To obtain a new certificate, a client submits a request containing:
  - DN
  - new public key
- This request must be signed with the old private key
- The communication must be encrypted using the server's public key

39

# Disadvantages?

- The client must obtain a new certificate each time the short-lived certificate expires
  - operations may interrupt highly critical functions
  - users may experience a delay while a new certificate is requested
- Many clients may ask for a new certificate at the same time (employees arrive at work)

40

# Key generation on thin clients

- Key generation can be a time consuming process
- Should employ smart cards that generate key pairs in an efficient and secure manner
- fast key generation
- sufficiently random keys
- secure storage of private keys

41

# Key generation ...

- It is important to have a large pool of key pairs
- It is unknown how key generation on the clients will affect the overall performance of the PKI

42

# Trust model (1)

- It is assumed that the RA, CA, and the central data base are placed in a secure environment
- The client OS is vulnerable to attacks and should be trusted as little as possible
- Must trust the clients' smart cards
  - can card API be misused by malicious software?

43

# Trust model (2)

- Client software is likely to be vulnerable
  - possible to change client software using decompilers and code changes?
- The server must validate all inputs from the clients

44

# Sources

- C. Adams and S. Lloyd, *Understanding PKI*, 2nd Edition, Addison Wesley, 2003
- W. Ford and M. S. Baum, *Secure Electronic Commerce*, 2nd Edition, Prentice Hall, 2001
- R. Housley and T. Polk, *Planning for PKI*, Wiley, 2001
- Y.-K. Hsu and S. P. Seymour, "An intranet security framework based on short-lived certificates," *IEEE Internet Computing*, March-April, 1998

