Welcome to the PKI WG (PKIX)

- Co-chairs:
  - Warwick Ford (Nortel)
  - Steve Kent (BBN)
- Area: Security
- Mailing list:
  - ietf-pkix@tandem.com
- Archive:
  - ftp://ftp.tandem.com/ietf/mailing-lists/current

Major Topics

- Scope of certificate use (application contexts)
- Profiling of X.509 certificates & CRLs
- PKI management (external to X.509)
  - certificate & CRL distribution
  - certificate registration
  - CA policy registration & distribution
  - user/application certificate validation management tools
- Other ...

PKI Application Contexts

- E-Mail
- IPSEC
- GSS-API
- WWW Security
- Payment Protocols
- Other?

Application Context Requirements for PKI

- Automated, identity-based access control
- Automated, rule-based access control
- Interactive, fuzzy, user-directed access control
- Naming people, machines, organizations, accounts, ...
- Online validation vs. CRLs
- Authentication vs. non-repudiation

Why Certification Authority Policies?

- Advising potential recipients of CA policies in support of management activities (e.g., CRLs)
- Limiting liability for CAs
- Limiting CA authority
- Limiting liability for users and organizations
- Input to identity or rule-based access control for applications accepting certificates

CA Attributes (from RFC 1422)

- Contact information (online & physical)
- Business characteristics (fees, liability, ...)
- Scope of certification (individuals, organizations, geography, application contexts, ...)
- CA security and privacy procedures
- Certification procedures & constraints
- Revocation management

Using a CA Policy Identifier

- ❑ For a CA
  - ✚ specify operational practices & procedures
  - ✚ establish CA liability limits
  - ✚ specify scope for certificate use
  - ✚ constrain a subordinate CA
- ❑ Constrain an end entity
  - ✚ liability limits
  - ✚ authorization characterization
- ❑ Other?

Relevant X.509v3 Extensions

- ❑ Identifying & constraining CA key use
  - ✚ CA vs. end entity
  - ✚ certificate signing vs. CRL signing vs. end entity uses
  - ✚ path lengths
  - ✚ subtree constraints
  - ✚ policy IDs applied to certificate use
- ❑ Pointers embedded in certificates
  - ✚ CA services, policy info, etc.
  - ✚ CRL distribution

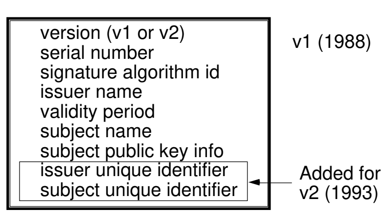
X.509 v3 Public Key Certificates

Warwick Ford
Bell-Northern Research
November 1995

PEM/X.509 Shortcomings

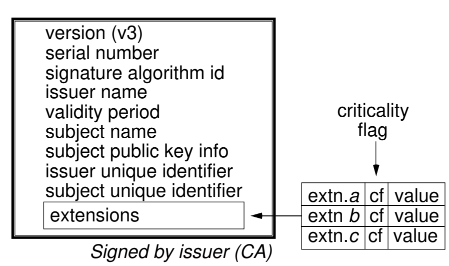
- ❑ An X.500 name does not adequately identify a subject to a certificate user
- ❑ PEM CA hierarchy is too rigid for many applications
- ❑ Certificate chains should start at point of most trust — may be close to certificate user, not always top of a hierarchy
- ❑ PCAs are an unwieldy way of managing policies
- ❑ Name subordination rule not good fit with many X.500 naming practices
- ❑ No support for key life cycles
- ❑ Revocation lists (CRLs) can grow excessively large

The X.509 Certificate (v1 and v2)



Signed by issuer (CA)

The X.509 Certificate (v3)



Signed by issuer (CA)

X.509 v3 Timeline

- 1988 — X.509 v1
- 1993 — PEM (RFC 1422)
- 1994 — ISO/IEC and ANSI X9 launch plans to upgrade X.509
- Jan 1995 — Draft Technical Corrigendum for v3 certificate and v2 CRL; Proposed Draft Amendment (PDAM) for standard extensions issued
- Aug 1995 — v3 Technical Corrigendum adopted; DAM for standard extensions issued
- Nov 1995 — ANSI Draft X9.55 includes minor enhamcements to DAM

Extension Criticality

- An extension comprises:
  - extension identifier (ASN.1 object identifier)
  - criticality indicator
  - extension field (syntax depends on extension identifier)
- Criticality identifier indicates whether an implementation which does not understand an extension may or may not safely ignore it
- Noncritical extensions do not inhibit interoperability

Key/Policy Information Extensions

- Authority Key Identifier: Identifies CA signing key by key-id or issuer/serial-no. pair
- Primary Key Attributes: Subfields for:
  - key-id
  - intended key usage indicator
  - private key usage period
- Certificate Policies: Carries policy identifier(s), with optional qualifier(s)
- Key Usage Restriction: Critical extension restricting use to specific policy(s)
- Policy Mappings: Mapping of policy id at domain boundary

Subject/Issuer Attribute Extensions

- Subject Alternative Names
  - Internet RFC822 name
  - Internet DNS name
  - X.400 Address
  - Directory name
  - Private name form
  - URL (added in X9.55)
- Issuer Alternative Names
- Subject Directory Attributes: Any other X.500 attributes that may help in identifying a subject

Chain Constraint Extensions

- Basic Constraints extension:
  - Indicates if subject is user or CA; if CA, can limit chain length
  - Constrains chain to specified name subtree(s)
- Name Constraints extension:
  - Constrains CA to specified name space or set of name spaces
  - Can stipulate name subordination to apply immediately or further down chain (2 different name subordination alternatives)
- Policy Constraints extension:
  - Can require explicit policy identifiers
  - Can inhibit policy mapping

CRL Distribution Points and Deltas

- CRL Distribution Points: Allows the certificate population from one CA to be partitioned into multiple CRLs
  - Each certificate indicates the Directory entry to go to for the correct CRL
  - May also partition based on revocation reason
- Delta CRL Indicator: Supports issuance of delta-CRL containing only changes since last CRL

Basic CRL Extensions

- CRL Number: Sequence number, which allows a missing CRL in the sequence of CRLs from one CA to be detected
- Reason Code: Gives revocation reason for each CRL entry
- Hold: Supports temporary suspension of a certificate
- Invalidity Date: Assists non-repudiation

ANSI X9.55 Enhancements on DAM

- GeneralName type includes RFC822 name, DNS name, X.400 name, EDI party name, directory name, private name, or URL
- Subject and issuer altnames now SEQUENCE OF GeneralName
- Subtrees constraint extended to permittedSubtrees and excludedSubtrees; each is SEQUENCE OF GeneralName
- New OID assigned to Basic Constraints
- CRL Distribution Point now identified by GeneralName

Where to Get Documents

- ISO/IEC TC, DAM, and ANSI draft all available at:

ftp://NC-17.MA02.Bull.com/pub/OSIdirectory/Certificates/

PKI Management Protocols

Warwick Ford
Bell-Northern Research
November 1995

Functions

- Registration: User first makes itself known to CA
- Initialization: Establish initial keys in required places, e.g., deliver CA public key to user
- Certification: CA issues certificate and delivers to client and/or posts in repository
- Key pair recovery: For use with a central private-key backup system (for encryption)
- Key pair update: Update key pair and issue new certificate
- Revocation request
- Cross certification: CA-to-CA exchange

Protocol Transport Options

- Protocol exchanges can be via:
  - on-line exchange
  - e-mail
  - WWW
- Protocol exchange definitions not tied to particular transport mechanisms

Specific Protocol Exchanges (1)

- ❏ One-Step Registration/Certification
- ❏ User Registration
- ❏ User Initialization/Certification with Client-Generated Encryption Key Pair
- ❏ User Initialization/Certification with Centrally-Generated Encryption Key Pair
- ❏ Encryption Key-Pair Recovery

Specific Protocol Exchanges (2)

- ❏ Key Pair Update for Client-Generated Key Pair(s)
- ❏ Key Pair Update for Centrally-Generated Encryption Key Pair
- ❏ Key Pair Update (Centrally-Initiated)
- ❏ Revocation Request
- ❏ Cross-Certification

4 December 1995

Denis Pinkas , Bull

IETF Public Key Infrastructure WG

Key Usage Control and Security Policy

Most common definition:

A private key is a key known at most
to the legitimate owner of the key, and
usable only by that person.

This is a wrong definition !

Correct definition:

A private key is a key usable only
by its legitimate owner

... and known at most to some trusted authorities.

Most common concept:

Signing a message with a private RSA key is
a decryption

This is a wrong concept !

Correct concept:

A signature scheme using RSA
MUST include a message redundancy.

This may be a one way hash function
or message redundancy as in ISO/ IEC 9796 :
Digital signature scheme giving message recovery

Correct statement:

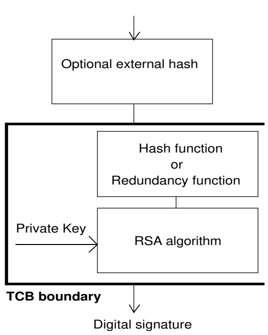
Enforcing certain types of key usage is _impossible_ in software.

Conclusion:

Key usage enforcement with a signature scheme using the RSA
algorithm can only be enforced by hardware

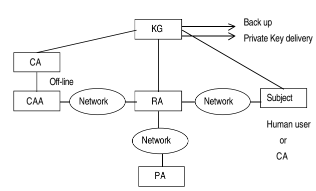
RSA based signature algorithm should be changed
Should always include a redundancy function or
a hash function is addition to an *optional* external hash.

This is necessary to enforce key control for the RSA algorithm.



Comments from Denis Pinkas (Bull)
on the Public Key Infrastructure (PKI) draft

The model we need:



CA : Certification Authority (off-line)

CAA : Certification Authority Agent (on-line)

RA : Registration Authorities

KG : Key Generator

PA : Publishing Authority (e.g. the Directory)

Users knows their RAs, not directly the CAs

Major comments:

A CA should be only off-line.
A CA should not keep track of issued certificates

Non repudiation should be supported :

- Problem with CRL issuing rule
  (do not remove as soon as expired)
- Problem with publication of CRLs
  (more than current one needed)
- How to go from a current trusted point to an old one

Note: invalidity date is supported and is fairly important

Renewal of certificate (without private key change)
is not supported.

Certificate requests should be extended to be requested by someone already registered (eliminates the chicken and egg problem)

Key installation seems to be a chicken and egg problem with manipulation of confidential information.

No key pair updates can be done centrally
except by E-mail : should be discouraged
(do not want unsolicited mails)

CA key change needs more work.
(validity of user certificate versus validity of CA key)
Basic question: one or two CAs keys needed per CA ?
Current and new ... OR ... previous and current.

CA key change from CA point of view should not be different whether the CA is a trust point or not.

The key usage restriction field, when present, has
a policy OID. What is the relationship, if any, with the certificate policy (4.2.4) "OID of the policy under which the certificate has been issued" ?

Should we really allow a CA to revoke certificate
not issued by itself ? How does this works with the "basic constraints" ?
Crypto devices (e.g. smart cards) generating key pairs should be supported

Minor comments

Back-up of private keys, when allowed, done by the KG component, i.e. not necessarily by the CA. "Difficult" problem since related to key escrow. Has to be a legislative decision, not a pure technical solution.
For further study ....

"Names will not be re-used" (p 11) . explanations needed. Is it because UUIDS can be replaced by
CA name + FDN + date of issue of certificate ?

Key pair update time should be advertised in certificate
(new extension needed)

Cross certification can be accomplished much more simply by exchanging hashes rather than secret "authentication codes"

CRLs inquiries should be supported

Revocation requests should be defined

RSA and DSA patent status missing