

# Computer Security

## Public Key Infrastructure

1

# Purpose

- enables secure use of unsecure network
- network may be public
- publishes public-key values

2

# Needs

- public key cryptography
- certificates

3

# Public Key Infrastructure

- Certification Authority
- Directories
- many commercial products exist

4

## Basic Operations

- Certification
- Retrieval
- Verification (or validation)

## Certification

- binds a public-key value to an entity (principal)
- entity may be
  - individual
  - organisation
  - permission
  - credential
  - some other system entity

## Verification

- process of verifying that a certificate is still valid
  - checking signature
  - checking validity period
  - checking for revocation

## Certificates

- form in which PKI transmits
  - public keys
  - information about public keys
  - both
- may simply be a public key and a name
- usually some information signed by its issuer

## Types of Certificates

- identity certificate - identifies an entity and gives public key(s)
- credential certificate - describes non-entities, such as permissions or credential

## Certificate User

- relies on information in certificate
- trusts issuing authority to issue correct information (“true” certificates)

## Certification Authority

- issuer of certificates
- how much you trust a CA will determine how much you trust the certificates it has signed

## Use

- without PKI need to know public key of entity to securely communicate with entity
- with PKI only need to know CA’s public signing key
- can then verify certificate supplied by entity with whom you wish to communicate

## Relationships

- are CAs distinct from subjects and users?
- can users and subjects be CAs?
- must each employer have a CA?

## Trust

- very complicated concept
- rarely defined in absolute terms
- PKI only a tool for expressing trust relationships

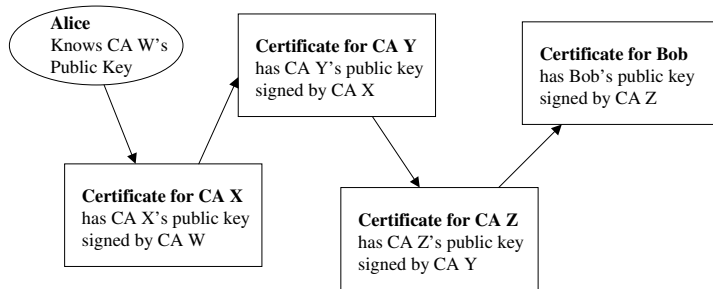
## Trust

- how much do users trust a CA?
- how much do CAs trust each other?
- Some PKIs allow for limited amounts of trust
- eg., I trust this CA to issue certificates which bind a public key to an e-mail address

## Multiple CAs

- obviously impractical to have all certificates signed by a single CA
- most PKIs permit CAs to certify other CAs
- there may be an arbitrary number of CAs “between” two communicating entities

## Multiple CAs



17

## Arrangement of CAs

- important feature of a PKI
- may be
  - general hierarchy
  - top-down hierarchy
  - unstructured (web)
- a hierarchy may use cross-certification to shorten paths

18

## General Hierarchy

- CAs certify children and parent
- likely to include cross-certification

19

## Top-Down Hierarchy

- CAs certify children only
- not likely to include cross-certification

20

## Unstructured

- no regular structure
- may simply work on cross-certification
- may require enough other CAs to issue certificates to trust binding
- this is called a **web of trust**

## Scalability

- can a PKI handle millions (or billions) of users?
- paths must be short
- paths must be easily discovered

## Unstructured

- paths short
- even for world-wide generally 6 to 7 at most
- but path discovery could be difficult

## Top-Down Hierarchy

- scales better
- however all users must trust root for all purposes
- all users need public key of root
- unsuitable for world-wide PKI

## General Hierarchy

- many paths still traced through root
- cross-certification can complicate path discovery

## Validation

- information can change over time
- user of certificate needs to know that the contents of a certificate still correct
- user needs to **validate** certificate

## Validation

- user can ask issuing CA - online validation
- CA can include validity period (two dates) in certificate - offline validation
- a PKI can use either or both

## Revocation

- what if certificate becomes invalid?
- not a problem with online, but bandwidth and processing requirements may be too high
- if use offline validation, need to be told
- most common method is **certificate revocation lists (CRLs)**

## CRLs

- list of revoked certificates
- list is periodically issued by CA
- list is signed by CA
- user needs to check latest CRL during validation

## Delay

- there is delay between revocation and distribution of CRL
- what if in that time user checks validity of a revoked certificate?
- user will assume certificate valid

## CRL Size

- CA may validate 1000's or even 100,000's of subjects
- CRL may grow large
- difficult to retrieve
- time-consuming to check signature

## Solution 1

- different categories of CRL
  - one for CAs, one for subjects
  - one for information changes, one for security compromise
- makes locating appropriate information easier
- good for size, does not help with delay

## Solution 2

- issue periodic updates (delta-CRLs)
- delta CRL is signed list of changes since last full CRL

## Authentication

- user validates a certificate for an entity
- entity then said to be authenticated
- degree of trust in certificate effects strength of authentication
- authentication may be for identity certificates or credential certificates or both

## In-Band/Out-of-Band

- when authentication done using PKI, either offline or online, termed in-band
- when authentication done outside computing system, termed out-of-band
- PKI design would like to minimise out-of-band, but will almost certainly never eliminate it
- users need to be authenticated out-of-band before added to system

## Irrefutability

- would like users not to be able to deny their signatures
- important if PKI basis for replacement of paper signatures
- the more out-of-band contact with a user the less opportunity for fraud

## Anonymity

- would like only the minimum necessary information to be used
- PKI should provide strong, irrefutable, authentication and a high degree of privacy through anonymity

## X.509

- authentication framework
- supports X.500 directories
- oldest proposal for a world-wide PKI

## X.500

- very similar to a telephone directory
- given a person's name can find other information

## Directory Contents

- more than just address and phone number - name of employers, job title, e-mail, etc
- X.500 directory entry can represent any real-world entity - computers, printers, companies, governments, nations
- entry can also contain certificate with entity's public key

## Naming

- each entity given globally unique name
- **distinguished name** or **DN**
- X.500 directory is hierarchical
- called the **Directory Information Tree (DIT)**

## DIT Structure

- strict hierarchy
- under root is an entry for each country (named by two letter code)
- under this entry for government, states, companies, etc
- eventually get down to people, etc

## Naming

- each vertex has a **relative distinguished name (RDN)**
- unique amongst siblings
- RDN's of ancestors concatenated with vertex's RDN to give DN

## X.509 Certificates

- X.509 created to support authentication of X.500 entries
- currently version 3 is standard
- first we will look at version 2
- There is a version 4, but not standardised yet and not all that different to version 3

## X.509v2 Certificate

Certificate Version
Certificate Serial Number
CA's Signature algorithm ID
CA's X.500 name
Validity Period
Subject's X.500 name
Subject's Public Key information
Issuer Unique Identifier
Subject Unique Identifier

Stored together with digital signature signed with CA's private key

## Fields

- version - X.509 version
- serial number - unique number assigned by issuing CA
- CA signature algorithm - identifies algorithm used by CA
- Issuer name - X.500 name of CA
- Validity Period - pair of dates

## Fields (cont.)

- Subject name - X.500 name of entity holding corresponding private key
- Public Key information - value of public key and identifier of algorithm with which it is to be use

## Optional Version 2 Fields

- Issuer unique id - bit string to make X.500 name of CA unique
- Subject unique id - ditto for subject
- X.500 names might be assigned, de-assigned and re-assigned
- these can address that problem
- not widely used, difficult to manage
- left out in many implementations

## X.509 CAs

- X.509 documentation does not dictate the CA hierarchy structure
- does describe general hierarchy with cross-certificate
- usually arranged in hierarchy following X.500 DIT

## Problems

- X.500 and X.509 designed in 80's
- before growth of Internet
- intended for offline environment
- machines only intermittently connected
- X.509v1 & v2 use simple CRLs
- no attention to size or time-lag problems

## X.509v3

- attempts to solve CRL size and time lag problems
- fundamental change - certificate and CRL formats are extensible
- some "standard extensions" defined

## Certificate Extensions

- certificate policies and policy mappings
- CAs may include a list of policies followed in creating certificate
- user can decide if certificate suitable
- eg., policy may say key certified for casual e-mail
- should then probably not be used for financial transactions

## Certificate Extensions

- alternative names
- certificate can contain one or more alternate names for subject or issuer
- X.509 can then operate without underlying X.500 directory
- may be e-mail addresses, URLs, etc

## Certificate Extensions

- subject directory attributes
- allows any additional X.500 directory entry attributes to be included in certificate
- certificate can now carry more than just subject name

## Certificate Extensions

- certification path constraints
- CA can restrict the certification paths that grow from certificates it issues
- can state if subject is a CA
- can restrict on policy and/or name space

## CRL Extensions

- CRL number and reason codes
- number assigned is always one greater than last one
- user can see if a CRL has been missed
- reason self-explanatory

## CRL Extensions

- CRL distribution points
- CA can partition CRL and issue segments from different points
- eg., a company CA partitions on division
- when a user wants to check another's certificate, checks the CRL for the subject's division

## CRL Extensions

- delta CRLs
- as above
- users can use deltas to update their copy of the CRL

## CRL Extensions

- indirect CRLs
- allows CRL to be issued from an entity other than the CA
- one point can gather a number of CAs' CRLs and issue them together

## Provision of Flexibility

- X.509v3 very flexible
- however, not provided in a very useable manner

## Object Identifiers

- need to identify many things - signature algorithms, policies, alternative names, user-defined extensions, etc
- X.509v3 assigns internationally defined **globally** unique object identifier

## Object Identifiers

- eg., 2-16-804-1-45356 for a CA
- standards body, category, country, organisation, the CA
- a policy it defines might be 2-16-804-1-45356-3-15
- 3 for policies, 15 for the policy

## Use

- any entity using identifier has to know what it means and how to use it
- same object can also be assigned multiple OIDs
- every CA could give a different number to each cryptographic algorithm
- at present no systematic method of resolving OIDs