

COMSM0213 : Public Key Infrastructure

E. Oswald and N.P. Smart

Department of Computer Science,
University Of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB
United Kingdom.

November 8, 2009

Outline

PKI Basics

Types of PKIs
PGP
X.509

Issues around Certificates and Signatures

Authorization vs. Authentication
Long term signatures
Key Escrow
Digital Signatures vs. Electronic Signatures

Real World Applications

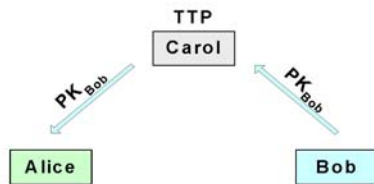
SSL Overview
IPSec

Distribution of Public Keys



- Get key face-to-face
 - Distance?
 - Meet every time you want to verify a signature?
 - Can you verify someone's identity?

Distribution of Public Keys



- Get key from a trusted third party (TTP):
 - Makes the distribution easier
 - How does one assure that the keys are authentic?

Two different types of PKIs are mainly used in practice.

- ▶ PGP
 - ▶ from the open-source area
 - ▶ relatively simple
- ▶ X.509
 - ▶ based on X.500
 - ▶ big and complex

The difference between the types is in the [trust model](#).

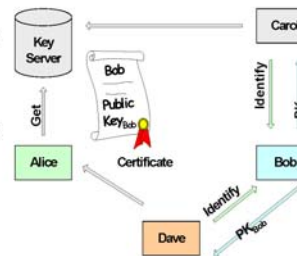
Distribution of Public Keys



- Get key in certified form:
 - Initial key exchange unsolved
 - Revocation mechanism required

Pretty Good Privacy (PGP)

- User-centric model ("web of trust" according to the PGP terminology).
- The user acts as CA and signs other users' keys.
- Is trust transitive?
- Revocation?
- User education?



Public Key Infrastructure

The system which provides authentic public keys to applications is called a [public key infrastructure](#) or PKI.

- In public key systems the main challenge is
- ▶ [key authentication](#) (i.e. which key belongs to who).
 - ▶ [key revocation](#) (i.e. notifying users which keys are no longer secure).

Keys need to be distributed via [authentic channels](#).

PGP certificates

- ▶ PGP version number
- ▶ certificate holder's public key
- ▶ certificate holder's information
 - ▶ identity information like name
 - ▶ user ID, photograph, etc.
- ▶ digital signature of the certificate owner (signer)
- ▶ validity period
- ▶ preferred symmetric encryption algorithm

The most remarkable feature of a PGP certificate is that it can contain signatures from more several signers.

PGP: Trust management

- ▶ Basically the trust level of a certificate is deduced from levels of trust that the user can assign to public keys:
 - ▶ complete trust
 - ▶ marginal trust
 - ▶ no trust
- ▶ Depending on the level of trust and the signatures, PGP assigns a validity level to each certificate:
 - ▶ valid
 - ▶ marginally valid
 - ▶ invalid

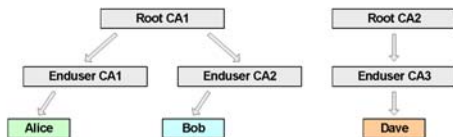
Signing With PGP

Either DSA or RSA can be used as the signature algorithm

PGP creates digital envelopes using a message digest

- ▶ MD5/SHA-1 used as the message digest function
- ▶ Timestamp is added
- ▶ Result signed with the private key

X.509



- Hierarchical structure:
 - A root CA is established and a self-signed root certificate is the basis of trust for all entities in the hierarchy
 - The root CA certifies zero or more CAs immediately below it
 - Each of those CAs certifies zero or more CAs immediately below it
 - At the second-to-last level, the CAs certify end-entities
- Public key certificates

Uses of PGP

Authenticating email

- ▶ Helps to stop problems of email masquerading

Authenticating other transactions

- ▶ Remote Web updating

Secure email for roaming employees

Known to have been used by human rights activists to help get details of atrocities out to the world,

- ▶ Without the senders being found out.

PGP: Key Pair Generation

The initial entropy to seed the PRNG

- ▶ Obtained from inter-character timings from user at the keyboard.

Finds prime numbers using simple tests for primality

- ▶ Not guaranteed to use prime numbers, but highly likely.

For RSA

- ▶ Stores p and q to speed up the private key operations

All private data protected by IDEA under a pass phrase.

Cross-Certification and Cert Chains

If more than one CA exists, then a user may not have a trusted copy of the CAs public key needed to verify another users certificate.

This is solved by **cross-certificates**, i.e. one CA's public key is signed by another CA.

The user first verifies the appropriate cross-certificate, and then verifies the user certificate itself.

With many CA's one can get quite long **certificate chains**

Basics Of PGP

Uses RSA public key encryption for low volume data such as session keys.

Session keys are generated by a cryptographic pseudo random number generator, PRNG.

Data is compressed before transmission.

IDEA is the bulk encryption algorithm

- ▶ 64 bit block size
- ▶ 128 bit key size
- ▶ Used in 64 bit CFB mode.

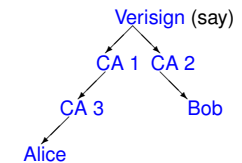
Managing PGP Key Rings

Requires inserting information into a local file

- ▶ Individuals have control over their own local public key store
- ▶ Does not rule out a centralised public key store, but this is not needed.

Invalidating compromised keys This is a major problem of all systems, particularly PGP

- ▶ Have an ad-hoc method of **tell your friends my key is broken**



Alice trusts Verisign

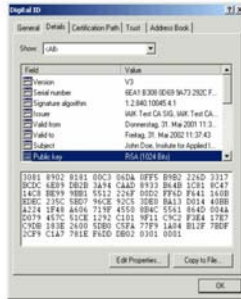
- ▶ She obtains Bobs public key which is signed by the private key of CA2
- ▶ She obtains CA2's public key which is signed by the private key of Verisign.

Hence she trusts Bob's public key.

X.509 Certificates

• Certificates contain

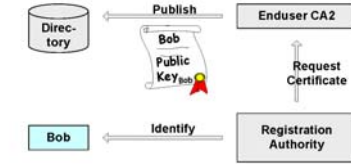
- Version Number
- Serial Number
- Issuer Name
- Validity Period
- Subject Name
- Issuer/Subject Unique IDs
- Public Key Info
- Extensions
- Signature of the CA



X.509v3 Certificate Extensions

- ▶ key identifiers
 - ▶ authority key id
 - ▶ subject key id
- ▶ key usage
 - ▶ digital signature
 - ▶ key encipherment
 - ▶ key agreement, etc.
- ▶ CRL distribution point
- ▶ certificate policies
- ▶ basic constraints
 - ▶ path length constraint
- ▶ extended key usage
 - ▶ TLS server authentication
 - ▶ code signing, etc.
- ▶ extensions marked as critical must be processed
- ▶ non-critical extensions are to be processed if possible

X.509 Registration of Users



- RA establishes and confirms the identity of an individual, in addition it:
 - May generate keying material on behalf of users
 - May perform key/certificate life-cycle management functions

X.509 Certificate Revocation

There are no standardized means of revoking certificates (yet).

However, there are two techniques for revocation checking:

- ▶ CRL (certificate revocation list)
- ▶ OCSP (online certificate status protocol)

A CRL is a list of the serial numbers of all the certificates revoked by a particular CA, signed by the CA concerned.

OCSP is a client-server approach for revocation checking. The client sends a request to the server, the server answers (answer is signed).

Attribute Certificates

A public key certificate binds a user to a public key. It can therefore be used in a protocol to authenticate a user.

In practice, it is also important to assign roles to users. Roles define what users may or may not do.

- ▶ use X.509 extension: not good because roles may change often
- ▶ attribute certificates: binds attributes to an entity

Long term signatures

In some applications it is necessary for signatures to remain valid for a long time.

Revocation of a public key, even long after the legitimate creation of the signature, potentially invalidates all digital signatures made using that key.

Methods need to prove that a digital signature was made prior to the revocation

Which brings us back to the concept of time stamping: A **Time Stamping Service (TSS)** is a means whereby a trusted entity will take a signed message, add a date/time-stamp and sign the result using its own private key. What happens if the key of this service gets revoked?

Key Escrow

There are problems with truly secret keys

- ▶ What is someone loses or forgets a key ?
- ▶ What if the holder of the key resigns or is killed ?
- ▶ What if the user is a criminal ?

Solution: Deposit your key with someone else incase you lose yours.

On the other hand simply divulging the key to anybody, even (or perhaps especially) the government is very insecure.

Key Escrow

A proposed solution is Key Escrow:

- ▶ The private key is broken into pieces, which can be verified to be correct.
- ▶ Each piece is given to some authority.
- ▶ The whole key can only be reconstructed if all the authorities agree.

The Escrow agency is another example of a Trusted Third Party, since you **really** have to trust it.

This was the basis of the proposed US Clipper Chip and some (now defunct) UK proposed legislation.

- ▶ Once a political hot potato

Electronic Signatures

The most important framework that regulates electronic signatures is the Directive 1999/93/EC of the European Parliament:

Data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication.

Hence, the definition of an electronic signature is pretty general. A digital signature fits however into this framework.

Advanced Electronic Signature

Needs to fulfill four requirements:

- ▶ It must be **uniquely** linked to the signatory.
- ▶ It must be capable of **identifying** the signatory.
- ▶ It must be created using means that the signatory can maintain under his **sole** control.
- ▶ It must be **linked to the data** to which it relates in such a manner that any subsequent change of the data is detectable.

Who may issue Qualified Certificates?

Only CSPs that fulfill certain criteria:

- ▶ Provide a secure directory and immediate revocation service.
- ▶ Provide verification of identity of users.
- ▶ Employ personnel who possesses the necessary qualifications.
- ▶ Use trustworthy systems and products and maintain sufficient financial resources.
- ▶ Record all relevant information. But it must not store or copy signature creation data.
- ▶ Inform users by durable means of the terms and conditions regarding the use of certificates.

SSL Overview: Structure

Bulk encryption algorithm with secure key exchange: Client and server agree on one of a number of possible bulk encryption algorithms, and use X.509 public key certificates for key authentication.

SSL uses a number of protocols

- ▶ Handshake protocol (authentication, key exchange, negotiate cipher suite)
- ▶ Change cipher spec protocol (part of handshake)
- ▶ Alert protocol (for error messages)
- ▶ Record protocol (encryption and authentication of data)

Secure Electronic Signature

A secure electronic signature is an advanced electronic signature

- ▶ which was produced on a **secure signature creation device** and
- ▶ which is based on a **qualified certificate**.
- ▶ It must be assured that the signature creation data exists only once and is protected from the user from the use of others.

Qualified electronic signatures are considered equal to handwritten signatures from a legal point of view (at least in Europe).

Secure Socket Layer, Overview

PGP was driven by altruistic ideals, SSL was driven by commercial requirements.

- ▶ Web based shopping/sales

SSL adds security to TCP level (Network layer)

- ▶ Security of data and not parties
- ▶ Various protocols can then be transparently layered on top
 - ▶ HTTP, FTP, TELNET, etc

Commercial standard driven by Netscape

- ▶ Issued as an RFC

Now SSL is known as **TLS**.

SSL Handshake

The following is a simplified overview

- ▶ Client establishes connection with Server
 - ▶ On a secure port
- ▶ Server gives certified public key to client
- ▶ Client chooses random secret
- ▶ Client encodes this with the Server's public key
- ▶ Client and Server now securely share secret
- ▶ Server authenticates itself by responding using the secret

This is the traditional way SSL operates.

Qualified Certificate

A qualified certificate must contain an indication that the certificate is issued as qualified certificate.

It must contain

- ▶ the identity of the certificate service provider (CSP) and the state
- ▶ the name of the signatory or pseudonym
- ▶ the signature-verification data, the validity period, the identity code of the certificate, and the advanced electronic signature of the CSP
- ▶ limitations on the scope of use (if applicable), and
- ▶ limits on the value of transactions (if applicable) must be included.

SSL Overview: Objectives

Aims to provide **channel security**

Private:

- ▶ All traffic is encrypted after an initial handshake.

Authenticated:

- ▶ The server end is always authenticated (for the benefit of the client)
- ▶ The client may optionally be authenticated too.

Reliable:

- ▶ The message transport includes an integrity check

SSL Handshake: Key Exchange

A DH based key exchange mechanism was introduced for a variety of reasons...

- ▶ Forward secrecy
 - ▶ This means better security
 - ▶ No long term encryption keys for governments to ask for
 - ▶ Only long term key is the signing key
- ▶ Some people do not like using RSA for encryption
 - ▶ Or even using RSA

SSL Handshake: Key Exchange using DH

The following is a simplified overview

- ▶ Client establishes connection with Server
 - ▶ On a secure port
- ▶ Server gives certified public key to client
- ▶ Client and server engage in a signed DH key exchange
 - ▶ Client signatures are optional and rarely done in practice
 - ▶ Signatures are on the DH parts, plus nonces to ensure freshness.

IPSec

Introduced to provide protection at the IP layer

Provides **Authentication** and **Confidentiality**

Used in building VPNs and stopping various attacks

Part of IP v6

Can be implemented directly as part of the IP stack.

Authentication Header

This has six basic fields

Next Header :

- ▶ 8 bits (Not of interest to us)

Payload Length :

- ▶ 8 bits (Not of interest to us)

Reserved for Future Use :

- ▶ 16 bits (Not of interest to us)

Security Parameters Index (SPI) :

- ▶ 32 bits

Sequence Number Field :

- ▶ 32 bits

Authentication Data :

- ▶ Multiple of 32-bit blocks

SSL Record Structure

Need to create a block to apply the bulk encryption algorithm

Records may be padded or unpadded

- ▶ Padding used to create an integral number of plaintext blocks
- ▶ Padding does not occur on records sent unencrypted

Records have sequence numbers to counter replay attacks

Dummy's Guide to an IP Packet

An IP (v4) packet essentially looks like the following

IP Header

- ▶ **IP Header Fields**
- ▶ **Source IP Address** (i.e. 123.456.789)
- ▶ **Destination IP Address** (i.e. 987.654.321)

Upper Layer Fields

The Upper Layer Fields contain the actual packet information and protocol information.

Many attacks

- ▶ e.g. Reply attacks, reflection attacks, spoofing, sniffing, hijacking

can make use of the fact that the Address fields can be altered at will

Authentication Header

The SPI defines a unique cryptographic algorithm to provide authentication

- ▶ e.g. A MAC

The Sequence Number Field can be used to prevent replay attacks

The Authentication Data is the integrity protection which is created using a secret key known to recipient and sender and allows the receiver to detect alterations in the IP Header.

SSL Authentication

Recall that the server gives it certified public key to the client

Uses the X.509 standard

- ▶ Hierarchical global CA system

Optimised reuse of session keys

- ▶ Client can quote previous session key
- ▶ Server can accept this or create a new one
- ▶ Session keys have very limited lifetime
- ▶ Any fatal error invalidates the session key

IPSec Packet

An IPSec packet looks as follows

IP Header

- ▶ **IP Header Fields**
- ▶ **Source IP Address** (i.e. 123.456.789)
- ▶ **Destination IP Address** (i.e. 987.654.321)

Authentication Header (AH)

Upper Layer Fields or **Encapsulating Security Payload (ESP)**

The AH authenticates the header whilst the ESP provides confidentiality (and authenticity) for the payload.

- ▶ The ESP is optional, can be confidentiality only or with authenticity
- ▶ Separate protocols needed for key management, e.g. Internet Key Exchange (IKE)
- ▶ All protocols can be arbitrarily combined!

Encapsulating Security Payload

The ESP is an encrypted version of the payload, it has 7 fields

- ▶ **Security Parameters Index (SPI) : 32 bits**
- ▶ **Sequence Number Field : 32 bits**
- ▶ **Payload Data : Multiple of 32-bit blocks**
- ▶ **Padding : 0-255 bytes**
- ▶ **Pad Length : 8 bits**
- ▶ **Next Header : 8 bits**
- ▶ **Authentication Data : Multiple of 32-bit blocks**

Encapsulating Security Payload

The SPI now defines the precise encryption algorithm

Sequence number is as before

The Payload Data is the ciphertext.

- ▶ Since the plaintext may need to be padded the padding is also given
- ▶ Requires secret key

The Authentication Data is **optional** and authenticates the ciphertext

- ▶ It **should** be mandatory since it could lead to some attacks if missed out

IPSEC can be used in 2 modes

transport mode (used between end-systems):

- ▶ protection is applied to the payload of the IP packet
- ▶ ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header
- ▶ AH in transport mode authenticates the IP payload and selected fields of the IP header

tunnel mode (used between gateways or host and gateway):

- ▶ the entire IP packet is considered as payload and encapsulated in another IP packet (with potentially different source and destination addresses)
- ▶ ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet
- ▶ AH in transport mode authenticates the entire inner IP packet and selected fields of the outer IP header

IKE Protocol

Both the AH and ESP require sender and receiver to agree on a secret key

IKE = Internet Key Exchange Protocol

Negotiates algorithms to be used, parameters etc

- ▶ Various options

Can be based on

- ▶ Preshared symmetric keys
- ▶ Public key encryption
- ▶ Public key signatures (similar to STS protocol)