Chapter 7. Identity Public Key Certificates and Infrastructure©

We learned in Chapter 6 about three mechanisms that deploy public keys: digital signatures, key agreement and public key cryptosystems (encryption). All of these mechanisms used the public key of an entity (to validate its signatures and to encrypt messages to it, respectively). In this chapter and the next one, we will explore questions related to the use of public keys, such as:

- Where and how can we obtain the public key of entities?
- How do we know it really belongs to the entity we want to interact with?

In phrasing the questions above, I intentionally avoided the question of identification of the entity. This question is highly controversial. The ITU-T X.509 Recommendation [X.509] is the core of the following simple, and widely adopted, approach: X.509 identifies each entity by a single, well known, public recognizable identifier, such as used in `normal` (non-electronic) commercial and social interactions. The X.509 term for such an identifier is a *Distinguished Name (DN)*; we discuss distinguished names (and the X.509 approach) in Section 7.2. A globally unique distinguished name may make it easy, for instance, to make legal arguments related to commitments done in Cyberspace, by a digital signature. *Public Key Infrastructure (PKI)*, the term usually associated with this camp, implies mechanisms that link between public keys and distinguished names. Indeed, this approach, of using distinguished names, is widely adopted by most legal and business experts and forums.

However, in the recent years, another approach seems to be gaining ground, especially among computer scientists and engineers (including the author). This approach considers identification as just one mechanism, and not always the best, to achieve the `real goal`, which is to allow public keys to be used for enabling secure commercial and social interaction (over the Net), following the economic principles of rational behavior and reputation. The `real` goals and objectives are:

- *Establishing trust and reputation,*
- *Allocating liability,* and
- *Penalizing undesirable actions.*

In fact, we claim that this approach is simply a generalization of the first approach. Namely, distinguished names or any other public, legally meaningful identifiers are simply

specific mechanisms for establishing trust and liability, and for penalizing undesirable actions. Using identifiers we can:

- Establish trust using recognized or reviewed identities (of corporations and individuals)
- Allocate liability by identifying the signer of a document
- Penalize undesirable actions by suing the identified entity.

By using distinguished names which are well defined, legally meaningful names, we can use the existing means of reputation, liability and judgment which exist in every modern country, and to substantial extent also internationally. However, sometimes it is possible to provide alternative or complementing mechanisms for trust, liability and penalties, as integral part of the electronic communication or commerce system. Some potential advantages include:

- More efficient mechanisms (in time or cost)
- Support for cases where the existing legal system may not be sufficient or appropriate, e.g. international interactions where legal and law-enforcing systems are unreliable, incompatible or non-interoperable.
- Privacy considerations – avoiding exposure of the identify of individuals (or companies)
- Dealing with problems of the distinguished names approach (see below).

In this chapter, we discuss the more traditional, identity-based public key infrastructure (PKI) approach of X.509. First, let us define the basic concepts, and in particular – certificates.

7.1. Certificates and related concepts

We begin with discussing the main concepts and terms related to certificates, in a generic manner, i.e. without focusing on one of the particular approaches or standards. We later focus on specific techniques and standards, in the following sections.

When Bob receives a public key, he needs to know some properties of the holder of the corresponding private key. When Bob knows the holder of the private key, say Alice, he may be content with receiving the public keys directly from, in person or via a secure (authenticated) channel. However, in other cases, Bob may not know Alice in advance. Or, when the public key is used to validate Alice's signature over some document, Bob may need a way to prove to a third party (possibly later) that Alice signed the document.

What Bob needs, in these cases, is a signed statement, called a *public key certificate*, claiming some well defined *attributes* (properties) related to the public key and to the holder of the corresponding secret key (Alice; often called the *subject* of the certificate). The party that signed and issued the certificate, called the *Issuer*[1], is responsible for its validity. Bob relies on the certificate; we therefore refer to him as the *relying party*.

[1] We define special kinds of issuers called Certificate Authority and Attribute Authority below.

Certificates are similar to charge card, membership cards, and similar mechanisms used in daily life. Figure 7.1 shows the library card of Alice and the corresponding public key certificate that the library issued to Alice.

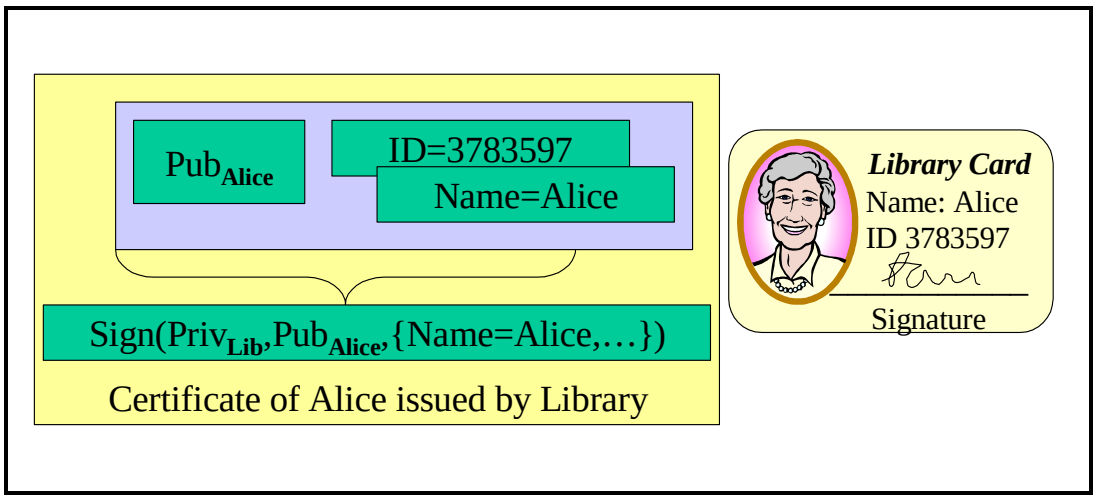

Figure 7.1: A Public Key Certificate and a Library Card

The library card contains the sample of Alice's signature; this allows any librarian to validate Alice's identity as the authorized owner of the card. Notice that in our example, the card also contains Alice's picture, but that is really just as an extra identification mechanism, used in addition to the signature or where the signature is not convenient. The librarian also uses the signature sample to confirm that Alice properly signs out her books, so that she cannot claim later that somebody else signed. The public key in the certificate serves the same purposes of identification and non-repudiation (signatures).

Both library card and certificate contain (in this example) two attributes: a name and a library identity number. The librarian uses these to retrieve and update Alice's file. Notice that the name does not really provide any direct identification; however, maybe the librarian can ask Alice to produce an additional identity card for additional confirmation of her identity, e.g. if the signature and picture got blurred. This problem is not relevant to the digital certificate.

We now define a public key certificate.

Definition 1 A *public key certificate* is a public key signature on a statement containing a (different) *public key* and one or more *attributes*: *Sign(Priv$_{Issuer}$,{Pub$_{Subject}$, attributes}).*

Often, the most important attribute in the certificate is the identity of the subject (the owner of the secret key). We call such a certificate *identity certificate.* In many manuscripts and systems, all public key certificates contain the identity of the subject (identity certificates according to our terminology).

Some works, most importantly the X.509 standard, define also a concept of an *attribute certificate*. An *attribute certificate* is a signed statement containing a list of attributes, but

(usually) *not* a public key. While this definition is general, the term attribute certificate is used almost only when discussing such statements which are encoded according to the X.509 standard. An attribute certificate identifies its subject using an identifier of the subject or by containing an identifier of a public key certificate.

The X.509 standard, and many other publications, use the term *Certificate Authority (CA)* for the issuer of identity certificates. Similarly, X.509 and others use the term *Attribute Authority (AA)* for the issuer of attribute certificates.

7.2. X.509 Public Key Frameworks

The International Telecommunications Union, ITU-T (formerly known as CCITT), is a multinational union that provides standards for telecommunication equipment and systems. In particular, ITU-T defined standards for open, electronic directories, the X.500 recommendation. From [RFC2693]:

> X.500 was to be a global, distributed database of named entities: people, computers, printers, etc. In other words, it was to be a global, on-line telephone book. The organizations owning some portion of the name space would maintain that portion and possibly even provide the computers on which it was stored. X.509 certificates were defined to bind public keys to X.500 path names (Distinguished Names) with the intention of noting which keyholder had permission to modify which X.500 directory nodes. In fact, the X.509 data record was originally designed to hold a password instead of a public key as the record-access authentication mechanism.
>
> The original X.500 plan is unlikely ever to come to fruition. Collections of directory entries (such as employee lists, customer lists, contact lists, etc.) are considered valuable or even confidential by those owning the lists and are not likely to be released to the world in the form of an X.500 directory sub-tree. For an extreme example, imagine the CIA adding its directory of agents to a world-wide X.500 pool.

The X.509 effort began as a definition of authentication mechanisms related to the directory, and included shared key cryptography as well as public key cryptography. The most important aspect, however, seems to be the part which deals with certificates: ITU-T Recommendation X.509: Public Key and Attribute Certificate Frameworks [X.509]. There are many deployments of the X.509 standard, including important protocols and standards such as TLS / SSL, IPSEC, S/Mime, WAP, and most notably the family of the PKIX standard protocols. X.509 is now in its fourth revision (released on 2000).

7.3. X.500 Distinguished Names

Distinguished Names (DN) are an important element of the X.500 directory, which has strong impact on X.509. The idea of a DN is to provide a single, globally unique name that everyone could use when referring to an entity. The distinguished names in X.500 and

X.509 are defined as a sequence of pre-defined keywords, and a string value for each of them. The keywords for the distinguished names are defined in another recommendation in the series, X.520. Table 7.1 lists some of the keywords.

Table 7.1: Partial list of Distinguished Name Keywords

Keyword	Meaning
C	Country
L	Locality name
O	Organization name
OU	Organization Unit name
CN	Common Name

The distinguished name was built as the ordered sequence of these keywords and their values, from the most general (usually country, C) to the most specific (usually common name, CN). For example: C=US/L=NY/O=NYPD/OU=Soho/CN=John Doe.
The keywords in the distinguished name were interpreted as a lexical ordering, from the more general to the more specific. The assumption was that (usually?) the distinguished names will form a hierarchy (tree), with each layer corresponding to a keyword; and each internal node mapping to an entity responsible for naming the lower layer. See example in the figure below.

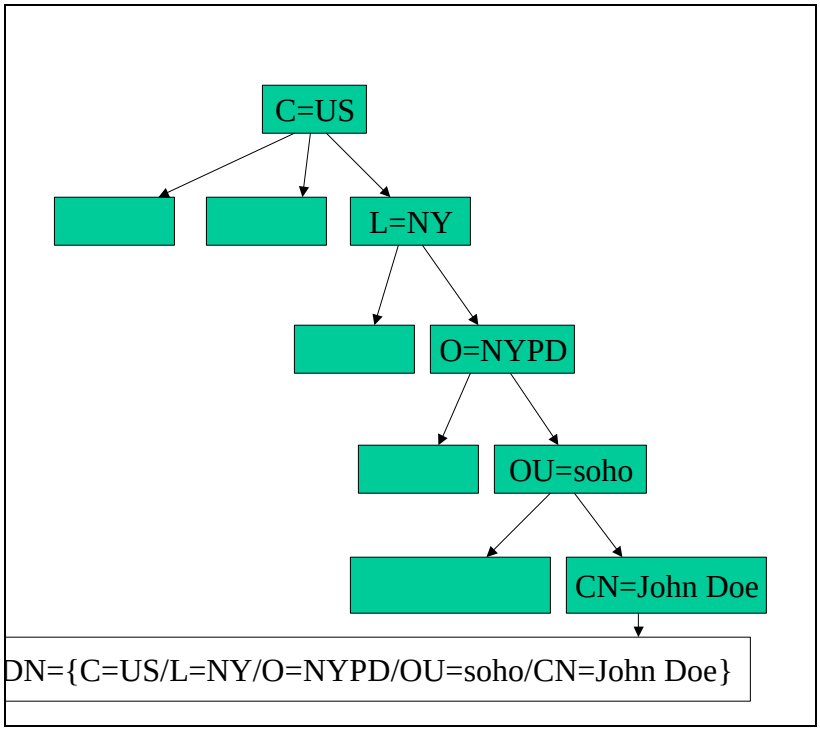

Figure 7.2: The Distinguished Name Hierarchy

The distinguished names should allow identification of the subject of a certificate. In the classical X.509 identity-certificate approach, this identification should have unambiguous legal interpretation. There are several problems with this requirement:

- Providing and validating sufficient details is difficult, expensive and intrusive.
- Some of the elements in the distinguished names may expose confidential details such as organization unit.
- The need to provide unique, legal identification may require including unique identifiers such as social security number, which may compromise privacy and allow identity theft.
- Relying parties will often consider only the easily understood keywords such as common name, but these are not sufficient for identification and non-repudiation.
- The same individual may receive multiple certificates from different organizations, e.g. company and community, with two seemingly unrelated distinguished names.
- The hierarchy among the keywords of the DN is not well defined, and efforts to define naming schemes were not successful.
- People are mobile; should your DN change when you change work? Should your public key?

In practice, distinguished names are rarely guaranteed to link to legally acceptable identifiers. Furthermore, to ensure uniqueness, issuers often place a random string as part of the DN.

Indeed, as of Version 2, X.509 certificates contain additional identifiers for the subject and issuer, besides their distinguished name. In particular, the X.509 certificate contains special fields for issuer and subject unique identifiers; see next section and in particular Figure 6.3.

7.4. X.509 Public Key Certificates

The core of X.509 is the definition of a public key certificate. An X.509 certificate consists of the following fields (see also in Figure 7.3):

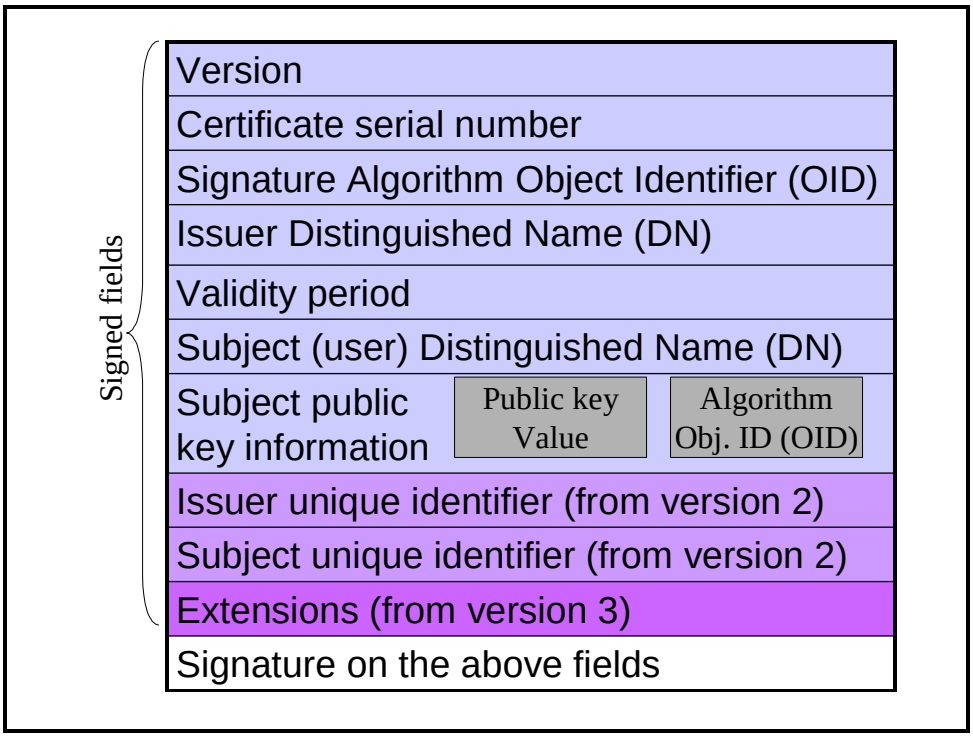

Figure 7.3: X.509 Certificate Structure

- *Version*: The version of the certificate used, e.g. X.509 v3 or v4.
- *Serial Number*: This is the serial number of the certificate. Normally, each issuer (typically Certificate Authority - CA) maintains a counter for the certificates it issues.
- *Signature algorithm identifier:* the *Object Identifier (OID)* identifying the algorithm used to sign the certificate. (We describe the OID below.)
- *Issuer*: This is the Distinguished Name (DN) of the issuer (CA).
- *Validity Period*: Indicates the period of time during which the certificate is valid.
- *Subject*: This is the Distinguished Name (DN) of the holder of the certified public key.
- *Subject public key information*: Contains the Object Identifier (OID) of the algorithm with which the subject public key may be used, and the value of the subject public key. (We describe the OID below.)
- Issuer and subject unique identifiers.
- *Extensions* field (from version 3 of X.509) are defined by particular PKI profile group (e.g. PKIX, WAP). In addition, X.509 itself defines some extensions. Each extension contains three components: an extension object identifier (OID; described below), a criticality flag, and an extension value field. We discuss the extensions mechanism below.
- Signature: This is the signature of the CA. The X.509 certificate is signed by the issuer (typically a Certificate Authority) to authenticate the binding between the subject (user's) name and the subject's public key, as well as to authenticate the other properties and attributes.

The most important extensions, standardized in X.509 (mostly in version 3 (1997)), are:

- Key and policy information. This includes different keys used by the subject (subject may use a different key for different actions: signing, encryption, key exchange and so on). It also includes Key usage (TLS/SSL, IPSEC, SET, etc.).
- Subject and issuer attributes, e.g. additional identifying fields (name, address, e-mail, phone number, etc.).
- Certification policy information
- Key usage: non-repudiation, key encryption, data encryption, certificate signing, certificate revocation list (CRL) signing, key agreement (Diffie-Hellman), or other signatures.
- Certificate Revocation List (CRL) numbers and revocation reasons
- CRL partitioning and delta CRLs.

Object identifiers (OID's) are global, unique identifiers, built as a sequence of numbers, with the top level numbers defined for the major international standardization bodies (mainly ITU-T and ISO), and lower level numbers assigned by the top level organizations, to national standardization bodies, specific companies and departments, or individuals. Object identifiers are part of the Abstract Syntax Notation (ASN.1) standard. There are many uses for them. The X.509 recommendation use object identifiers (OID) to uniquely identify algorithms (e.g. signature) and extensions.

The X.509 certificate is signed by the issuer (typically a Certificate Authority) to authenticate the binding between the subject (user's) name and the subject's public key, as well as to authenticate the other properties and attributes.

7.4.1. ASN.1 Certificate Encoding - BER, DER and XER

To be added (?)

7.5. X.509 Certificate Authorities (CA)

Certificate authorities are the central players in X.509. Certificate Authorities receive request with a public key, identify the subject of that key, and issue public key certificate, linking between the public key and the Distinguished Name (and possibly additional attributes) of the subject. If the relying party knows the public key of the CA and trusts it, then the relying party can use the subject's public key from the certificate, e.g. to validate a message sent and signed by the subject. Figure 7.4 illustrates this scenario.

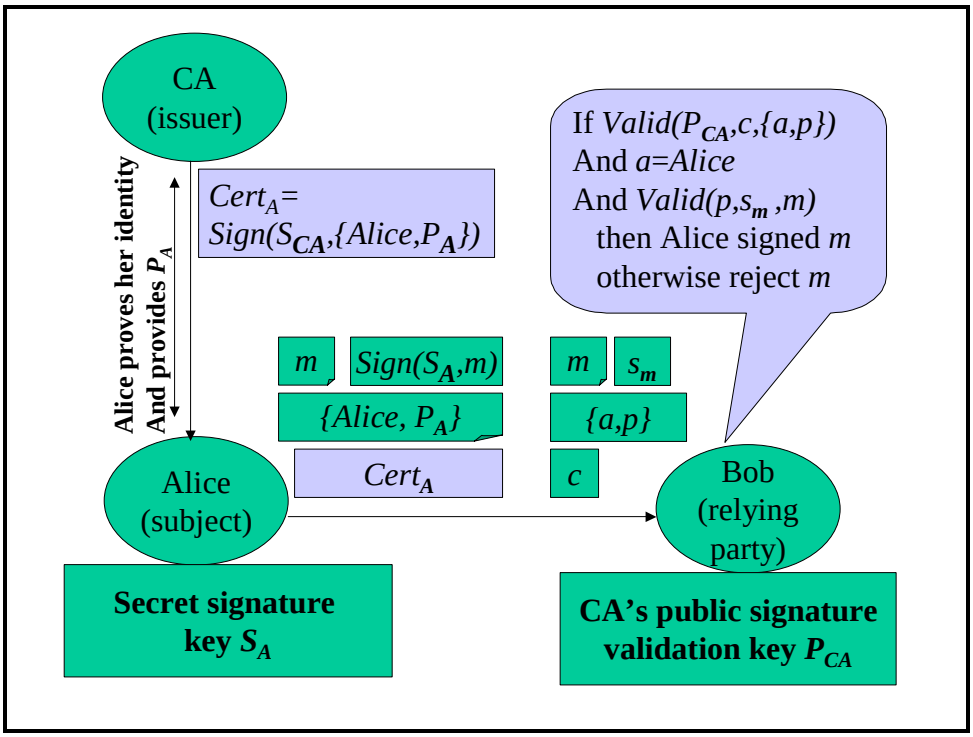

Figure 7.4: Bob relies on CA's certificate to validate Alice's signature

The X.509 framework expected there to be multiple certification authorities. X.509 expected the certification authorities to certify each other, e.g. that the a certification authority for the US will give a certificate to the certification authority of the state of New York. The certificate will be properly marked, and therefore a relying party which knows the public key of the US government, but not of the state of New York, will be able to validate certificates issued by the state of New York, by first receiving the certificate of the state of NY signed by the US government.

X.509 initially assumed that the relationship between certification authorities will form a hierarchy, corresponding to the hierarchy of keywords for the distinguished names (see Figure ??? above). The top level CA is called the *root CA*. The initial expectation was for single, global X.500 directory, supported by a corresponding single, global root CA and hierarchy of certification authorities. Figure 7.5 below illustrates this; an arrow going from one CA to another indicates that the first CA issues a certificate for the other.

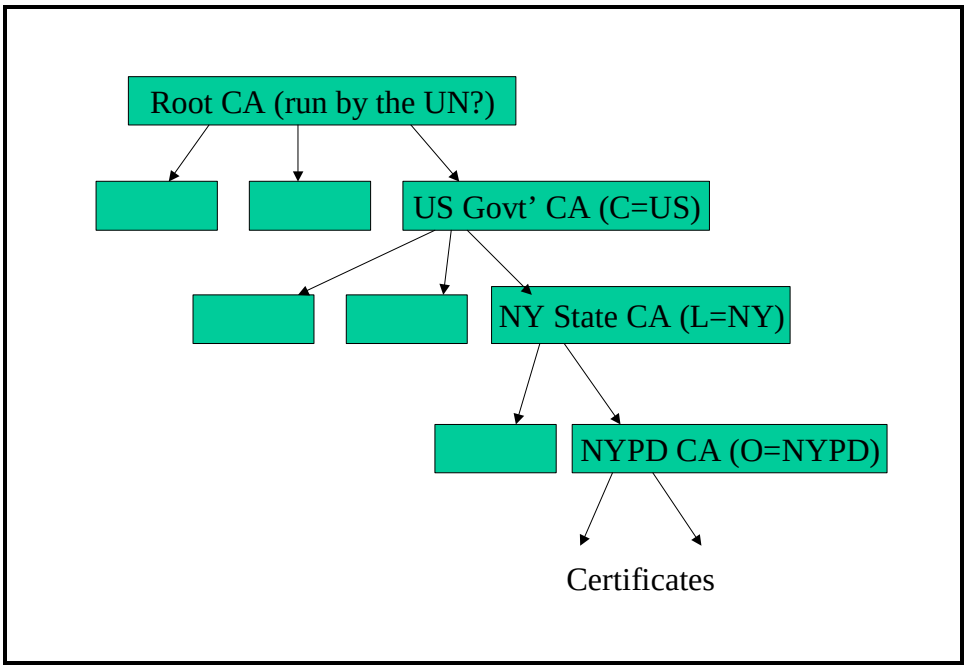

Figure 7.5: Global X.509 Certified Authorities Hierarchy

In reality, there is not one global root CA or hierarchy. Certification authorities do issue certificates to other certificate authorities, but this is usually a symmetric relation – each CA provides a certificate for the public key of the other CA; we call this *cross certification.* Cross certification often happens between the root CA's of two companies, allowing the servers of one company to identify signatures of employees of the other company; this typical cross-certification is shown as the blue arrow in Figure 7.6 below. Lower layer certification authorities may also cross certify directly, as shown in the orange arrow between the Japan CA of IBM and NTT. In this case, the lower CA may sometimes also issue a certificate for the root CA (or for a higher layer CA), as in the dashed purple arrows; this allows interoperability between the entire organizations (and not just the sub-organizations which have cross certified, in this example IBM Japan and NTT Japan).

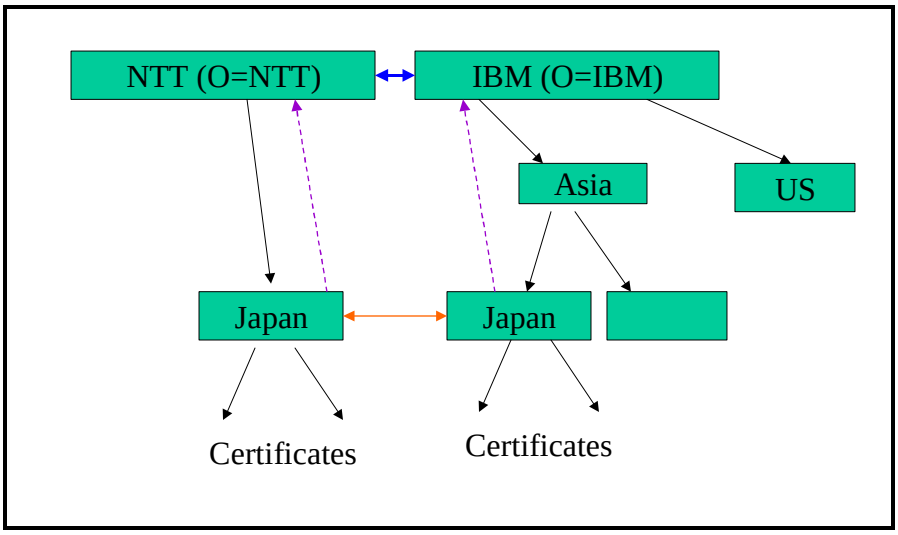

Figure 7.6: Cross Certification

At the present, there are a disappointing small number of X.509 certification authorities certifying each other. In particular, the widest deployment of X.509 in practice is in the certification of web servers for the SSL protocol (see next chapter). There are several certification authorities providing SSL certificates, however, they do not cross certify. Instead, SSL uses a *flat hierarchy*: the browser (acting as the relying party) simply has a list of multiple trusted certification authorities.

7.6. Issuing certificates and certification policies

The certification authority is responsible for the identification of Alice as the owner of a particular private key, corresponding to the public key in the certificate. This is a critical element in this design. If an adversary is able to receive a certificate with Alice's name, then he may impersonate as Alice; this is a serious threat known as *identity theft.* Furthermore, if it is *possible* for an adversary to impersonate as Alice, then Alice can dispute the fact that the public key belongs to her (i.e. that she generated it together with the corresponding private key), thereby disputing the signatures done using this key.

Therefore, any identification weakness or failure in the CA, may damage subjects of certificates and relying parties. The extent of the potential damage is essentially unpredictable, as it really depends on the future adoption of public key certificates and signatures as a valid identification method. The CA may be liable for these damages, e.g. if negligible.

The *certification policy* extension allows the CA to specify its identification mechanisms and to state limits on its liability[2], as well the purposes for which the certificate may be used. Unfortunately, the policy description is complex; X.509 splits the certificate policy into:

[2] There may be laws that limit the effect of such liability limiting statement.

- The well-defined *certificate policy* focus is on identification mechanisms. With cross-certified certificates, X.509 defines a complex mapping of policies, to provide coherent policy to the relying party.
- The legally binding *certificate policy statement (CPS)* document focus is on liability (and often on disclaimers). The CPS requires manual, legal interpretation. Therefore, the relying party can evaluate liability only of certificates with known, fixed CPS. Furthermore, the liability restrictions in the CPS are often extreme, which creates a difficulty in designing systems where an impersonation can cause significant damage, using identity-based certificates.

Furthermore, it is not easy to ensure secure identification. Possible attacks include:

1. Exposure of the Certificate Authority's private key, and using it to create false certificates. To prevent this risk, certificate authorities use methods such as:
    a. Invest substantially in physical and computer security.
    b. Screen their staff, motivate them properly, and audit procedures rigorously.
    c. Operate completely or mostly offline.
    d. Distribute the certification authority private key among several servers.
    e. Use an especially long private key, e.g. 2048 bits RSA key (where most applications use a key of 1024 bits). This prevents cryptoanalysis attack on the CA private key.
    f. Keep the CA private key in a dedicated, tamper resistant hardware unit, which uses it to sign certificates. The private key is never exposed outside the dedicated hardware.
2. Impersonation by presentation of false identity (name, etc.) when applying for the certificate. This is especially problematic when trying to allow remote certification, e.g. over the Internet.

Performing all identification directly, physically by the certificate authority is hardly practical. Most designs allow remote identification, in one of two ways:

1. By identifying the subjects via trusted agents of the CA, called *Registration Authorities*. The registration authority (RA) validates the identity of the subject (typically by looking at physical documents and evidences), and then sends the relevant identity information to the CA, with the subject's public key, over secure channel. See Figure 7.7.

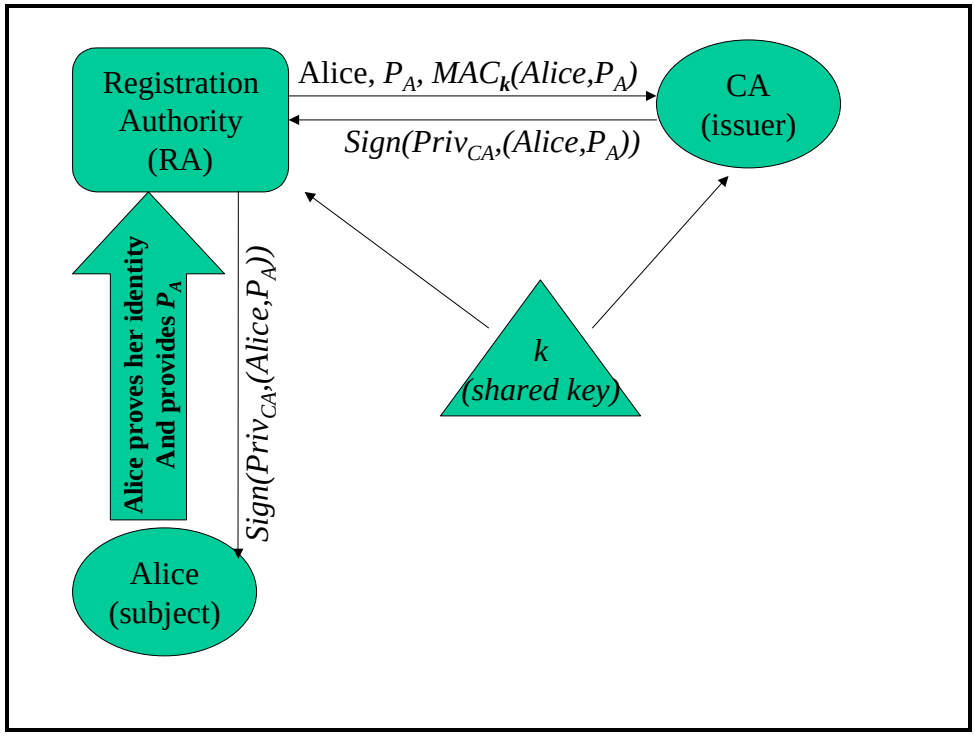

Figure 7.7: Issuing certificate with Registration Authority

2. By communication between the subject and the CA, in which the subject identifies by presenting some secret information to the CA. The CA may send the secret information to the subject in advance in some secure way, e.g. using traditional certified mail delivery. Alternatively, the `secret` may be some personal information such as spouse name, which the CA knows about the subject but hopes the adversary would not know. The subject should protect this communication in transit. To hide the secret, the subject typically encrypts it with the CA's public key. To ensure authenticity, the subject attaches a message authentication code (MAC), computed over the public key and any other details of the request, using the secret shared with the CA (or another secret which it sends encrypted to the CA). The client preferably also signs the request, proving possession of the secret key corresponding to the public key, as there are some attacks where the adversary receives a certificate with someone else's key on the adversary's identity. See Figure 7.8. We notice that this procedure is not specific to the issuing of identity certificates – the identity does not really play a major role here, except maybe to ensure the correct delivery of the initial secret by out-of-band means (e.g. traditional certified mail).

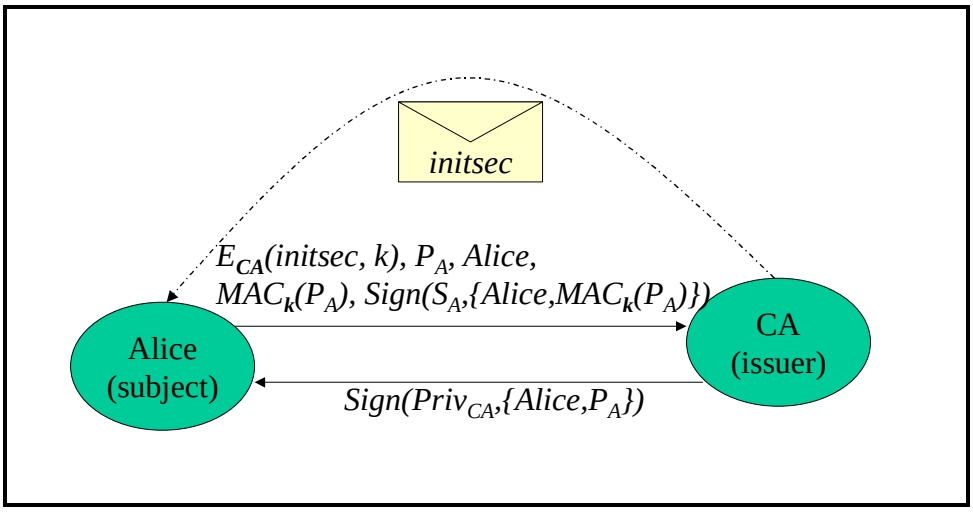

Figure 7.8: Issuing certificate by shared initial secret

7.7. Certificate Validity and Revocation – To be written

A certificate makes an assertion about the holder of the secret key (the subject of the certificate). For X.509 certificates, a major part of the assertion is the Distinguished Name, linking it to a given public key. There are other assertions in certificates; specifically, in X.509 certificates, some of the other important assertions include:

- CA indicator, specifying

The CA tries to validate that statement when issuing the certificate.

As we discussed, it is hard to anticipate future e-commerce applications, and therefore consider the potential liability of each certificate. This is a concern to both CA and certificate subject (key owner). One way to limit the liability risk is to limit the validity time of the certificate; thereby, the CA and subject can at least limit the risk to this period, which allows better projection of the actual financial risk. The *Validity period* field of X.509 certificates allows the CA to define such a validity period. By limiting certificates to short periods, we can limit the risk of

X.509 defines an additional mechanism, called

7.8. Identrus – to be done

7.9. Exercises

1. Consider Figure 7.8: Issuing certificate by shared initial secret.
    1. Motivate the use of a separate key *k* rather than authenticating using the pre-shared initial secret *initsec*. Hint: consider the case where *initsec* is a four digit PIN.