Universidad de Oviedo

Departamento de Informática

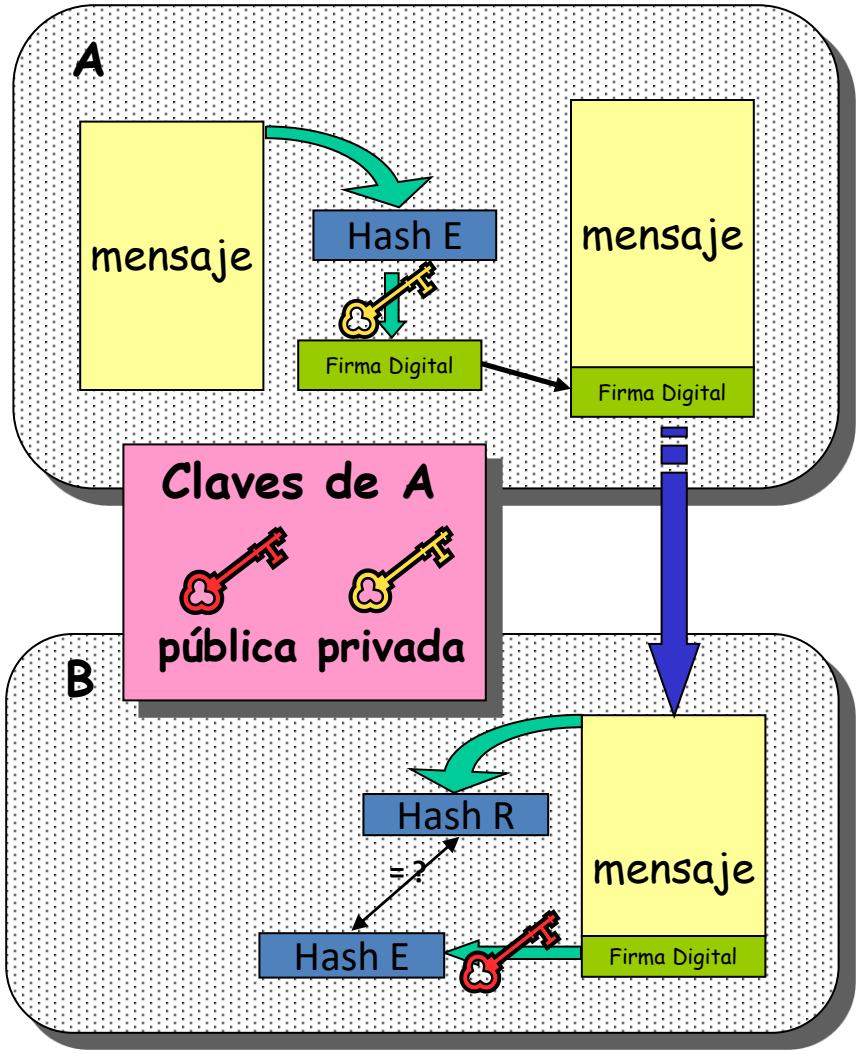
Campus de Gijón

Certificados Digitales

Presentación

Daniel F. García

Necesidad de los certificados digitales (1)



1. A Calcula el hash del mensaje (Hash E)
   Valor de 128 bits basado en el mensaje
2. A encripta el hash usando su clave privada
   El hash encriptado es la Firma Digital
3. A envía en mensaje firmado a B
4. B calcula el hash del mensaje (Hash R)
5. B Desencripta la firma digital de A
   Usa la clave pública de A **¿es fiable?**
6. COMPARACIÓN: ¿ Hash R == Hash E ?

   Si son IGUALES, entonces el mensaje
   - Fue generado por A
   - Y no fue modificado

Necesidad de los certificados digitales (2)

La firma digital de A es útil para B si se cumplen 2 condiciones

1. La clave privada de A no está comprometida → Solo la conoce A
2. B tiene acceso a la clave pública de A de un modo seguro

¿ **Cómo puede B estar seguro de que la clave pública de A es realmente la clave pública de A y no la de un impostor** ?

Una tercera parte establece la asociación:

Clave pública ⟷ Identidad de su propietario

Ambos, A y B, confían en esta tercera parte

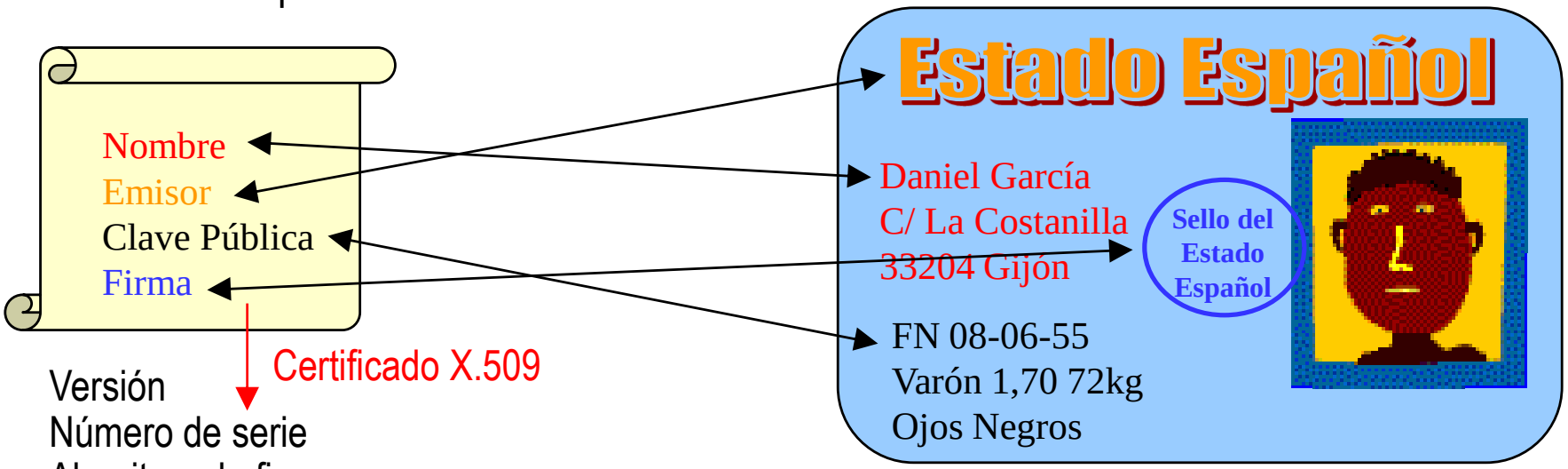
A esta tercera parte se la denomina: **Autoridad de Certificación (AC)**

Las ACs crean, distribuyen, gestionan, …

Certificados Digitales, que contienen la asociación < Clave pública – Identidad >

Certificados digitales: concepto

Certificado digital: documento electrónico que asocia una clave pública con una identidad
Y son firmados por el emisor



Certificado X.509

Versión
Número de serie
Algoritmo de firma
Emisor (AC)
Validez: No antes de / No después de
Usuario
Clave pública (Algoritmo + Clave)
Identificador único de emisor (opcional)
Identificador único de usuario (opcional)
Extensiones (opcional)
Firma de AC (Algoritmo + Firma)

Introducidos en la Versión 2

Introducidas en la Versión 3

Ejemplo de Certificado X.509 (De Usuario)

```
Certificate:
Data:
    Version: 1 (0x0)
    Serial Number: 7829 (0x1e95)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
            OU=Certification Services Division,
            CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    Validity
        Not Before: Jul  9 16:04:02 1998 GMT
        Not After : Jul  9 16:04:02 1999 GMT
    Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
             OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public Key: (1024 bit)
            Modulus (1024 bit):
                00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
                66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
                70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
                16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
                c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
                8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
                d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
                e8:35:1c:9e:27:52:7e:41:8f
            Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
    93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
    92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
    ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
    d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
    0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
    5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
    8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
    68:9f
```

Autoridad Certificadora

Usuario

Clave Pública del Usuario

Firma Digital de la Autoridad Certificadora

Ejemplo de Certificado X.509 (RAIZ)

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
            OU=Certification Services Division,
            CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    Validity
        Not Before: Aug  1 00:00:00 1996 GMT
        Not After : Dec 31 23:59:59 2020 GMT
    Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
             OU=Certification Services Division,
             CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
        RSA Public Key: (1024 bit)
            Modulus (1024 bit):
                00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
                68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
                85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
                6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
                6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
                29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
                6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
                5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
                3a:c2:b5:66:22:12:d6:87:0d
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints: critical
            CA:TRUE
Signature Algorithm: md5WithRSAEncryption
    07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
    a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
    3e:59:43:7d:4f:95:3d:a1:8b:b7:0b:62:98:7a:75:8a:dd:88:
    4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
    8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
    e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
    b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
    70:47
```

Autoridad Certificadora

Usuario Certificado

Plazo de validez Muy grande

Tipos de certificados (1)

Según el nivel de Verificación de los datos que incluye

Certificados de Clase 1

La AC sólo verifica el nombre y la dirección de correo electrónico del titular

Son los certificados más fáciles de obtener pues involucran pocas verificaciones de los datos que figuran en él

Certificados de Clase 2

La AC comprueba además :

- El documento de identidad o permiso de conducir que incluya fotografía
- El número de la Seguridad Social
- La fecha de nacimiento

Certificados de Clase 3

La AC añade a las comprobaciones de la Clase 2 la verificación de crédito de la persona o empresa mediante un servicio del tipo Equifax

Certificados de Clase 4

La AC añade a todas las comprobaciones anteriores la verificación del cargo o la posición de una persona dentro de una organización

Tipos de certificados (2)

Según la Finalidad para la que se emite el certificado

Certificados SSL para cliente

Usados para identificar a un cliente ante un servidor en comunicaciones basadas en el protocolo SSL /TLS (Secure Sockets Layer / Transport Layer Security)
Generalmente se expiden a una persona física

Certificados SSL para servidor

Usados para identificar a un servidor ante un cliente en comunicaciones basadas en SSL/TLS

Certificados S/MIME

Usados para servicios de correo electrónico firmado y cifrado
Generalmente se expiden a una persona física

Certificados para firmar programas

Usados para identificar al autor de programas que se descargan o ejecutan desde la red

Certificados para ACs (Autoridades Certificadoras)

Usados por el software cliente para determinar si puede confiar en un certificado
Accede al certificado de la AC y comprueba que ésta es de confianza

Más Clasificaciones

Pero la variedad de las clasificaciones es INFINITA ⋯ Ejemplos:

La Fábrica Nacional de Moneda y Timbre https://www.cert.fnmt.es/

*Proyecto CERES (**CER**tificación **ES**pañola)*

Emite certificados de cuatro clases: **FNMT Clase 2 CA**

- Persona física (También conocidos como Certificados de Usuario)
- De Representante Administradores, Personas jurídicas, Entidades sin personalidad jurídica
- Administración Pública
- De Componente Certificados de Servidor SSL

El banco de España https://pki.bde.es/certificados.htm

Emite certificados para cada el envío de información firmada y cifrada

Agencia Notarial de Certificación (ANCERT) https://www.ancert.com/

Emite certificados personales, corporativos, de servidor, etc.

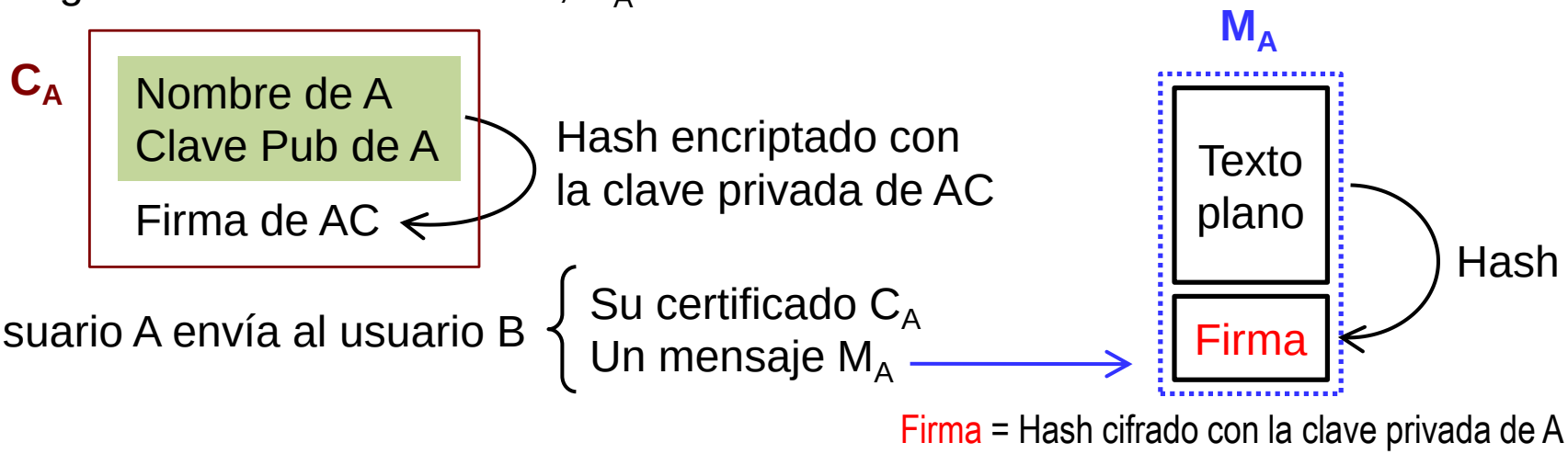
Ministerio del Interior https://www.dnielectronico.es/

Emite certificados relacionados con el DNI electrónico

Uso elemental de un certificado

El usuario A solicita a la Autoridad Certificadora (AC) un certificado digital C_A

La AC genera el certificado de A, C_A:



El usuario B tiene que usar la clave pública de A para verificar la firma de M_A

La clave pública de A está en C_A el certificado de A

Para confiar en C_A

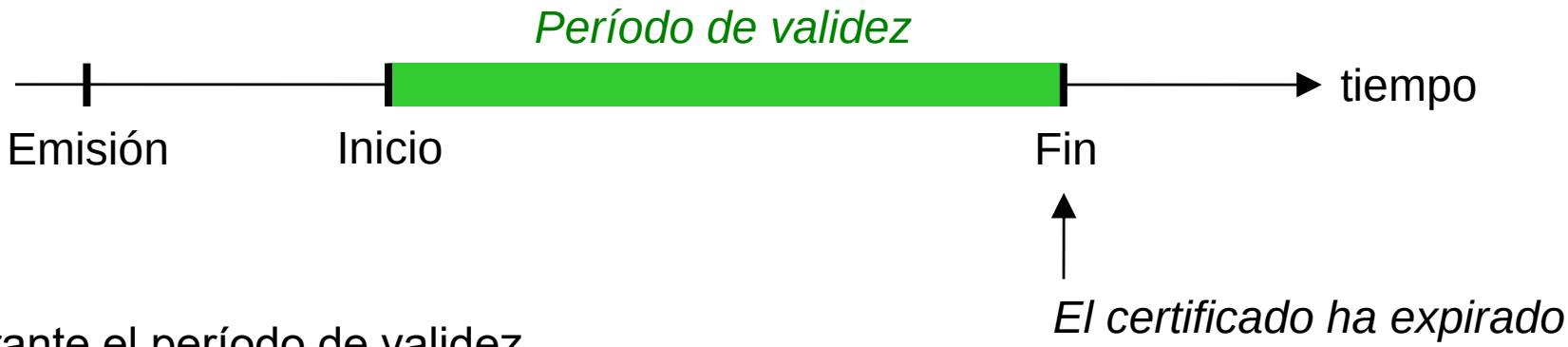
B obtiene la clave pública de la AC y verifica que la firma de AC en C_A es correcta

Si es correcta solo la AC ha podido emitir el certificado de A

Pues se supone que la clave privada de AC no está comprometida

Si B confía en la AC asume que el certificado es auténtico y debe comprobar si es válido antes de usarlo

Estados de un certificado



Durante el período de validez …
En ocasiones puede ser necesario:

Revocar el certificado

- Porque la clave privada asociada al certificado se ha visto comprometida
- Porque se han producido cambios en los datos asociados al certificado

Suspender el certificado

La suspensión es una revocación temporal

Se hacen las mismas actuaciones que en una revocación pero son reversibles

Las ACs deben mantener "Listas de Revocación de Certificados" (***CRL**: Certificate Revocation List*) incluyendo referencias a todos los certificados que han revocado

Descripción de certificados X.509 (1)

Un certificado X.509 se puede describir formalmente usando ASN.1

ASN.1 == Abstract Syntax Notation number One

ASN.1 está especificado en los estándares X.680, X.681, X.682 y X.683 de la ITU-T

https://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx

ASN.1 es una notación formal para describir datos independientemente de:
el lenguaje de implementación y la representación física

Proporciona **tipos primitivos**:

INTEGER, BOOLEAN, IA5String, UniversalString, BIT STRING, etc.

Proporciona **tipos construidos**:

SEQUENCE (≈estructura), SEQUENCE OF (≈lista), SET, SET OF, etc.

A partir de ellos se define cualquier información simple o compleja

ASN.1 solo cubre los aspectos "estructurales" de la información

No hay operadores para manipular o hacer cálculos con los valores definidos

Por tanto ASN.1 **NO** es un lenguaje de programación

Pero una descripción ASN.1 se puede mapear a C/C++ java o XML fácilmente

Descripción de certificados X.509 (2)

Descripción de un certificado X.509 en ASN.1:

RFC-5280 (Mayo 2008)
https://datatracker.ietf.org/doc/pdf/rfc5280.pdf

```
Certificate ::= SEQUENCE {
    tbsCertificate TBSCertificate,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue BIT STRING }

TBSCertificate ::= SEQUENCE {
    version [0] EXPLICIT Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,
    validity Validity,
    subject Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL,
    -- If present, version MUST be v2 or v3
    subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,
    -- If present, version MUST be v2 or v3
    extensions [3] EXPLICIT Extensions OPTIONAL
    -- If present, version MUST be v3 }

Version ::= INTEGER { v1(0), v2(1), v3(2) }
```

Descripción de certificados X.509 (3)

```
CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
    notBefore Time,
    notAfter Time }

Time ::= CHOICE {
    utcTime UTCTime,
    generalTime GeneralizedTime }

UniqueIdentifier ::= BIT STRING

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT STRING }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnID OBJECT IDENTIFIER,
    critical BOOLEAN DEFAULT FALSE,
    extnValue OCTET STRING
    -- contains the DER encoding of an ASN.1 value
    -- corresponding to the extension type identified
    -- by extnID }
```

```
AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL }
```

Codificación de certificados X.509 (1)

Los certificados X.509 se describen en ASN.1 y …

El ASN.1 se codifica en varios formatos descritos en el estándar X.690 de la ITU-T

https://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

Formatos → { BER (*Basic Encoding Rules*) | CER (*Canonical Encoding Rules*) | DER (*Distinguished Encoding Rules*) }

Formato BER (*Basic Encoding Rules*)

Formato que incluye mecanismos de **auto-descripción** y **auto-delimitación** de los datos

Generalmente incluye los 4 campos siguientes de varios octetos:

Called TLV Encoding (Type-Length-Value)

Type – Codifica el tipo de dato: *Boolean*, *Integer*, *Enumerated*, …

Hay 2 formas de codificar la longitud { *Definite* → *Length* codifica la longitud en bytes | *Indefinite* → *Length* indica que se usa *End-of-Contents* }

Value – Codifica el valor del dato

End-of-Contents – Marca el final del contenido

Codificación de certificados X.509 (2)

Formato BER (*Basic Encoding Rules*)

- Permite múltiples codificaciones alternativas para un mismo dato
  - Ej.: Un booleano true se codifica como un byte ≠ 0 → 255 codificaciones posibles

Formato CER (*Canonical Encoding Rules*)

- Solo permite una de las codificaciones BER
- Usa la forma ***indefinite*** para algunos casos especificados con precisión
  - Generalmente requiere menos metadatos para valores codificados grandes
  - Apropiado si los valores no caben fácilmente en memoria ó
    hay que codificar y transmitir parte de un valor antes de disponer de todo el valor

Formato DER (*Distinguished Encoding Rules*)

- Solo permite una de las codificaciones BER
- Siempre usa la forma ***definite*** empleando la codificación más corta posible
  - Generalmente requiere menos metadatos para valores codificados pequeños
  - Apropiado si los valores caben bien en memoria y
    hay que saltar rápidamente algunos valores anidados

Codificación de certificados X.509 (3)

La codificación Base64

Consiste en codificar el certificado en un sistema de numeración posicional en base 64

- Se codifican grupos de 6 bits (números 0-63) en los caracteres ASCII imprimibles A-Z a-z 0-9 + /
- Los dos últimos caracteres pueden diferir entre diversas codificaciones base 64
- El símbolo = se usa como un sufijo especial

La codificación se encapsula entre dos cadenas (BEGIN y END)

La codificación se organiza en líneas de 64 caracteres separadas por CR-LF, excepto la última

```
-----BEGIN CERTIFICATE-----
MIIBuTCCASKgAwIBAgIQNdNhtuV5GbNHYZsf+LvM0zANBgkqhkiG9w0BAQUFADAb
MRkwFwYDVQQDExBFZGlkZXYgU21va2VUZXN0MB4XDTA4MTExMjE5NTEzNVoXDTM5
MTIzMTIzNTk1OVowGzEZMBcGA1UEAxMQRWRpZGV2IFNtb2tlVGVzdDCBnzANBgkq
hkiG9w0BAQEFAAOBjQAwgYkCgYEAm6zGzqxejwswWTNLcSsa7P8xqODspX9VQBuq
5W1RoTgQ0LNR64+7ywLjH8+wrb/lB6QV7s2SFUiWDeduVesvMJkWtZ5zzQyl3iUa
CBpT4S5AaO3/wkYQSKdI108pXH7Aue0e/ZOwgEEX1N6OaPQn7AmAB4uq1h+ffw+r
RKNHqnsCAwEAATANBgkqhkiG9w0BAQUFAAOBgQCZmj+pgRsN6HpoICawK3XXNAmi
cgfQkailX9akIjD3xSCwEQx4nG6tZjTz30u4NoSffW7pch58SxuZQDqW5NsJcQNq
Ngo/dMoqqpXdi2/0BYEcJ8pjsngrFm+fM2BnyGpXH7aWuKsWjVFGlWlF+yi8I35Q
8wFJt2Z/XGA7WWDjvw==
-----END CERTIFICATE-----
```

La 1ª aplicación de la codificación Base64 fue en el protocolo **PEM** (*Privacy Enhanced Mail*)

Mensajes criptográficos (muchos incluyen certificados)

El estándar PKCS#7 o CMS

Describe la sintaxis para representar datos (mensajes) relacionados con operaciones criptográficas
- Ej. Firmas digitales, claves encapsuladas (*enveloped*), datos cifrados, etc.

La descripción de los mensajes se basa en ASN.1
Los valores generados por este estándar serán codificados en BER

RFC-2315 Cryptographic Message Syntax v1.5 Marzo 1998
https://datatracker.ietf.org/doc/pdf/rfc2315.pdf

Evolución RFCs: 2630 (Junio1999) → 3369 (Agosto 2002) → 3852 (Julio 2004)

RFC-5652 Cryptographic Message Syntax (CMS) Septiembre 2009
https://datatracker.ietf.org/doc/pdf/rfc5652.pdf

CMS (*Cryptographic Message Syntax*) se utiliza como componente de otros protocolos:

S/MIME, PKCS#12, Digital timestamping protcol (RFC 3161)

Exportación e Importación de certificados

El estándar PKCS#12

Define una sintaxis para el intercambio de información personal que permite guardar múltiples objetos criptográficos en un mismo fichero

---

PKCS#12 v1.0: Personal Information Exchange Syntax 24-Junio-1999

PKCS#12 v1.0: Technical Corrigendum 1 4-Febrero-2000

PKCS#12 v1.1: Personal Information Exchange Syntax 27-Octubre-2012

RFC 7292 PKCS#12: Personal Information Exchange Syntax v1.1 Julio 2014

https://datatracker.ietf.org/doc/pdf/rfc7292

Se usa para agrupar → { Un certificado X.509 con su clave privada; Todos los elementos de una cadena de certificados }

Comúnmente utilizado para exportar/importar claves privadas, certificados, etc.

Un fichero PKCS#12 puede ser cifrado y firmado

Los contenedores internos (*SafeBags*) pueden cifrarse y firmarse → Uso → { Claves; Certificados; Listas de revocación; Secretos (definible) }

Almacenamiento de certificados en ficheros

Las codificaciones se almacenan en ficheros con diversas extensiones:

Sin incluir la clave privada

DER → .cer ó .der ó .crt

.cer es usada por Microsoft y .crt por los sistemas Unix

PKCS#7 → .p7b

BASE 64 → .cer ó .pem

.cer es usada por Microsoft y .pem se asocia más a los sistemas de e-mail

Incluyendo la clave privada

PKCS#12 → .pfx ó .p12

.pfx es usada por Microsoft y .p12 fue propuesta por Netscape