

Seguretat Informàtica (SI)

Tema 3. Infraestructura PKI

Davide Careglio

Fuentes: Jordi Nin, "Certificates", Computer Security, 2014
Francisco Jordan, "PKI and Certificates", Computer Security, 2018
Jaime Delgado, "Certificates", Computer Security, 2017

Temario

- ▶ Tema 1. Introducción
- ▶ Tema 2. Criptografía
- ▶ **Tema 3. Infraestructura PKI**
- ▶ Tema 4. Seguridad en la red
- ▶ Tema 5. Seguridad en las aplicaciones
- ▶ Tema 6. Seguridad en los sistemas operativos
- ▶ Tema 7. Análisis forense

Tema 3. Índice

- ▶ 3.1 - Conceptos básicos
- ▶ 3.2 - Componentes de una PKI
- ▶ 3.3 - Modelos de confianza (trust models)
- ▶ 3.4 - Certificados X.509
- ▶ 3.5 - Public-Key Cryptography Standards (PKCS)

3.1 - Conceptos básicos

- ▶ La criptografía publica permita el intercambio de mensajes *secretos y auténticos*
 - ▶ Secretos: solo el destino con la clave privada sabe descifrarlos
 - ▶ Auténticos: son los mensajes originales del origen, i.e., se garantiza que no han sido modificado
- ▶ Basta conocer la clave publica del destino para transmitir estos mensajes
- ▶ Pero ...

3.1 - Conceptos básicos

- ▶ La criptografía publica permita el intercambio de mensajes *secretos y auténticos*
 - ▶ Secretos: solo el destino con la clave privada sabe descifrarlos
 - ▶ Auténticos: son los mensajes originales del origen, i.e., se garantiza que no han sido modificado
- ▶ Basta conocer la clave publica del destino para transmitir estos mensajes
- ▶ Pero ...
- ▶ ¿Como podemos estar seguros que la clave pública que hemos usado es realmente del destino que queremos?
- ▶ Alguien podría haber manipulado esta clave o haberla sustituido por otra

3.1 - Conceptos básicos

Primera propuesta

- ▶ En el 1976, Diffie-Hellman propone el uso de un único repositorio seguro que almacene todas las claves públicas
- ▶ Problemas

3.1 - Conceptos básicos

Primera propuesta

- ▶ En el 1976, Diffie-Hellman propone el uso de un único repositorio seguro que almacene todas las claves públicas
- ▶ Problemas
 - ▶ Prestaciones muy bajas (cuello de botella)
 - ▶ Requisitos de seguridad extremos (concentración en un único punto)
 - ▶ Necesidad de backup (replica de los datos) que requiere la transmisión de nuevos datos constantemente: posible manipulación, pérdida, inserción, etc.

3.1 - Conceptos básicos

Segunda propuesta

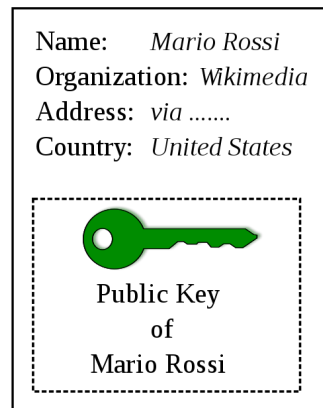
- ▶ En el 1978, Kohnfelder consideró la idea de usar autoridades de confianza centralizadas
- ▶ Y propuso la creación de un conjunto de registros de datos firmados (certificados) que permitan un sistema de distribución de claves públicas de confianza
- ▶ Un certificado digital es una estructura de datos que contiene
 - ▶ La identidad del propietario (puede ser persona física, organismo o empresa)
 - ▶ Su clave pública
 - ▶ La firma digital de una entidad de confianza que certifica que la clave pública es de este propietario que luego tendrá en su poder la clave privada correspondiente

3.1 - Conceptos básicos

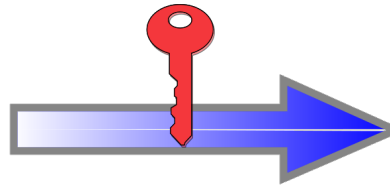
Segunda propuesta

- ▶ La certificación que la identidad del propietario y de la clave publica es correcta recae en una autoridad de confianza

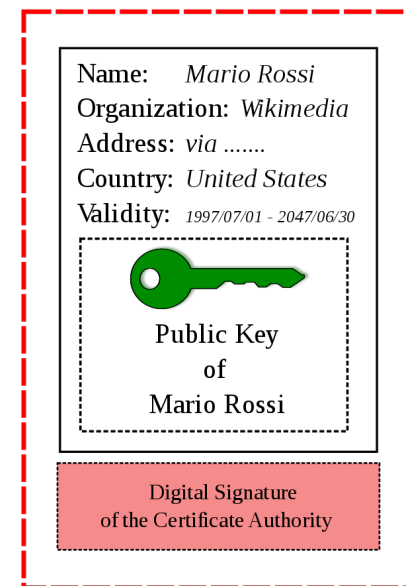
Identity Information and
Public Key of Mario Rossi



Certificate Authority
verifies the identity of Mario Rossi
and encrypts with its Private Key



Certificate of Mario Rossi



Digitally Signed by
Certificate Authority

- ▶ De forma que la confianza en esta autoridad se “extiende” a la clave publica
 - ▶ i.e., la clave publica es de confianza

Fuente imagen: Wikipedia

Tema 3. Índice

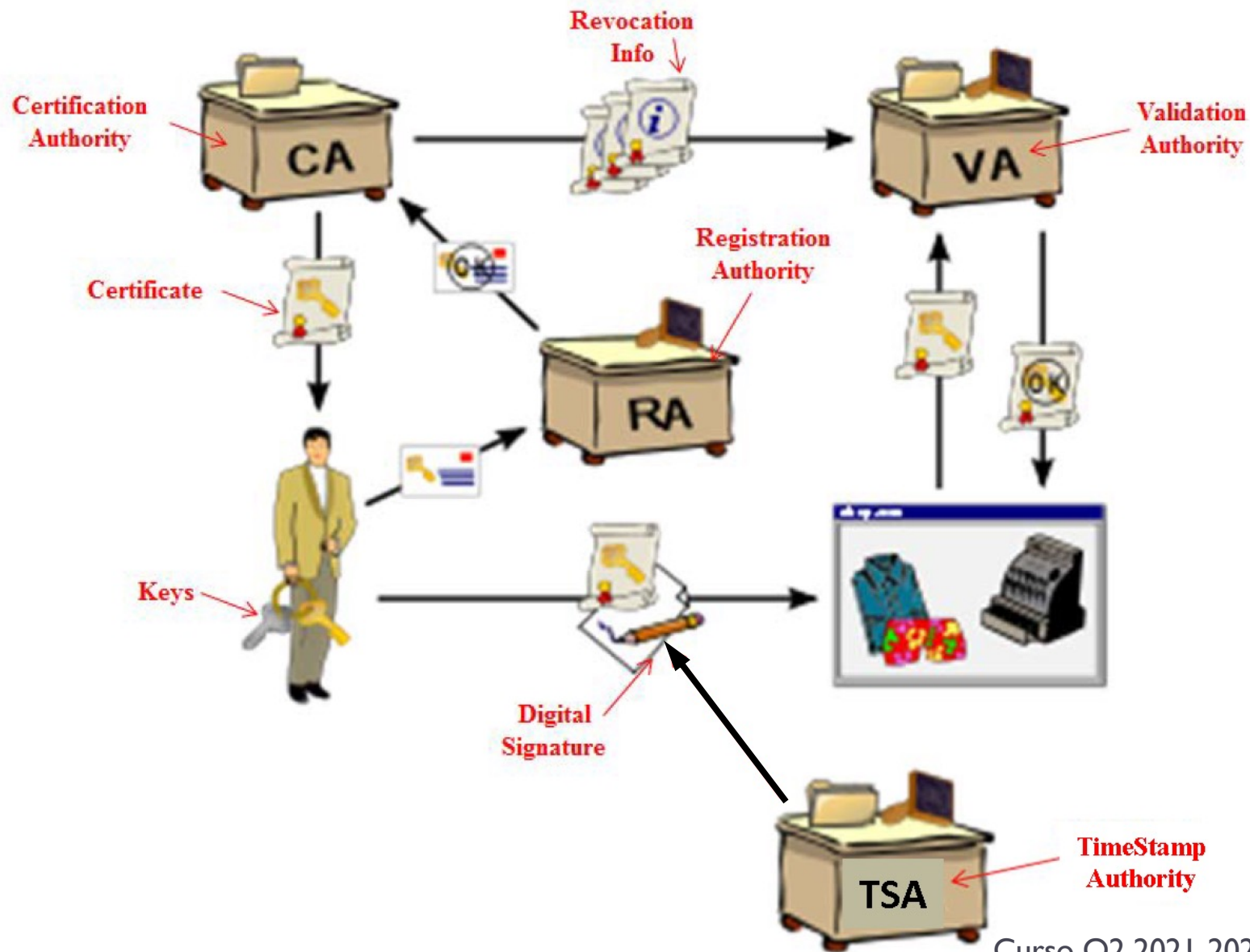
- ▶ 3.1 - Conceptos básicos
- ▶ 3.2 - Componentes de una PKI
- ▶ 3.3 - Modelos de confianza (trust models)
- ▶ 3.4 - Certificados X.509
- ▶ 3.5 - Public-Key Cryptography Standards (PKCS)

3.2 – Componentes de una PKI

- ▶ Una Public Key Infrastructure (PKI) es el conjunto de computadoras, software, individuos, políticas y procedimientos necesarios para crear y administrar los certificados digitales basados en criptografía de clave pública
- ▶ El objetivo es la gestión eficiente y confiable de los certificados digitales y sus claves criptográficas

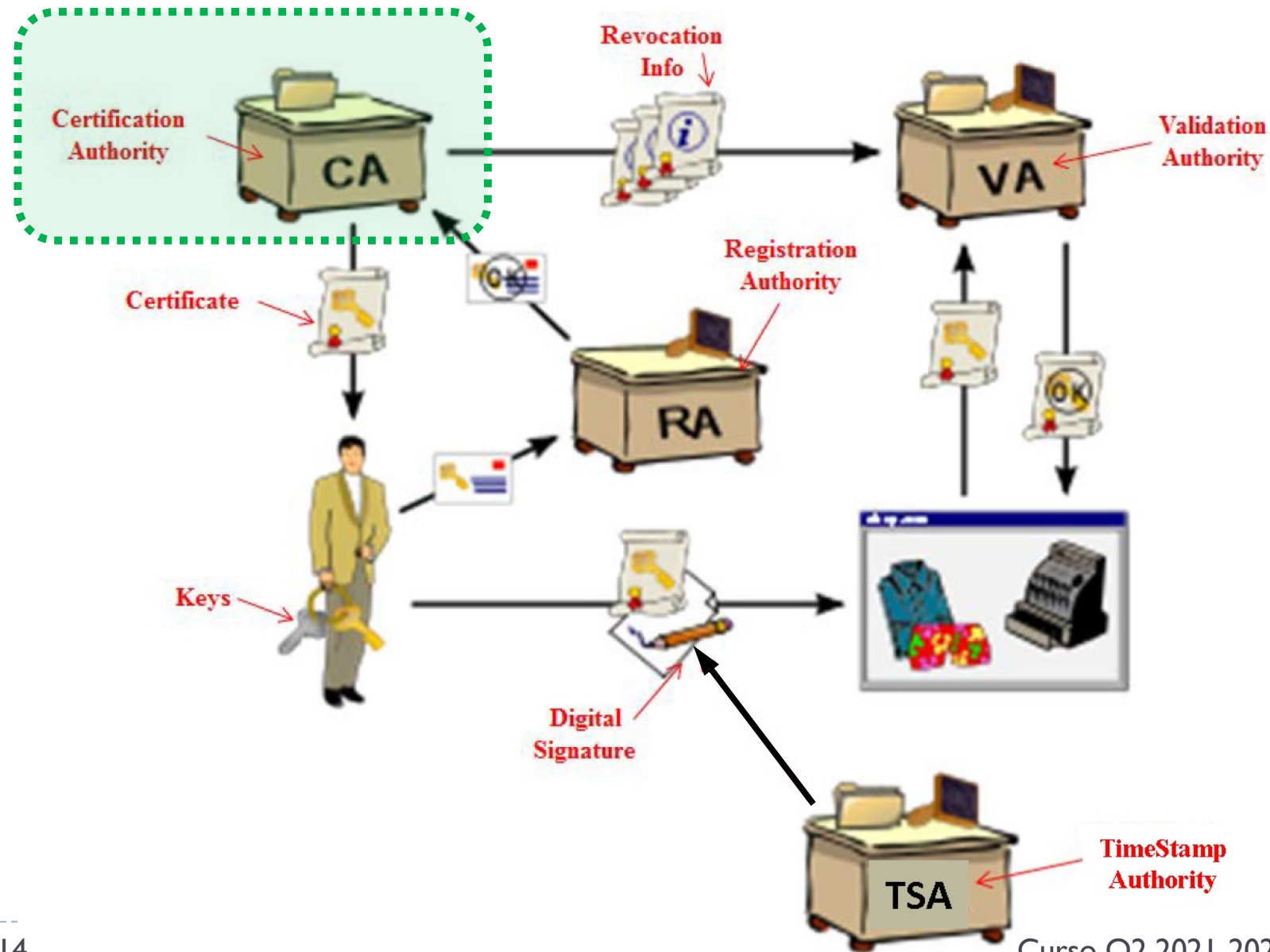
3.2 – Componentes de una PKI

Estructura



3.2 – Componentes de una PKI

Estructura



3.2 – Componentes de una PKI

Certification Authority (CA)

- ▶ Se necesitan las CA
 - ▶ Son las Trusted Third Party (TTP)
 - ▶ Emiten los certificados digitales
 - ▶ Opcionalmente pueden generar las claves públicas/privadas
 - ▶ Deben mantener sus propias claves públicas/privadas muy protegidas (recordar que las CA firman digitalmente los certificados)
 - ▶ Típicamente guardadas en hardware criptográficos sin conectividad a la red y con políticas de acceso físico muy restrictivas
- ▶ Una PKI puede tener más de una CA
- ▶ Pero...

3.2 – Componentes de una PKI

Certification Authority (CA)

- ▶ Se necesitan las CA
 - ▶ Son las Trusted Third Party (TTP)
 - ▶ Emiten los certificados digitales
 - ▶ Opcionalmente pueden generar las claves publicas/privadas
 - ▶ Deben mantener sus propias claves publicas/privadas muy protegidas (recordar que las CA firman digitalmente los certificados)
 - ▶ Típicamente guardadas en hardware criptográficos sin conectividad a la red y con políticas de acceso físico muy restrictivas
- ▶ Una PKI puede tener más de una CA
- ▶ Pero ... ¿Quien certifica que una CA es de confianza?
 - ▶ Otras CA
 - ▶ La misma CA

} Veremos los modelos más adelante

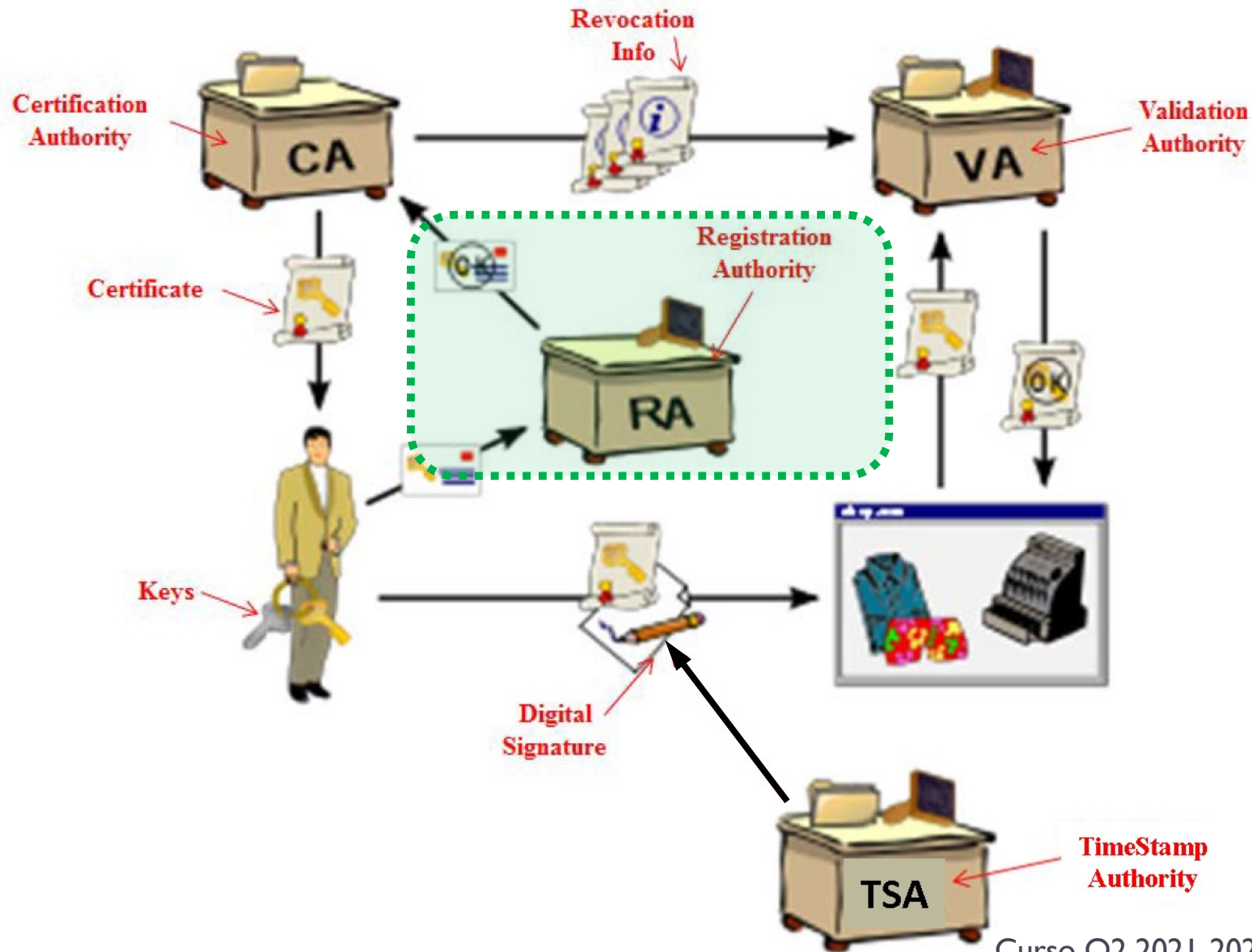
3.2 – Componentes de una PKI

Ejemplos de CA en España

- ▶ Dirección General de la Policía (DNle)
- ▶ Fábrica Nacional de Moneda y Timbre (FNMT)
- ▶ Agència Catalana de Certificació (CATCert)
- ▶ AC Camerfirma
- ▶ Firma profesional
- ▶ BANESTO
- ▶ Autoridad de Certificación de la Abogacía (ACA)
- ▶ ...

3.2 – Componentes de una PKI

Estructura



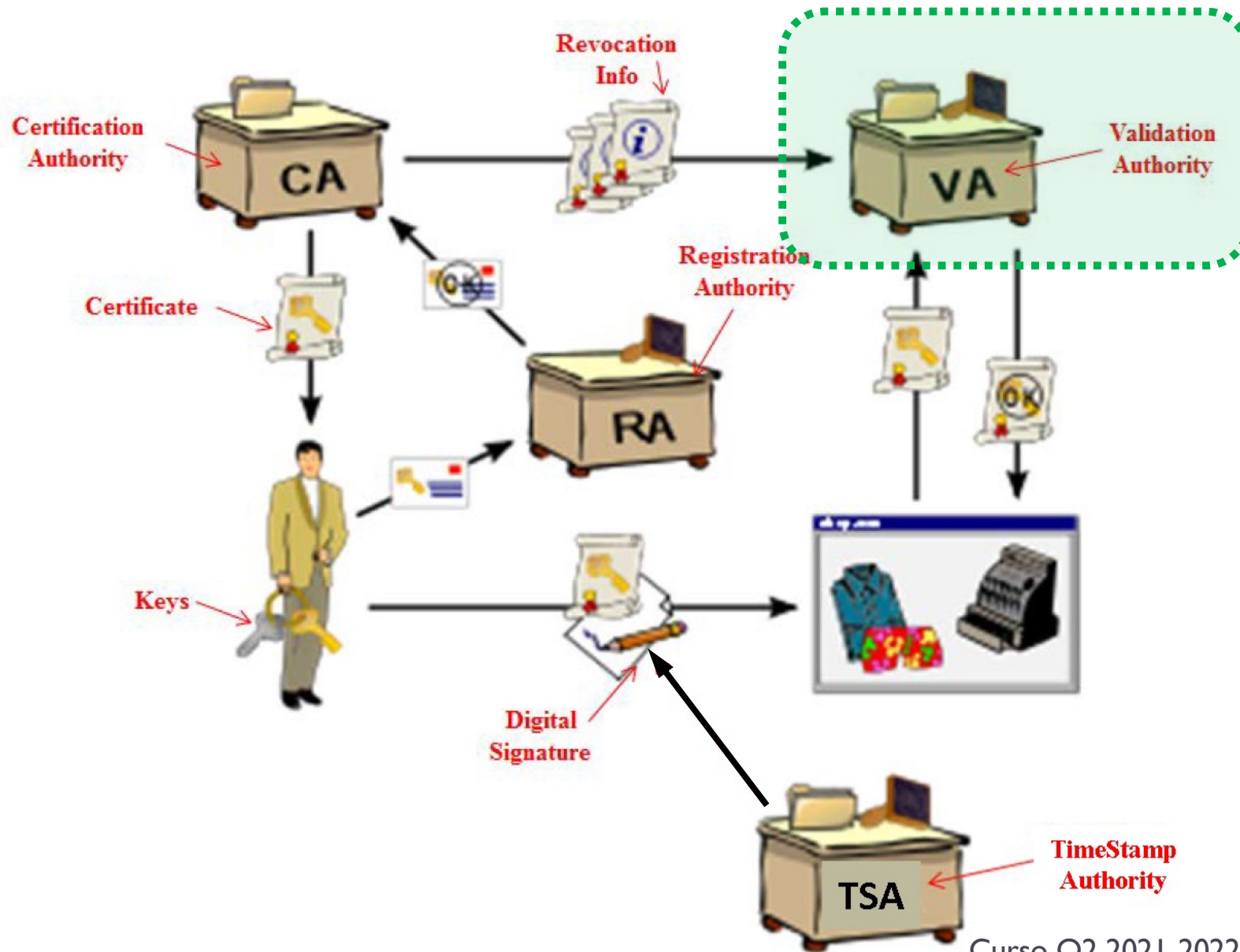
3.2 – Componentes de una PKI

Registration Authority (RA)

- ▶ Una RA se encarga de verificar la relación entre la clave publica y la identidad del propietario
- ▶ Por ejemplo en España, la RA de la FNMT es la Agencia Tributaria local
- ▶ La RA es un componente opcional de una PKI
 - ▶ Sirve para que una CA no se ocupe de temas administrativos
 - ▶ Si no hay una RA, la CA hace también de RA

3.2 – Componentes de una PKI

Estructura



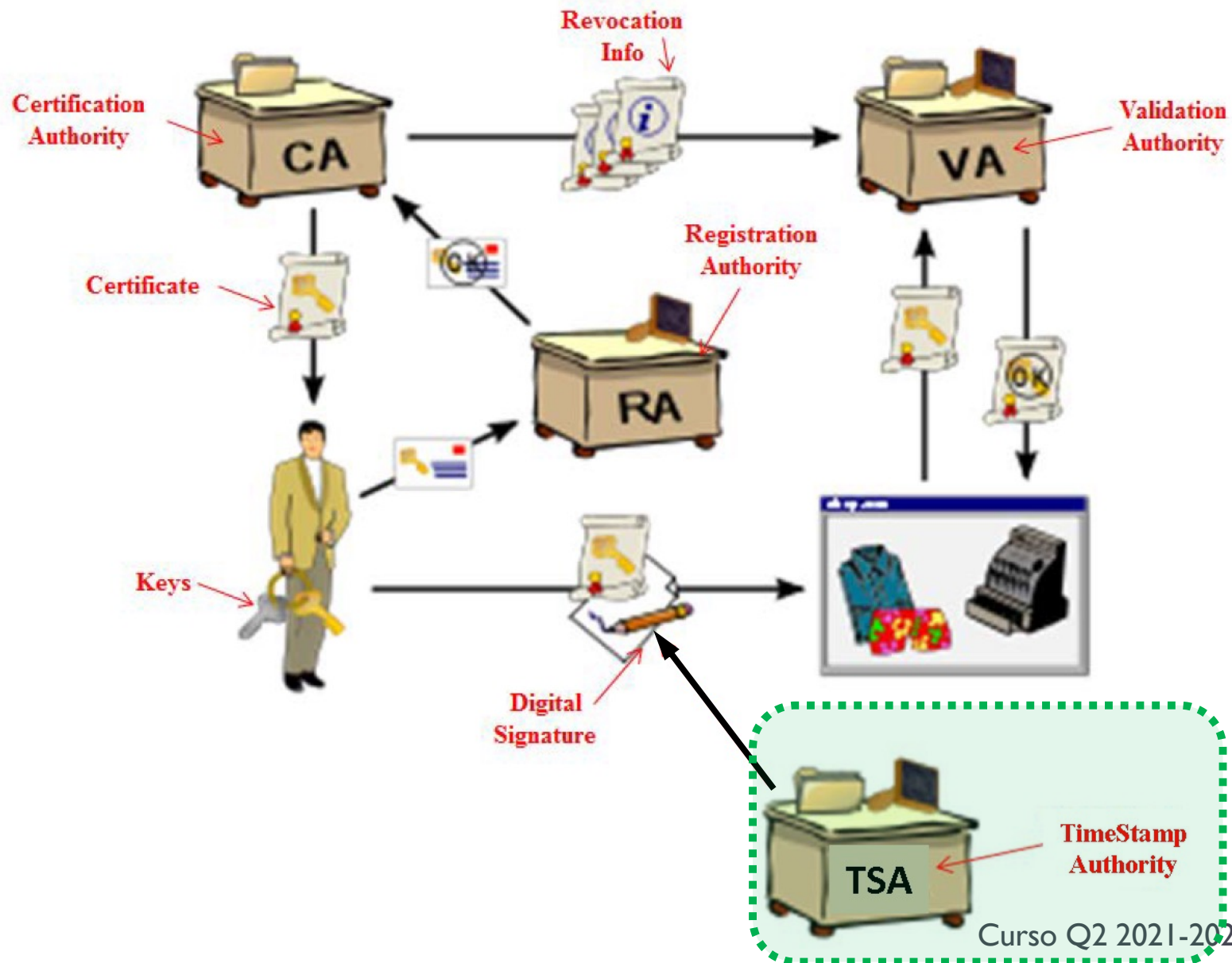
3.2 – Componentes de una PKI

Validation authority (VA)

- ▶ Una VA se encarga de verificar la validez de los certificados digitales
 - ▶ Puede ser directamente la CA o
 - ▶ Una entidad externa
- ▶ Proporciona información en tiempo real sobre el estado de un certificado (valid, suspend, revoked, unknown)
- ▶ Se usan protocolos de validación para conocer este estado actual
 - ▶ Online certificate status protocol (OCSP)
 - ▶ Simple certificate validation protocol (SCVP)

3.2 – Componentes de una PKI

Estructura



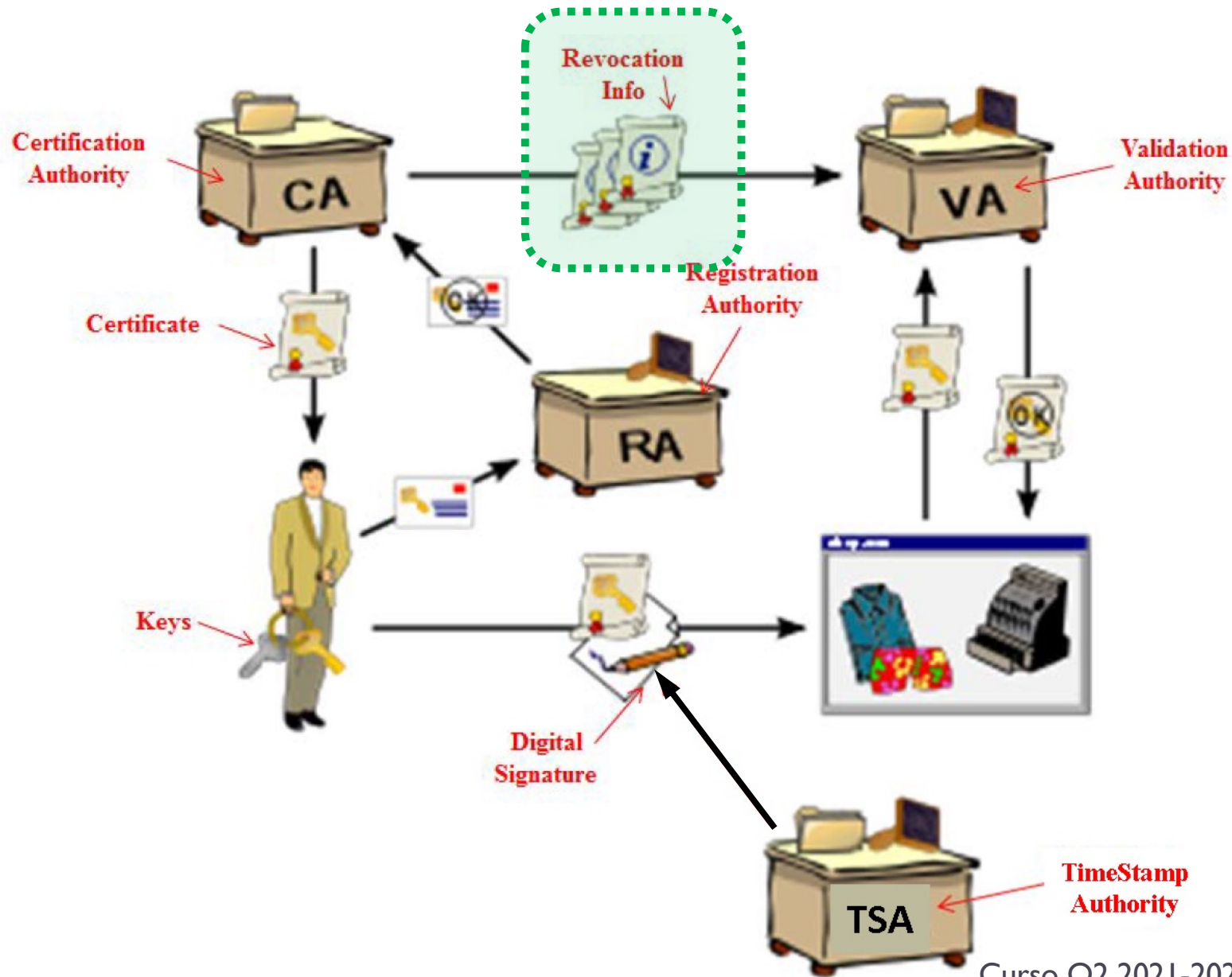
3.2 – Componentes de una PKI

Time stamping authority (TSA)

- ▶ Una TSA se encarga de firmar un mensaje para marcar en que momento se generó
- ▶ Los escenarios en los que las TSA son muy importantes:
 - ▶ Verificación de un documento firmado digitalmente. Si el certificado correspondiente ha sido revocado, la marca de tiempo permite decidir si el documento fue firmado antes de la revocación
 - ▶ Fecha límite de entrega de documentos. La marca de tiempo permite verificar si el documento se entregó a tiempo
 - ▶ Auditorias. Las marcas de tiempo permiten fechar todas las entradas pasadas (p.e., facturas, correos, documentos)

3.2 – Componentes de una PKI

Estructura



3.2 – Componentes de una PKI

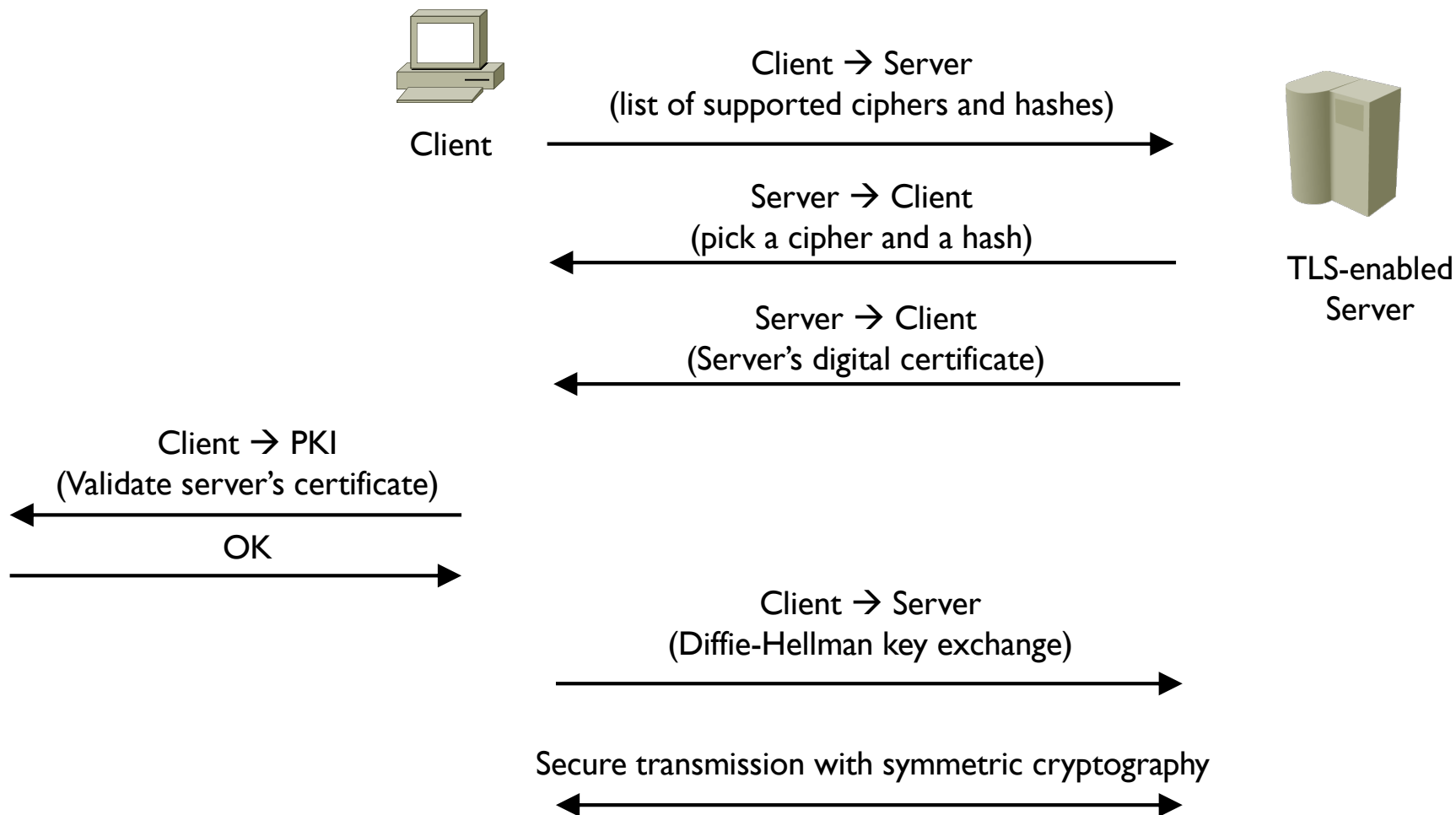
Repositorios

- ▶ Los **repositorios** son estructuras de datos para almacenar toda la información sobre una PKI
- ▶ Los dos repositorios más importantes son
 - ▶ El repositorio de certificados
 - ▶ El repositorio de la lista de certificaciones en revocación (CRL). Una CRL es una lista de certificados que se han revocado y, por lo tanto, no se debe confiar en ella
- ▶ Las **directories** son las estructuras de repositorio más comunes utilizadas en una PKI
 - ▶ Son bases de datos diseñadas para almacenar una gran cantidad de objetos tipificados (como certificados)
 - ▶ Están optimizados para leer, navegar y buscar

3.2 – Componentes de una PKI

Ejemplo: autenticación del servidor

- ▶ Para conectarse a un servidor web (HTTPS,TLS)



Tema 3. Índice

- ▶ 3.1 - Conceptos básicos
- ▶ 3.2 - Componentes
- ▶ 3.3 - Modelos de confianza (trust models)
- ▶ 3.4 - Certificados X.509
- ▶ 3.5 - Public-Key Cryptography Standards (PKCS)

3.3 - Trust models

- ▶ Modelo distribuido
- ▶ Modelo plano
- ▶ Modelo jerárquico
- ▶ Modelos híbridos
 - ▶ Modelo de lista de confianza jerárquica
 - ▶ Modelo de certificación cruzada jerárquica
 - ▶ Modelo de certificación de puente

3.3 - Trust models

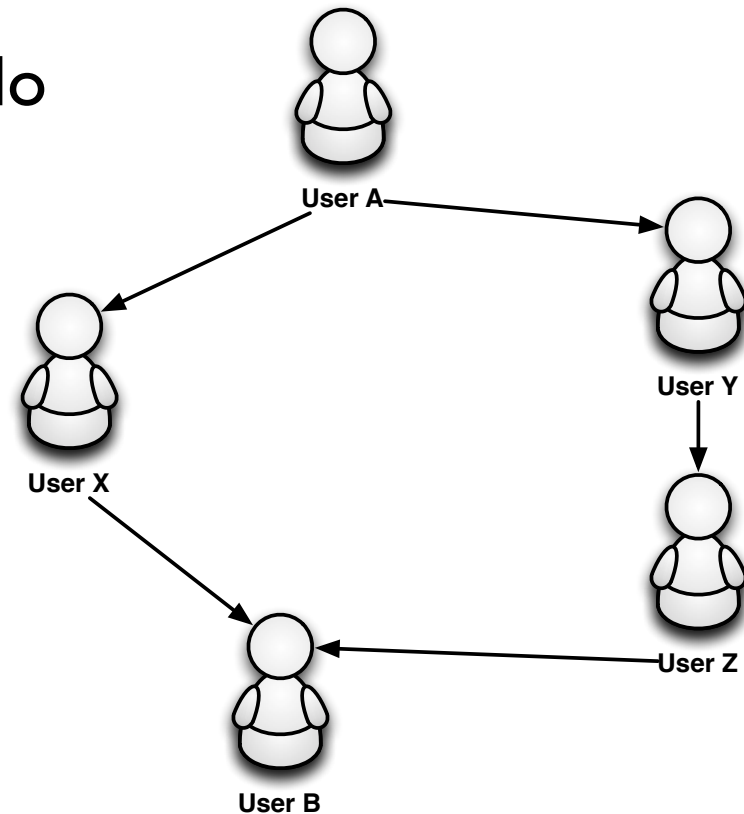
Modelo distribuido

- ▶ Modelo más simple
 - ▶ Funciona en una comunidad de pocos usuarios
 - ▶ Un usuario crea y firma certificados para otros usuarios
 - ▶ Los usuarios tienen confianza entre ellos
 - ▶ No necesitan TTP
-
- ▶ Pretty Good Privacy (PGP) usa este modelo
 - ▶ Asume que los usuarios son competentes para decidir si confiar en otro o no

3.3 - Trust models

Modelo distribuido - ejemplo

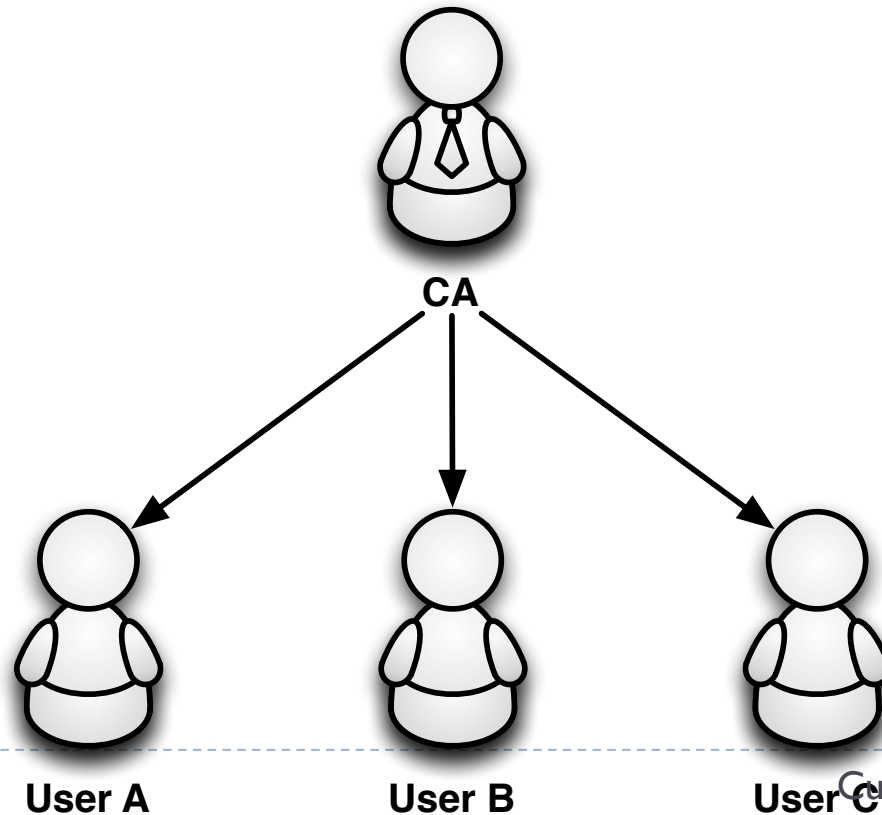
- ▶ X afirma (con una firma) que la clave de B es correcta
- ▶ Como A (el verificador) ha firmado el certificado de X, A está seguro de que el certificado de B es correcto.
- ▶ Además, encontró otra ruta de certificación que atraviesa a los usuarios Y y Z



3.3 - Trust models

Modelo plano

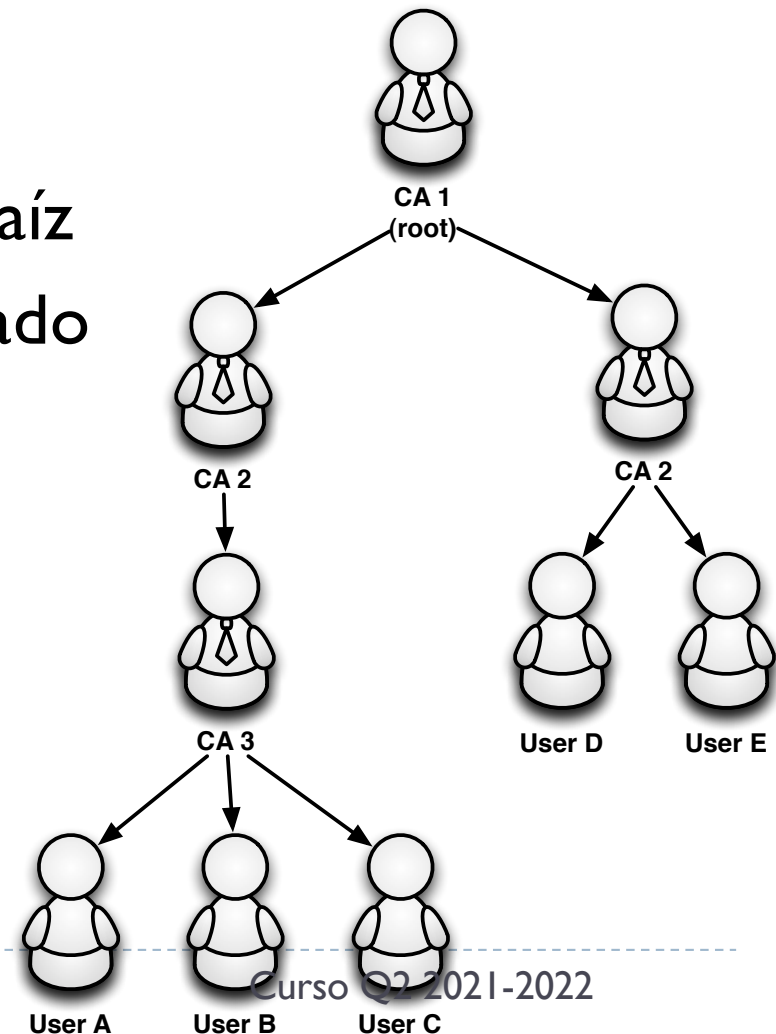
- ▶ El modelo simple solo tiene una CA que actúa como TTP
- ▶ Los usuarios validan la identidad de los suscriptores utilizando el certificado de CA
- ▶ El certificado de CA está autofirmado



3.3 - Trust models

Modelo jerárquico

- ▶ Los certificados de los usuarios se firman mediante un TTP
- ▶ Este TTP está identificado por medio de otro certificado emitido por otra CA con un nivel jerárquico superior
- ▶ Y así sucesivamente hasta el CA raíz
- ▶ El CA raíz (root) tiene un certificado autofirmado
- ▶ Para validar un certificado, se sigue el árbol desde abajo hasta el CA raíz



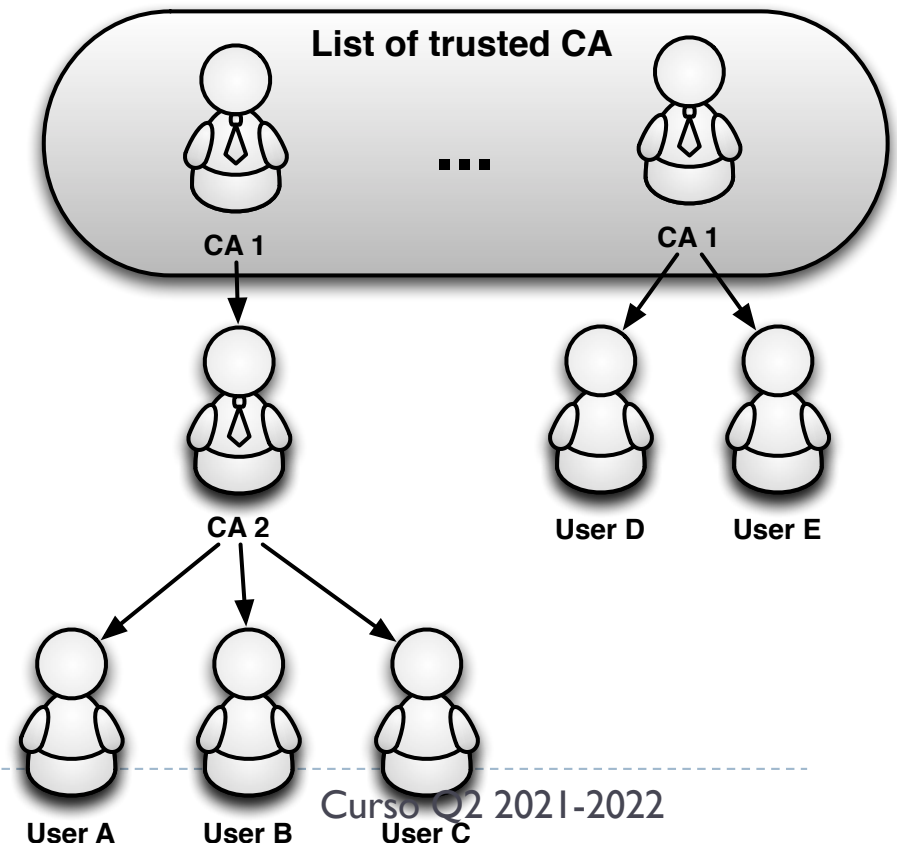
3.3 - Trust models

- ▶ Modelo distribuido
- ▶ Modelo plano
- ▶ Modelo jerárquico
- ▶ **Modelos híbridos**
 - ▶ Modelo de lista de confianza jerárquica
 - ▶ Modelo de certificación cruzada jerárquica
 - ▶ Modelo de certificación de puente

3.3 - Trust models

Modelo de lista de confianza jerárquica

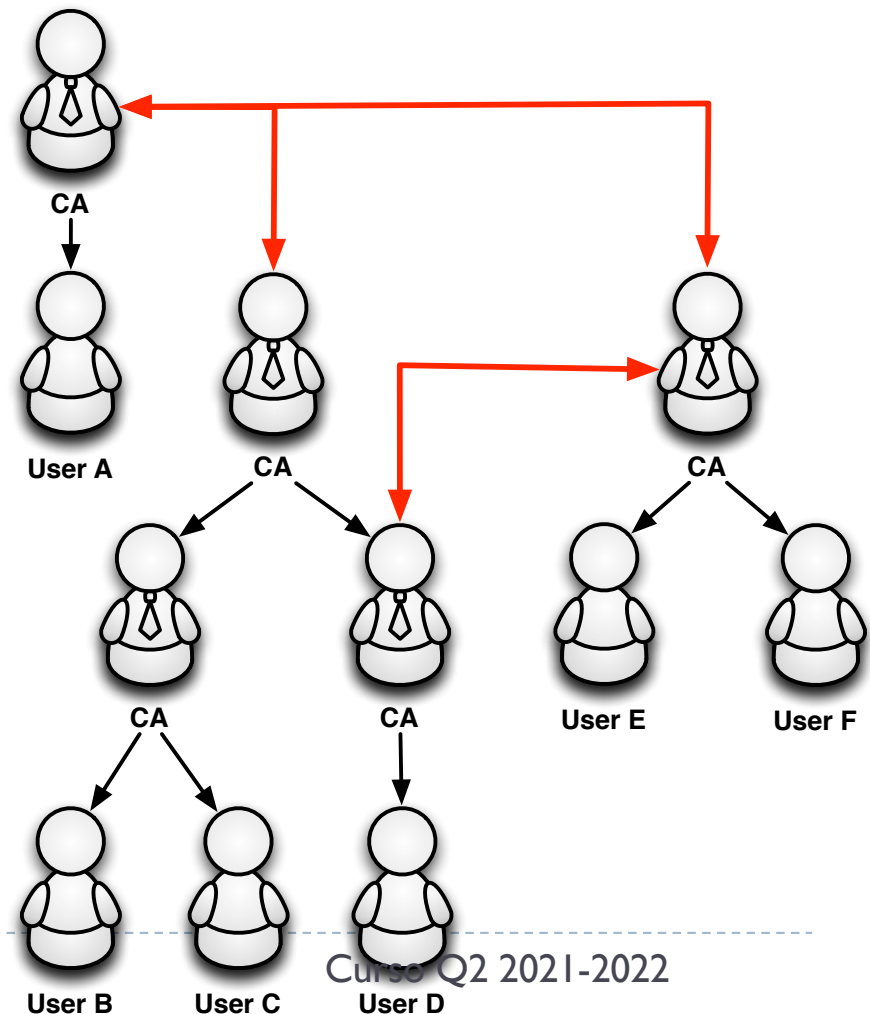
- ▶ El modelo más común es el de lista de confianza jerárquica
- ▶ También llamado modelo de centro de usuario
- ▶ En este caso, cada aplicación tiene una lista de CA de confianza (con sus correspondientes claves publicas)
- ▶ Se implementa en la mayoría de los navegadores web.
- ▶ Es muy flexible, el usuario puede agregar/eliminar CA de confianza
- ▶ Principal inconveniente no hay diferencia entre PKI buenas y no tan buenas



3.3 - Trust models

Modelo de certificación cruzada jerárquica

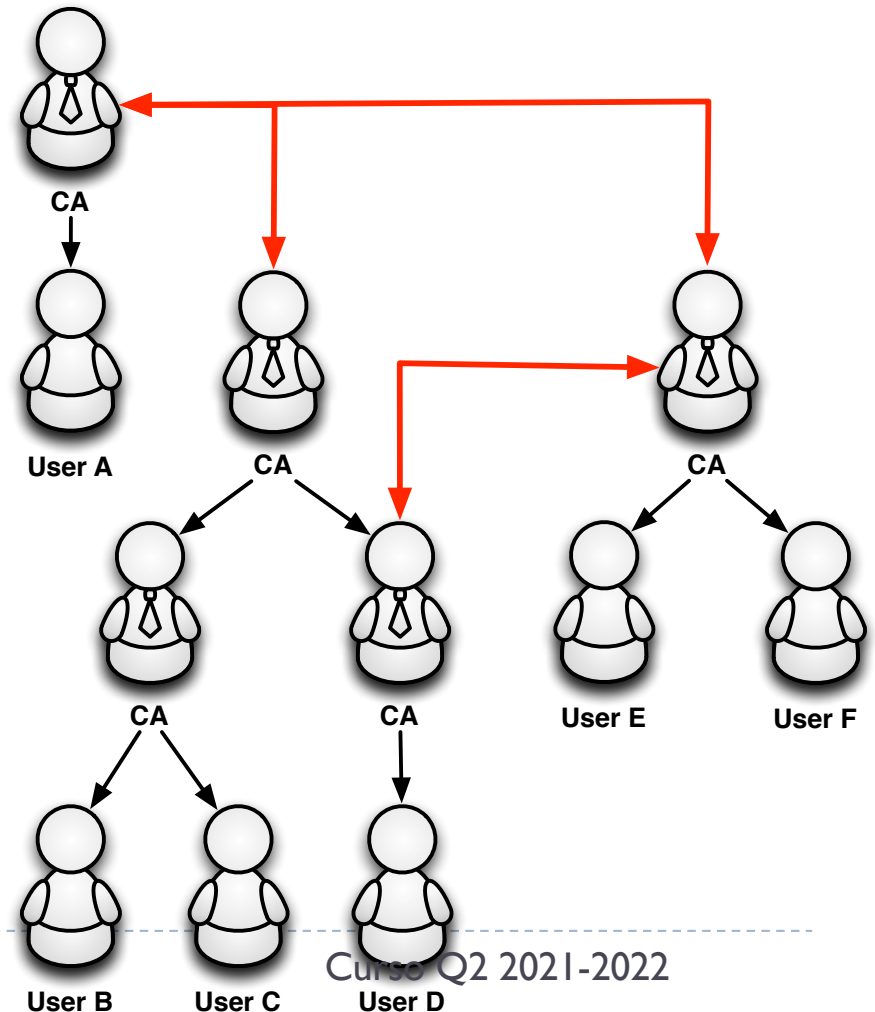
- ▶ Las CA raíz emiten certificados entre ellos
- ▶ Estos certificados certifican una CA utilizando la firma de otra CA (certificados cruzados)
- ▶ Las CA raíz son locales, no globales
- ▶ La verificación del certificado puede requerir una búsqueda costosa
- ▶ Como en el modelo jerárquico, un usuario confía en una sola CA



3.3 - Trust models

Modelo de certificación cruzada jerárquica

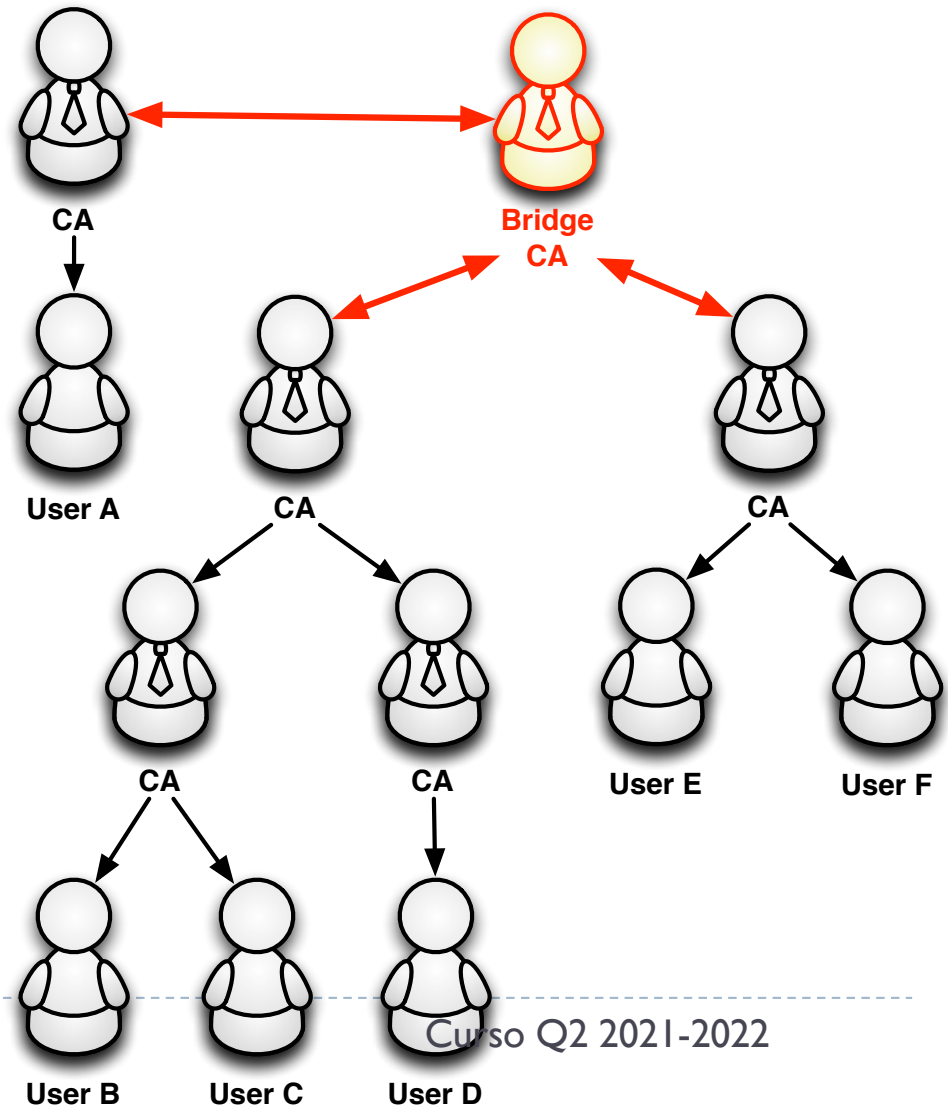
- ▶ Problema: el número de certificados cruzados crece enormemente con el número de CA
- ▶ Con 3 CA, se necesitan 6 certificados cruzados
- ▶ Con 100 CA, se necesitan 9,900 certificados cruzados



3.3 - Trust models

Modelo de certificación de puente

- ▶ El modelo CA puente (bridge) reduce la cantidad de certificados necesarios y aumenta la escalabilidad
- ▶ Un CA puente es una entidad externa que actúa de puente entre dos PKI
- ▶ Todos los CA raíz emiten un certificado cruzado con el CA puente
- ▶ Para 3 CA, se necesitan 6 certificados cruzados
- ▶ Para 100 CA, se necesitan 200 certificados cruzados



Tema 3. Índice

- ▶ 3.1 - Conceptos básicos
- ▶ 3.2 - Componentes
- ▶ 3.3 - Modelos de confianza (trust models)
- ▶ 3.4 - Certificados X.509
- ▶ 3.5 - Public-Key Cryptography Standards (PKCS)

3.4 - Certificados X.509

- ▶ Define un marco de autenticación
- ▶ También se le conoce como PKIX
- ▶ Proporciona un estándar para la certificación de clave pública
- ▶ Aspectos principales del formato X.509
 - ▶ Publicado oficialmente en 1998 a partir de la norma X.500
 - ▶ Modelo jerárquico (en la v3 se incluye cruzado y puente)
 - ▶ Mientras X.500 de ITU-T se usa poco (intercambio de información entre naciones), X.509 se ha convertido en el estándar en Internet y web
 - ▶ Versión 3 actual IETF RFC 5280 (mayo 2008)
 - ▶ Actualizada en 6818 (enero 2013), 8398 (mayo 2018), 8399 (mayo 2018)
 - ▶ Incluye aspectos de la PKI, formato de los certificados y de los CRLs

3.4 - Certificados X.509

- ▶ Una CA emite un certificado asociando una clave pública a un **Nombre Distinguido** (Distinguished Name, DN)

- ▶ Un DN es un conjunto de atributos con un cierto valor

- ▶ Permite a las organizaciones decidir cuáles son los atributos más adecuados para identificar una entidad en una PKI

- ▶ Si estos atributos no son suficientes se puede elegir un Nombre

Alternativo tal como una dirección

de correo electrónico, una entrada de DNS, una @IP

usual attributes	
CN	common name
OU	organization unit
O	organization
C	country
L	location
S	state
STREET	address
T	title

3.4 - Certificados X.509

Atributos básicos

Attribute	Description
version	Identifies the certificate version: v1, v2, v3
serialNumber	Certificate id assigned by the CA
signature	signature algorithm description
issuer	DN of the CA
validity	Two fields defining the time period validity (not before, not after)
subject	DN of the certificate owner
subjectPublicKeyInfo	Information about certificate public key
issuerUniqueId	Allows for CA names re-use (optional)
subjectUniqueId	Allows for owner names re-use (optional)
extensions	Contains all extensions (optional)

3.4 - Certificados X.509

Atributos de la firma

Attribute	Description
signatureAlgorithm	Contains the hash algorithm: MD2, MD5, SHA-1 and public key algorithm: RSA, DSA used to sign the certificate
signature	Contains the signature value of the basic attributes (including the extensions)

3.4 - Certificados X.509

Extensión de atributos

Attribute	Description
subjectAltName	Owner alternative name (email, ...)
issuerAltName	CA alternative name
keyUsage	Defines and limits certificate usage
basicConstraints	<i>Final entity</i> = 0 or <i>CA</i> = 1, it limits the certification chain length
extKeyUsageSyntax	Complements <i>basicConstraints</i> and <i>keyUsage</i> attributes to set up the usages inside the PKI
nameConstraints	Limit the name space of the next certificates of the chain
issuerAltName	CA alternative name
certificatePolicies	Information about the CPS of the CA
policyMappings	Maps different CA policies
...	...

3.4 - Certificados X.509

Ejemplo

DATA:

Version: 3 (0x2)

Serial Number: 7829 (0x1e95)

Signature Algorithm: md5WithRSAEncryption

La función de Hash
usada para firmar

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

La CA que ha
emitido y firmado
el certificado

Validity:

Not Before: Jul 9 16:04:02 2011 GMT

Not After: Jul 9 16:04:02 2012 GMT

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala, OU=FreeSoft,
CN=www.freesoft.org
emailAddress=baccala@freesoft.org

El propietario del
certificado

Subject Public Key Info:

Public Key Algorithm:

RSA Public Key:

rsaEncryption

(1024 bit)

Modulus (1024 bit): ...

Exponent: 65537 (0x10001)

Algoritmo, modulo
y exponente de la
clave pública del
propietario del
certificado

SIGNATURE:

Certificate Signature Algorithm: md5WithRSAEncryption

Certificate Signature: ... → Firma digital de la CA

3.4 - Certificados X.509

Ejemplo

Certificate: Data: Version: 3 (0x2) 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)

Modulo y exponente de la clave pública del propietario del certificado

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)

Certificate Signature Algorithm: md5WithRSAEncryption Certificate

Certificate Signature:

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

Firma digital de la CA

3.4 - Certificados X.509

Ejemplos



Safari is using an encrypted connection to discussions.apple.com.

Encryption with a digital certificate keeps information private as it's sent to or from the https website discussions.apple.com.

Apple Inc. has identified discussions.apple.com as being owned by Apple Inc. in Cupertino, California, US.

DigiCert High Assurance EV Root CA
Apple Public EV Server RSA CA 2 - G1
discussions.apple.com

discussions.apple.com
Issued by: Apple Public EV Server RSA CA 2 - G1
Expires: Saturday, 12 February 2022 at 22:23:43 Central European Standard Time
This certificate is valid

> Trust
v Details

Subject Name

Business Category Private Organization
Inc. Country/Region US
Inc. County California
Serial Number C0806592
Country or Region US
County California
Locality Cupertino
Organisation Apple Inc.
Organisational Unit management:idms.group.1208920
Common Name discussions.apple.com

Issuer Name

Country or Region US
Organisation Apple Inc.
Common Name Apple Public EV Server RSA CA 2 - G1

Serial Number 3A AA 0C A4 91 B0 08 AF 73 74 85 75 3C 18 1B 61
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None

Not Valid Before Wednesday, 13 January 2021 at 22:23:43 Central European Standard Time
Not Valid After Saturday, 12 February 2022 at 22:23:43 Central European Standard Time

Public Key Info

Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters None
Public Key 256 bytes: EB 2A 53 7D 0C 83 C6 83 ...
Exponent 65537
Key Size 2048 bits
Key Usage Encrypt, Verify, Wrap, Derive
Signature 256 bytes: 7C 62 F2 CE 7D 27 58 9F ...

El propietario del certificado

La CA que ha emitido y firmado el certificado

Algoritmo, modulo y exponente de la clave pública del propietario del certificado

DigiCert High Assurance EV Root CA
DigiCert SHA2 High Assurance Server CA
*.wikipedia.org

DigiCert SHA2 High Assurance Server CA
Intermediate certificate authority
Expires: Sunday, 22 October 2028 at 14:00:00 Central European Summer Time
This certificate is valid

> Trust
v Details

Subject Name

Country or Region US
Organisation DigiCert Inc
Organisational Unit www.digicert.com
Common Name DigiCert SHA2 High Assurance Server CA

Issuer Name

Country or Region US
Organisation DigiCert Inc
Organisational Unit www.digicert.com
Common Name DigiCert High Assurance EV Root CA

Serial Number 04 E1 E7 A4 DC 5C F2 F3 6D C0 2B 42 B8 5D 15 9F
Version 3
Signature Algorithm SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters None

Not Valid Before Tuesday, 22 October 2013 at 14:00:00 Central European Summer Time
Not Valid After Sunday, 22 October 2028 at 14:00:00 Central European Summer Time

Public Key Info

Algorithm RSA Encryption (1.2.840.113549.1.1.1)
Parameters None
Public Key 256 bytes: B6 E0 2F C2 24 06 C8 6D ...
Exponent 65537
Key Size 2048 bits
Key Usage Encrypt, Verify, Derive
Signature 256 bytes: 18 8A 95 89 03 E6 6D DF ...

Firma digital de la CA

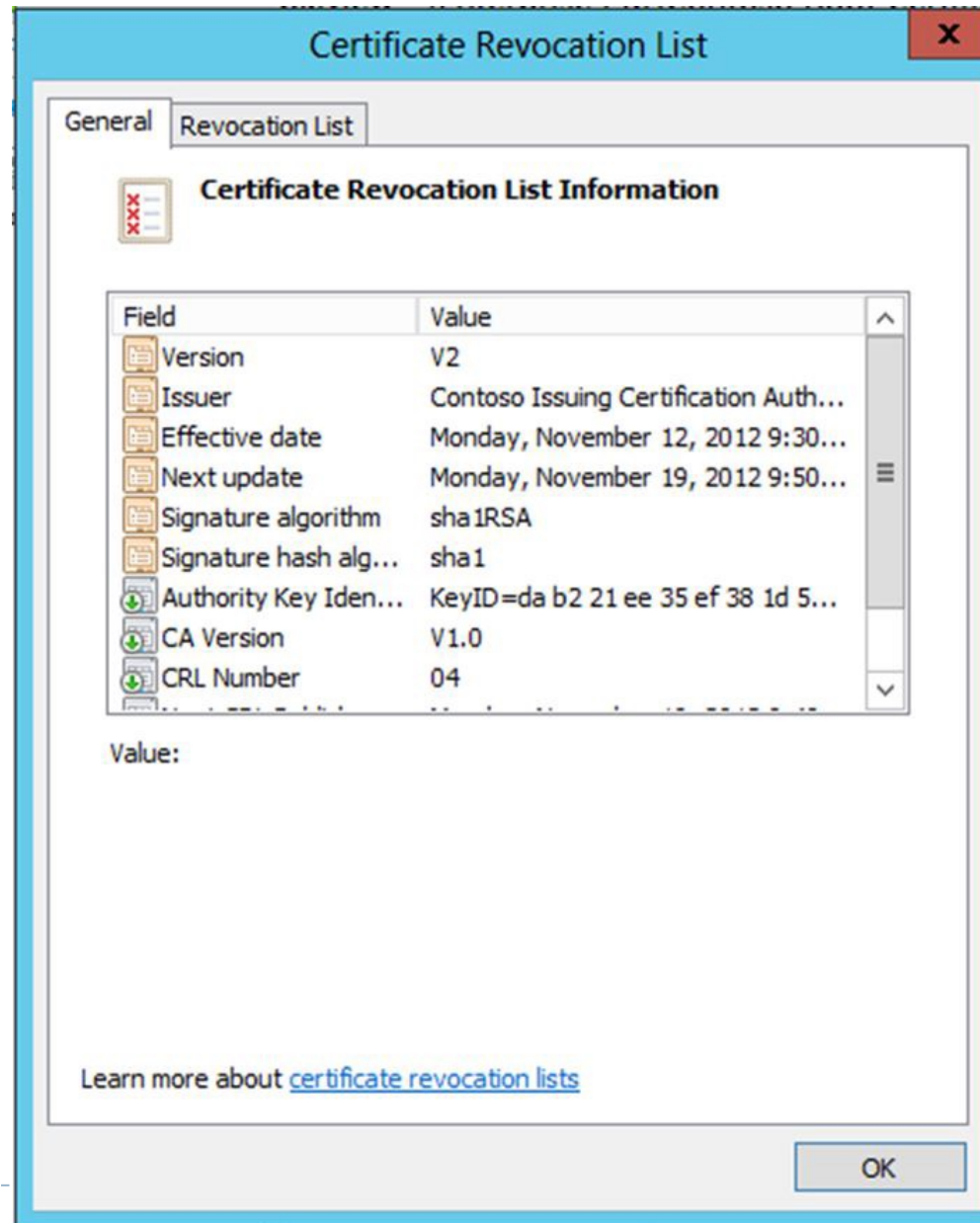
3.4 - Certificados X.509

- ▶ Una CRL es una estructura de datos firmada por una CA
- ▶ Contiene los datos esenciales para determinar el estado del certificado
- ▶ Atributos básicos de los CRLs

Attribute	Description
version	Identifies the CRL version: v1 or v2
signature	Signature algorithm description
issuer	DN of the signing CA (usually the same who issues the list)
thisUpdate	Issue date
nextUpdate	Next issue date
revokedCertificates	List of revoked certificates
crlExtensions	Contains extensions related to the CRL (optional)

3.4 - Certificados X.509

Ejemplo de Certificate Revocation List



Tema 3. Índice

- ▶ 3.1 - Conceptos básicos
- ▶ 3.2 - Componentes
- ▶ 3.3 - Modelos de confianza (trust models)
- ▶ 3.4 - Certificados X.509
- ▶ 3.5 - Public-Key Cryptography Standards (PKCS)

3.5 - Public-Key Cryptography Standards

- ▶ Grupos de estándares promovidos por RSA Security LLC desde 1991
 - ▶ Empresa que promueve el uso de las técnicas de criptografía
 - ▶ Tiene las patentes del algoritmo RSA, del algoritmo de firma Schnorr y de muchos otros
- ▶ No son estándares (la empresa retiene el control sobre ellos)
- ▶ Pero se han convertido en estándares de facto
 - ▶ Empresas como Apple, Microsoft, DEC, Sun, etc. lo han adoptado
 - ▶ Grupos de trabajo en IETF y PKIX

3.5 - Public-Key Cryptography Standards

- ▶ Estos estándares están numerados del 1 al 15
 - ▶ PKCS#1, PKCS#2, ..., PKCS#15
- ▶ De estos, actualmente solo se han completado y se usan 10
 - ▶ PKCS#1; #3; #5; #7; #8; #9; #10; #11; #12; #13; #14; #15
 - ▶ PKCS#13 (elliptic curves) y #14 (pseudorandom numbers generation) parece que se han abandonado
 - ▶ PKCS#2 (cifrado de resúmenes de mensajes) y #4 (sintaxis de la clave) se han integrado en PKCS#1
 - ▶ PKCS#6 (extensiones para X.509v1) se ha abandonado a favor del certificado X.509v3
 - ▶ Updated list <https://en.wikipedia.org/wiki/PKCS>

3.5 - Public-Key Cryptography Standards

PKCS	Description	Standard IETF
1	Defines encryption and signature protocols for RSA. It includes a syntax (equal to X.509) for private and public keys	RFC 8017
3	Defines a Diffie-Hellman key agreement	
5	Defines a protocol to encrypt a text m with a private key obtained from the hash of a pass phrase p . $E_{H(p)}(m) = c$. H is MD2 or MD5	RFC 8018
7	Defines the syntax of an encrypted and(or) signed message	RFC 2315 y RFC 5652
8	Defines the information format of a private key	RFC 5958
9	Defines attribute types for PKCS standards	RFC 2985
10	Defines the format of a certification request	RFC 2986
11	Defines the Cryptoki interface, an independent programming language for smart cards	

3.5 - Public-Key Cryptography Standards

PKCS	Description	Standard IETF
12	Defines a portable format to store private keys, certificates, ...	RFC 7292
13	Define methods to encrypt and sign messages with elliptic curves cryptography	Abandonado
14	Devoted to pseudorandom numbers generation	Abandonado
15	Complements <i>PKCS#11</i> defining the format of the cryptographical credentials stored into cryptographical devices	ISO/IEC 7816-15

3.5 - Public-Key Cryptography Standards

PKCS#7

- ▶ Cryptographic Message Syntax Standard
- ▶ Versión 1.5
- ▶ IETF RFC 5652 (Sep. 2009) y RFC 8933 (Dic. 2020)
- ▶ Define la sintaxis de los mensajes cifrados y/o firmados
 - ▶ Permite que un usuario firme un mensaje ya firmado (encriptado)
- ▶ Hoy en día se utiliza para
 - ▶ firmar y/o cifrar mensajes bajo una PKI
 - ▶ la difusión de certificados
 - ▶ el inicio de sesión única (single sign-on)

3.5 - Public-Key Cryptography Standards

PKCS#7 elementos

- ▶ **Data**

- ▶ Información sin firmar

- ▶ **SignedData**

- ▶ Información firmada digitalmente

- ▶ **EnvelopedData**

- ▶ Información para un grupo de destinos
- ▶ Se crea un sobre digital para cada destino
- ▶ Se genera (aleatoria) una clave privada y se cifra la información (cifrado simétrico)
- ▶ Se cifra la clave privada con la clave pública de cada destino (cifrado asimétrico)
- ▶ Cada destino podrá sacar la clave privada del cifrado simétrico usando su clave privada (cifrado asimétrico)
- ▶ Con la clave privada del cifrado simétrico podrá descifrar la información

3.5 - Public-Key Cryptography Standards

PKCS#7 elementos

▶ SignedAndEnvelopedData

- ▶ Información doblemente cifrada
- ▶ Se firma la información con la clave privada (cifrado asimétrico)
- ▶ Se genera (aleatoria) una clave privada y se cifra la información (cifrado simétrico)
- ▶ Se cifra la clave privada con la clave publica de cada destino (cifrado asimétrico)
- ▶ Cada destino podrá sacar la clave privada del cifrado simétrico usando su clave privada (cifrado asimétrico)
- ▶ El destino con la clave privada del cifrado simétrico podrá descifrar la información
- ▶ Y con la clave pública del origen verificar quien ha firmado

3.5 - Public-Key Cryptography Standards

PKCS#7 elementos

- ▶ **EncryptedData**

- ▶ Información cifrada para ningún destinatario concreto

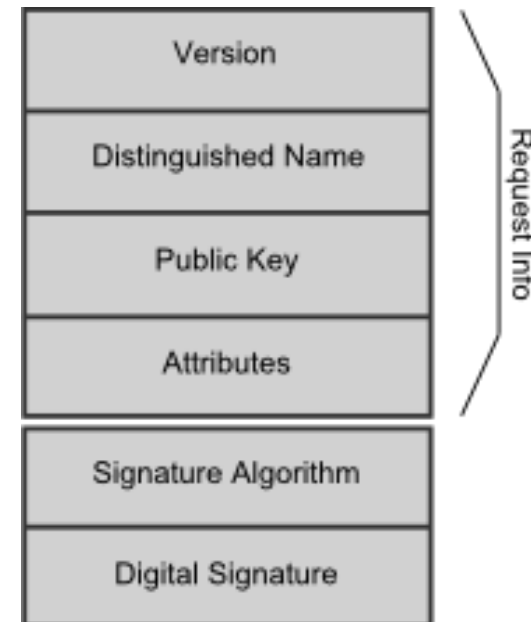
- ▶ **DigestedData**

- ▶ Se calcula un resumen de la información (por ejemplo función Hash)
- ▶ Se junta a la información y su resumen
- ▶ Esta unión se puede luego usar como input de otros tipos

3.5 - Public-Key Cryptography Standards

PKCS#10

- ▶ Certification Request Standard
- ▶ Define el formato de una solicitud de certificado
 - ▶ Una solicitud de certificado consta de un nombre, una clave pública y un conjunto opcional de atributos de usuario (por ejemplo, una clave de revocación)
- ▶ Versión: PKCS#10
- ▶ DN del usuario
- ▶ Información sobre la clave publica
- ▶ Información adicional sobre el usuario



3.5 - Public-Key Cryptography Standards

PKCS#12

- ▶ Personal Information Exchange Syntax Standard
- ▶ Versión 1.1
- ▶ IETF RFC 7292 (Julio 2014)
- ▶ Describe el formato para la transferencia de información personal, que puede incluir claves privadas, certificados, extensiones, etc.
 - ▶ En principio, las aplicaciones que admiten este formato permiten un intercambio seguro de datos.

3.5 - Public-Key Cryptography Standards

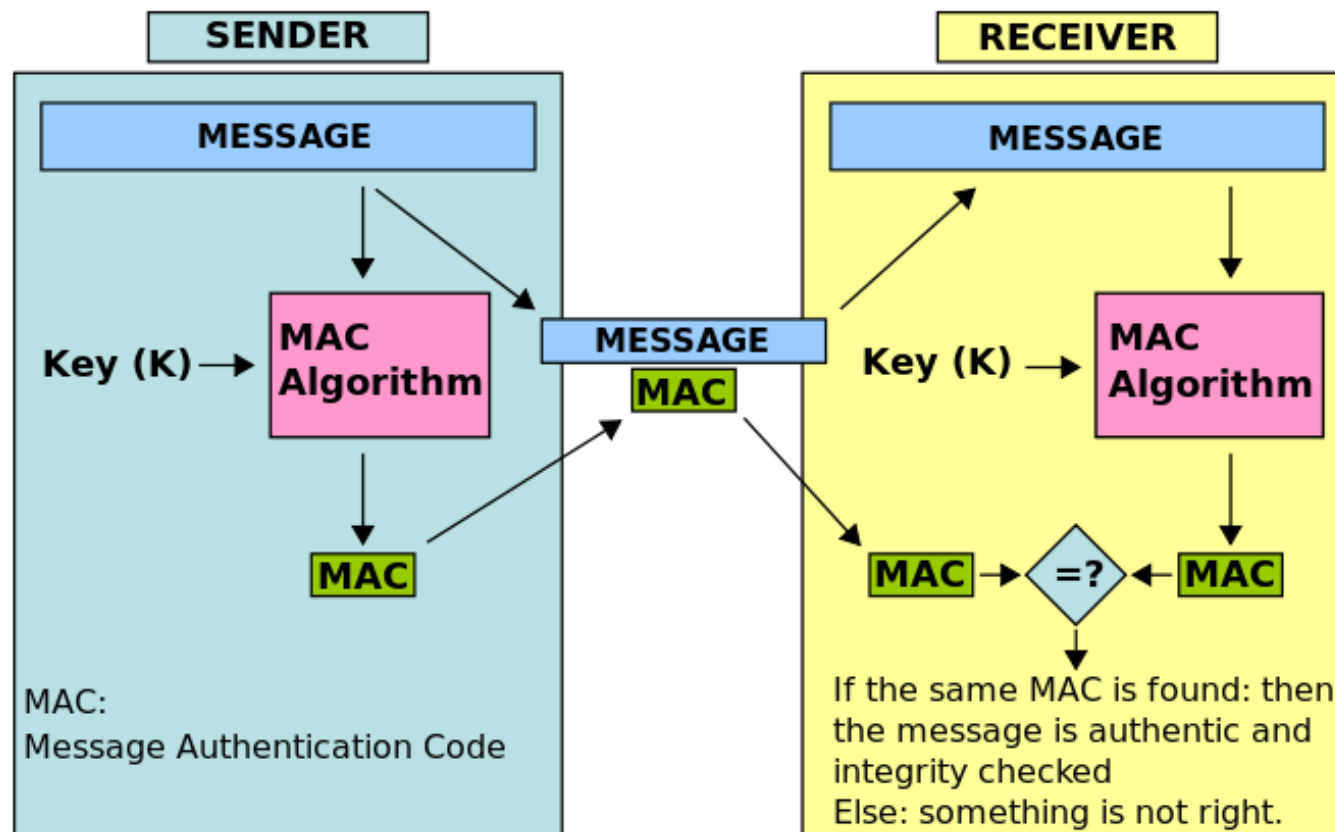
PKCS#12

- ▶ Tiene cuatro modos diferentes dependiendo de si el usuario quiere privacidad o integridad:
 - 1) **Public key privacy**: la información personal se empaqueta y se cifra en el repositorio de origen con la clave pública del repositorio de destino
 - 2) **Password privacy**: la información personal está empaquetada y encriptada con nombre de usuario y contraseña
 - 3) **Public key integrity**: la integridad se otorga mediante una firma con la clave privada del repositorio de origen. El destino verifica usando la clave publica del origen
 - 4) **Password integrity**: la integridad se garantiza utilizando un código de autenticación del mensaje (MAC) derivado de una contraseña de integridad del origen. Solo lo puede verificar quien tiene la información cifrada y la contraseña

3.5 - Public-Key Cryptography Standards

PKCS#12

- ▶ **MAC: Message Authentication Code**
 - ▶ Parecido a una función Hash
 - ▶ Usado para validar la integridad de cada mensaje



Fuente imagen: Wikipedia

3.5 - Public-Key Cryptography Standards

PKCS#12

- ▶ Se recomienda el uso de los modos con clave publica
 - ▶ Una contraseña puede no ser suficiente
 - ▶ No obstante los métodos con contraseña son los únicos posibles si no se tiene el certificado del destino
- ▶ Formato

PKCS#12	Description
version	PKCS#12 message version used
authSafe	PKCS#7 structure containing a <i>signedData</i> attribute when public key method is used or <i>data</i> otherwise
macData	It is only used when integrity is granted by means of a password
mac	PKCS#7 digestData
macSalt	random seed for the hash
iterations	Not in use, it is only for historical reasons

Seguretat Informàtica (SI)

Tema 3. Infraestructura PKI

Davide Careglio