Tecnologías GRID

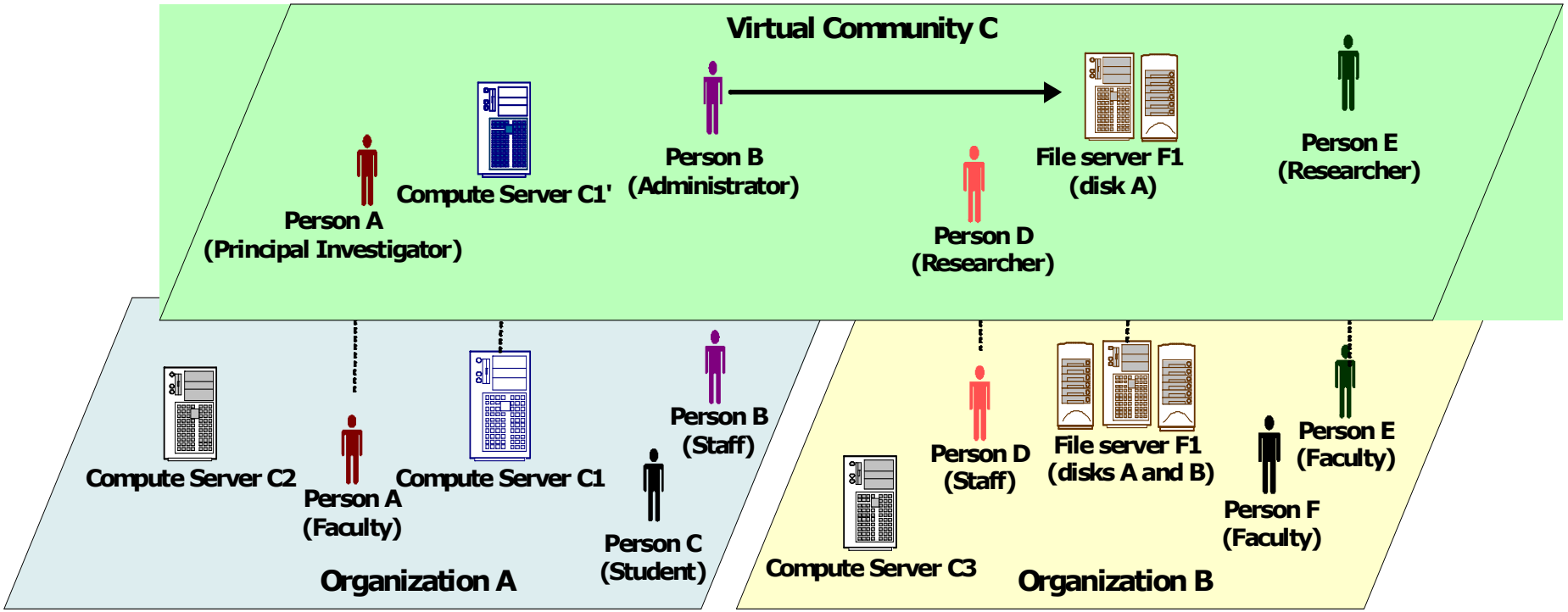
Estándares de Seguridad

Área de Arquitectura y Tecnología de Computadores
Departamento de Informática de la Universidad de Oviedo

Necesidades de seguridad en Grids

Objetivos del Grid: Coordinar el uso de recursos en organizaciones multi-institucionales

Organizaciones Virtuales (OVs)



Por tanto: Seguridad en un Grid ← → Seguridad en Organizaciones Virtuales

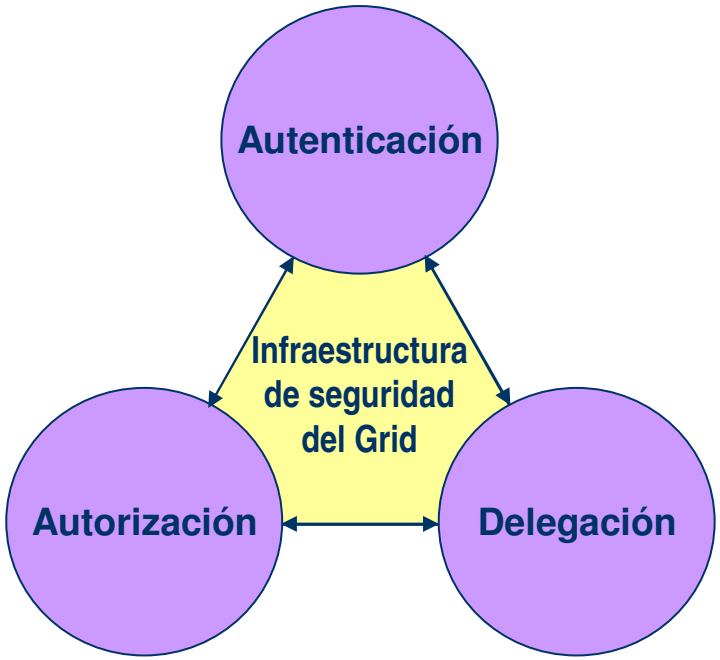
¿Cuáles son las necesidades de seguridad en OVs?

Áreas básicas de la seguridad en los Grids

1. Autenticación:
   Comunicación de la identidad

2. Autorización:
   Una vez conocida la identidad
   ¿Qué puede hacer un usuario?

3. Delegación:
   A permite a B actuar en nombre de A



Aspectos convenientes (aunque no estrictamente obligatorios):

- Confidencialidad de mensajes: Solo el emisor y el receptor pueden entender el mensaje
- No repudiación: sabido quien hizo que y cuando, no podrá negarlo
- Integridad de los mensajes: se podrá detectar las alteraciones en los mensajes

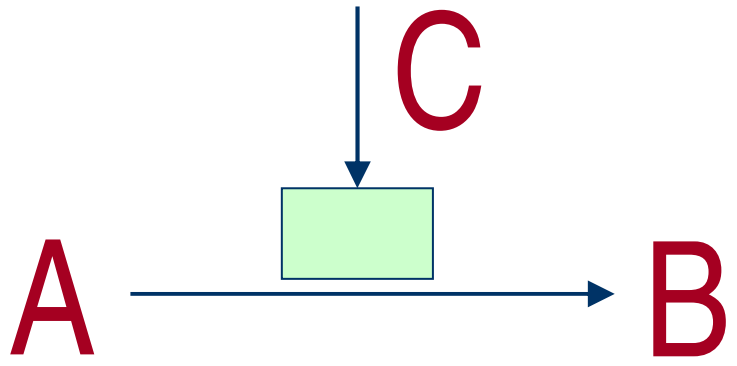
Fundamentos: Criptografía

Objetivos básicos de la criptografía:

USOS	SERVICIO	PROTEGE CONTRA
Mantener el secreto	Confidencialidad	Escucha a escondidas
Probar la identidad	Autenticación	Falsificación y Suplantación
Verificar la información	Integridad	Alteración

Ejemplo con tres personas: A, B y C

A envía una postal a B y C trata de actuar (sobre la postal)



Fundamentos: Criptografía

① Mantener el secreto

¿Cómo garantizar que solo B puede leer la postal?

A encripta (cifra) el texto y C no puede interpretarlo
B desencripta (descifra) el texto para leerlo

A y B comparten la técnica de Encriptación/Desencriptación

② Probar la identidad

¿Cómo sabe B que la postal recibida es de A?

A puede incluir una frase secreta en la postal que solo la conocen A y B
Cuando B lee la postal, si encuentra la frase, sabe que es de A

③ Prevenir alteraciones

¿Cómo sabe B que la postal no ha sido alterada?

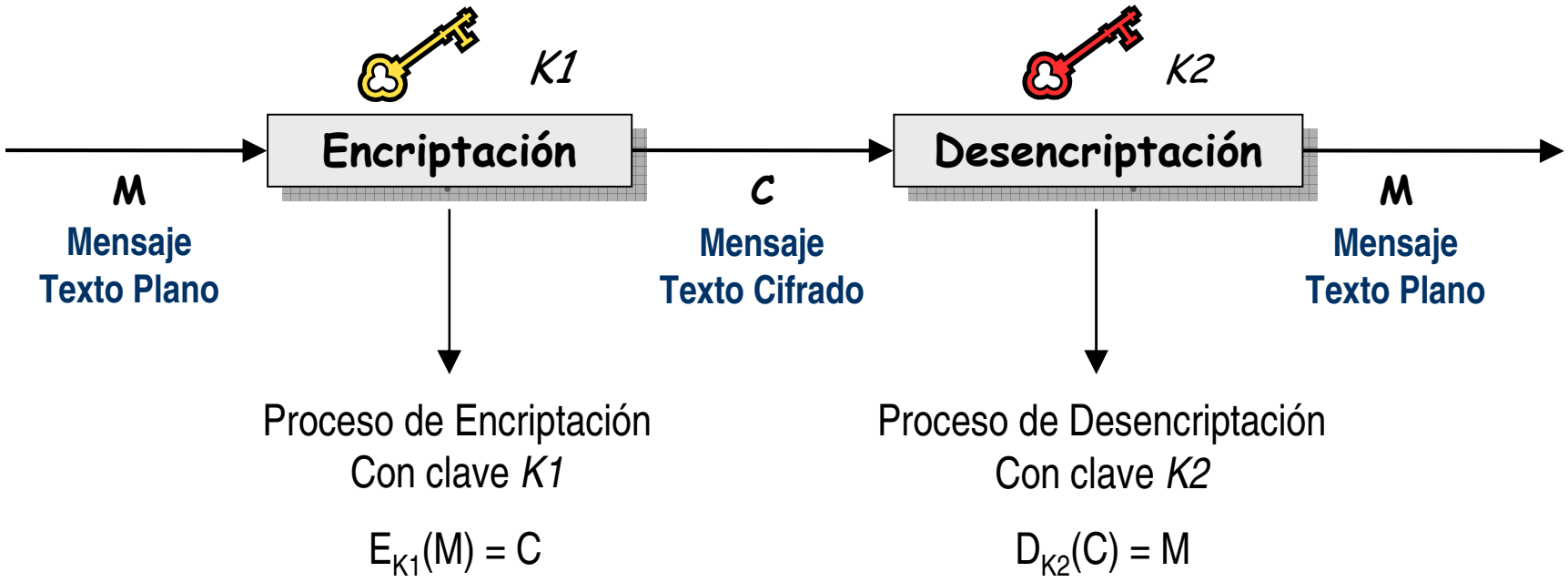
A calcula un valor a partir de la información que contiene la tarjeta
Le suma un nuevo valor conocido solo por A y B y lo incluye en la tarjeta

B calcula el valor a partir de la información que contiene la tarjeta
El valor debe coincidir con el valor contenido en la propia tarjeta

Criptografía: El cifrado de la información

Base: Algoritmos de Encriptación y Desencriptación

Son los elementos básicos para implementar una infraestructura de seguridad



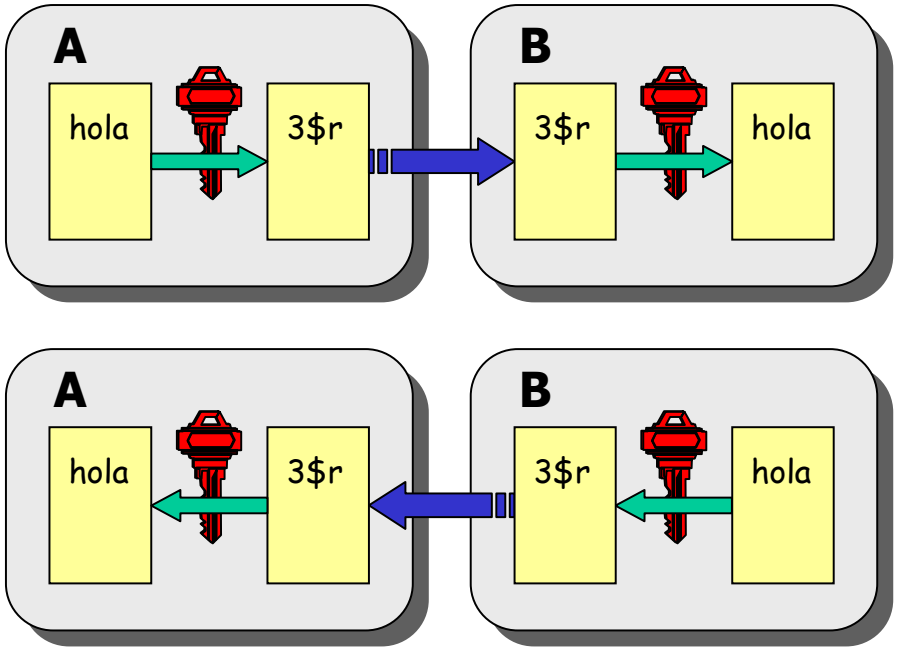
Tipos de Algoritmos → { Simétricos: $K1 = K2$; Asimétricos: $K1 \neq K2$

Criptografía: Algoritmos Simétricos

Ambas partes de una comunicación comparten la misma clave

La clave es **DESCONOCIDA** por otros → Criptografía de clave secreta (o privada)

La MISMA clave es usada para la encriptación y la desencriptación



Desventajas

A y B deben tener la misma clave
¿Cómo la intercambian?

El número de claves de $O(n^2)$
n = número de usuarios

Efectividad de la encriptación ← → Función de la longitud de la clave

Criptografía: Algoritmos Simétricos

Los cifradores pueden procesar los datos de 2 formas:

1. **Stream:** Byte a byte
2. **Block:** En bloques de tamaño fijo (típico: 8 bytes)

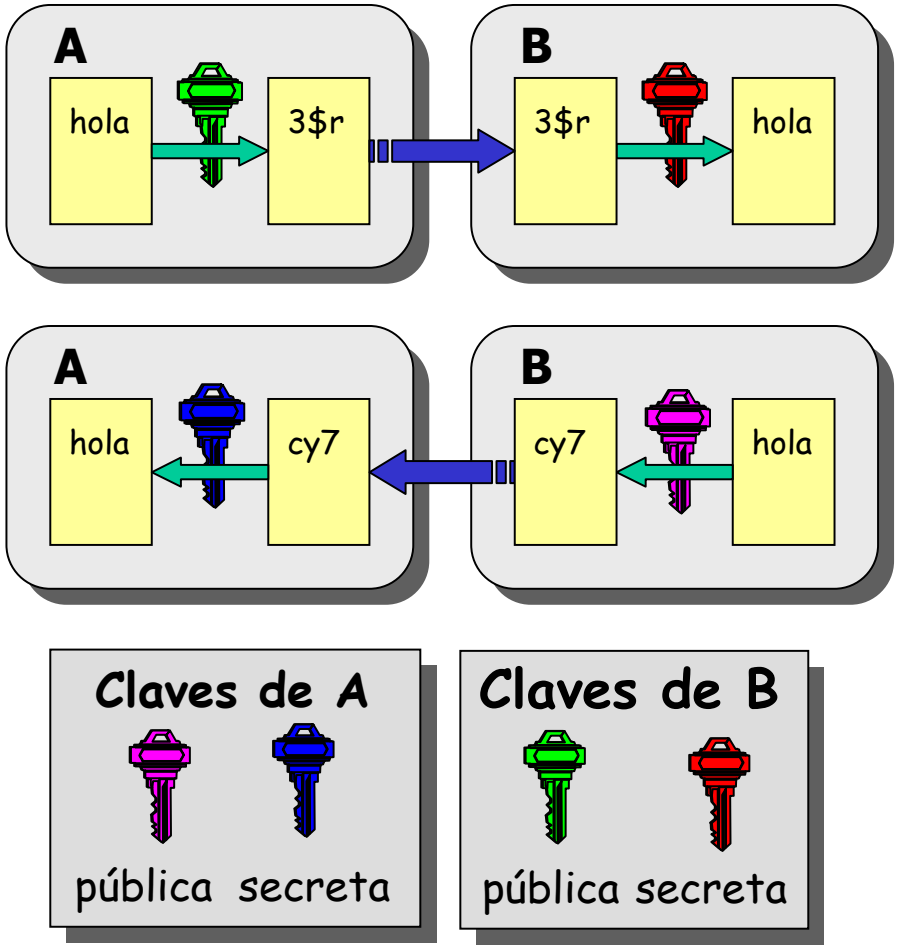
   Necesitan menos CPU y son algo más vulnerables a ataques
   Hay que completar los bloques con relleno hasta 8 bytes
   Requieren un vector de datos para su inicialización

Algoritmos de cifrado simétricos más comunes:

Acrónimo	Algoritmo	Tipo
DES	Data Encryption Standard	Block
3DES	Triple-Strength Data Encryption Standard	Block
RC2	Rivest Cipher 2	Block
RC4	Rivest Cipher 4	Stream

Criptografía: Algoritmos Asimétricos

Ambas partes de una comunicación usan DOS claves distintas { Una para encriptación / Otra para desencriptación



B desea que A le envíe un mensaje confidencial

1. B crea 2 claves { Pública / Secreta
2. B difunde su clave pública que lee A
3. A encripta el mensaje CON la clave pública de B
4. A envía el mensaje encriptado a B
5. B desencripta el mensaje CON la clave secreta de B

Solo B puede descifrar el mensaje
¡El emisor A NO PUEDE descifrarlo!

Criptografía: Combinación de Algoritmos

Encriptación con clave pública → Muy compleja → Coste computacional ↑↑

Solución: Combinar algoritmos asimétricos con simétricos

PASOS:

Variante: Pasos 1 a 3

Usar algoritmo de clave pública para intercambio de claves

El más usado: Diffie-Hellman

1. B crea 2 claves: Pública + Privada
   B difunde su clave pública
2. A recupera la clave pública de B
   A genera una lista de números aleatorios Na
   A encripta la lista de Na con la clave pública de B y los envía a B
3. B descifra los números aleatorios con su clave privada
   Solo puede hacerlo B pues solo B tiene la clave privada
4. A y B usan los números aleatorios como clave simétrica compartida
   Todos los intercambios de información los hacen con criptografía simétrica
   (a un coste computacional bajo)

Criptografía: Algoritmo RSA

El algoritmo de clave pública más utilizado: **RSA** Inventado por Rivest, Shamir y Adleman

Puede trabajar al revés (de cómo se ha explicado hasta ahora):

Información ENCRIPTADA con una clave privada	→	Puede ser DESENCRIPTADA con La clave pública correspondiente

Aplicación 1: B prueba su identidad a A

1. B encripta una información (conocida por A y B) CON la clave privada de B
2. B envía la información encriptada a A
3. A desencripta la información CON la clave pública de B
4. A compara las informaciones (descifrada y conocida) Si Coinciden A sabe que la ha enviado B

Criptografía: Algoritmo RSA

Aplicación 2: Firmas digitales

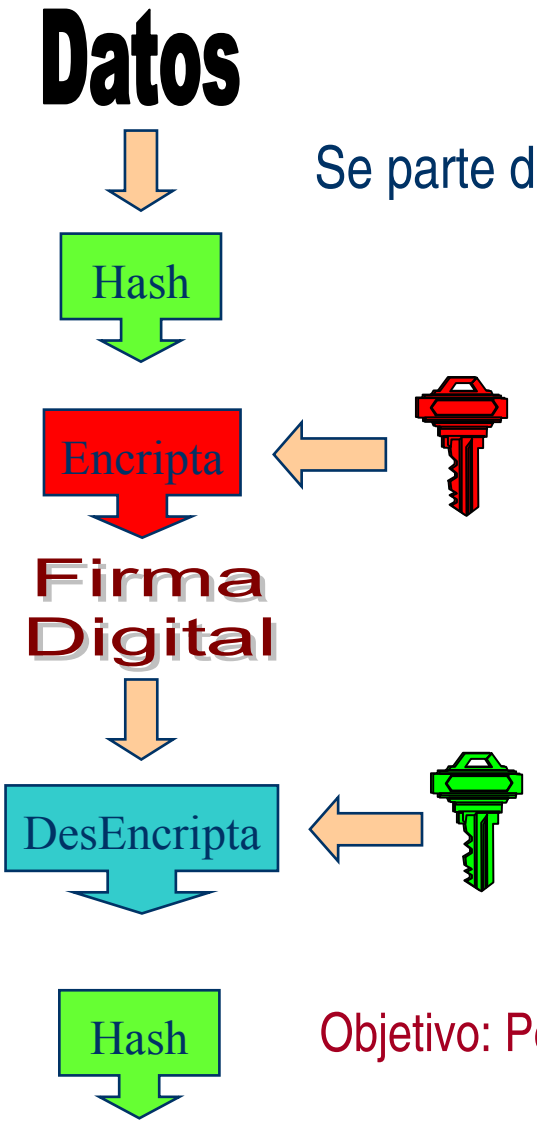
B necesita una información de A, y además …
B necesita que A no pueda negar que la información es de A

→ Resumen: B necesita que A ***firme*** le información que le envía

1. A encripta la info
   CON la clave privada de A
2. Cualquiera puede descifrar la info
   CON la clave pública de A (accesible)
3. Solo A puede haber encriptado la info pues
   Solo A conoce su clave privada
4. Esto garantiza que A es el autor de la info

Profundicemos un poco en las Firmas digitales →

Criptografía: Firmas Digitales (Digital Signatures)



Se parte de unos Datos y se calcula su “hash”

MI firma digital se crea encriptando el “hash” con
MI clave privada

MI firma digital solo puede ser desencriptada con
MI clave pública

Objetivo: Permite a cualquiera verificar que YO soy el creador de los Datos

Criptografía: Funciones Hash

Función Hash Criptográfica

$M \xrightarrow{H} h$

Función que usando como entrada un mensaje de longitud variable (M)
Produce como salida una cadena de longitud fija (h)

Propiedades:
- Dado M debe ser FÁCIL calcular H(M) = h
- Dado h debe ser DIFÍCIL calcular M = $H^{-1}(h)$
- Dado M debe ser DIFÍCIL encontrar M' tal que H(M) = H'(M)

Funciones Hash típicas

MD4 / MD5: Message Digest 4/5

Inventadas por Ronald L. Rivest (MIT) en 1990 y 1991

Generan un hash de 128 bis

Son estándares de Internet: RFC1320 y RFC1321

SHA: Secure Hash Algorithm

Relacionadas con la Agencia de Seguridad Nacional de EU

Publicadas por el NIST (National Institute of Standards and Technology)

Son estándares FIPS (Federal Information Processing Standards)
http://www.itl.nist.gov/fipspubs/

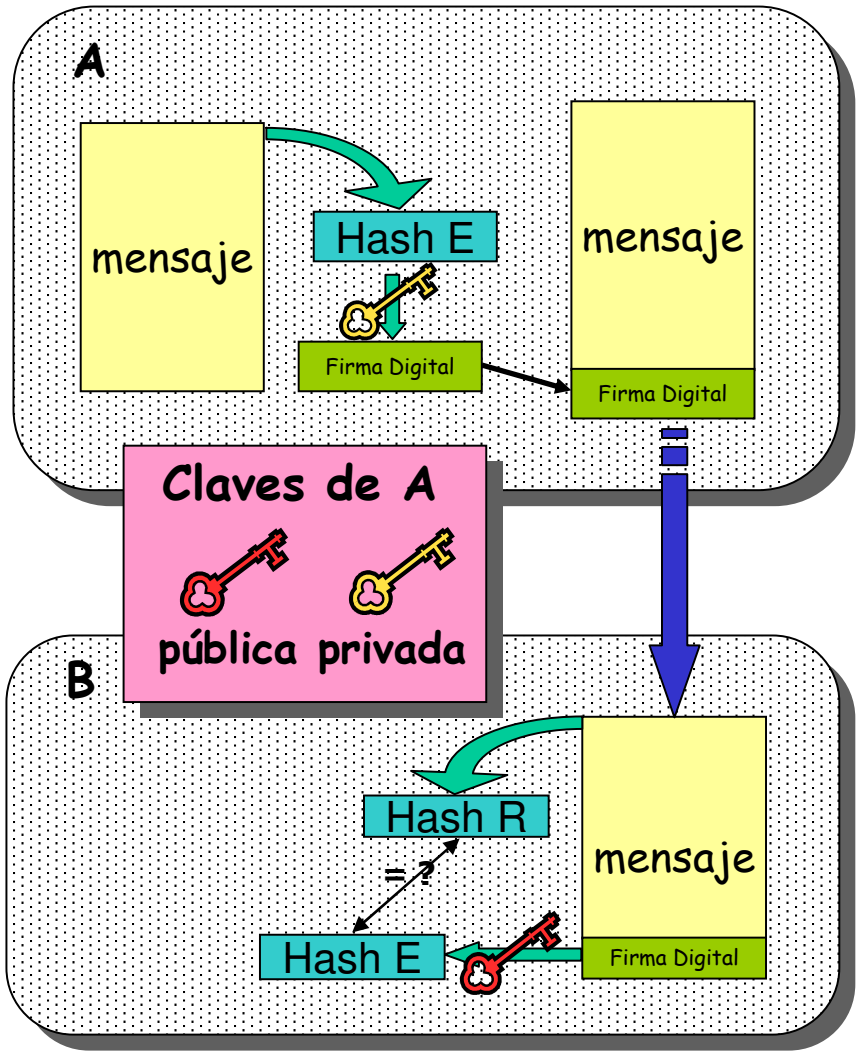
Alg. SHA	Año	Longit Entrada	Longit Salida
SHA-0	1993	$2^{64}-1$	160
SHA-1	1995	$2^{64}-1$	160
SHA-2	2001		
SHA-224		$2^{64}-1$	224
SHA-256		$2^{64}-1$	256
SHA-384		$2^{64}-1$	384
SHA-512		$2^{128}-1$	512
SHA-3	2010?		

MD6 Creada en 2008 por Rivest y otros

Enviada a la competición del NIST para el desarrollo de →

Firmas Digitales: No Repudiación y Control de Integridad

Se puede usar una firma digital para dos funciones:



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
6. COMPARACIÓN: ¿ Hash R == Hash E ?

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

Certificados Digitales: Necesidad

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

1. La clave privada de A no esta comprometida → Solo la conoce A
2. B tiene acceso a la clave pública de A

¿ 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 correspondencia:

Clave Pública ⟷ Identidad de su Propietario

Ambos, A y B, confían en esta tercera parte

Esta tercera parte se denomina Autoridad de Certificación (AC)
Certification Authorities (CAs)

Las ACs están integradas de una Infraestructura de Clave Pública

Infraestructura de Clave Pública

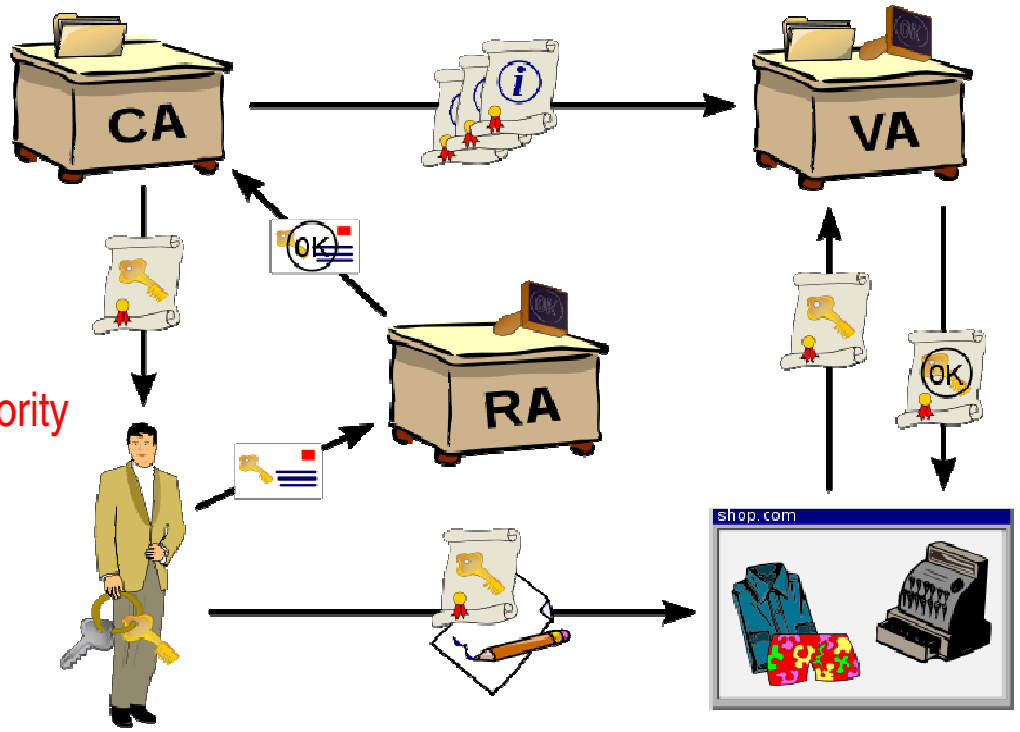
Infraestructura de Clave Pública ó PKI (Public Key Infrastructure)

Conjunto de hardware, software, personas, políticas y procedimientos necesarios para Crear, Gestionar, Distribuir, Almacenar y Revocar Certificados Digitales

Autoridad de Certificación
CA: Certification Authority

Autoridad de Validación
VA: Validation Authority

Autoridad de Registro
RA: Registration Authority



Estándar X.509 de Infraestructura de Clave Pública

El estándar X.509 de PKI es gestionado por la ITU-T (International Telecommunications Union)

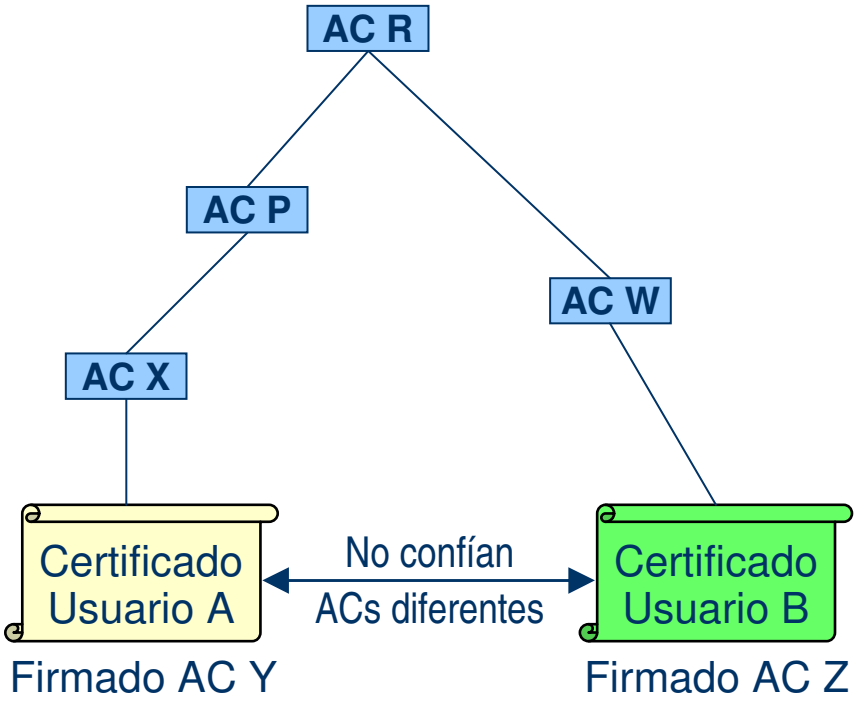
La estandarización es muy importante

Para que un sistema PKI sea efectivo y usable, los formatos y algoritmos que utiliza deben ser conocidos y accesibles por todo el mundo

El estándar X.509 especifica:

Formatos
- Certificados de clave pública
- Listas de revocación de certificados
- Certificados de autorizaciones

Algoritmo para validar cadenas de certificados



Asume una organización jerárquica de las Autoridades de Certificación

Ej.: Usuarios con certificados de CAs distintas

Ascender por el árbol de certificación hasta …

Punto más alto posible: Certificado Raíz (Sin firmar o Auto firmado)

Autoridades de Certificación

Las ACs emiten **Certificados Digitales** para usuarios, programas y máquinas

- Combinan una clave pública + la información del propietario
- Son firmados por la AC usando su certificado privado
- Pueden usar el certificado público de ACs para comprobar la integridad de certificados

Las ACs comprueban identidad y datos personales del solicitante de un certificado

- Las autoridades de Registros realizan la validación de los datos

Las ACs publican periódicamente una lista de certificados comprometidos

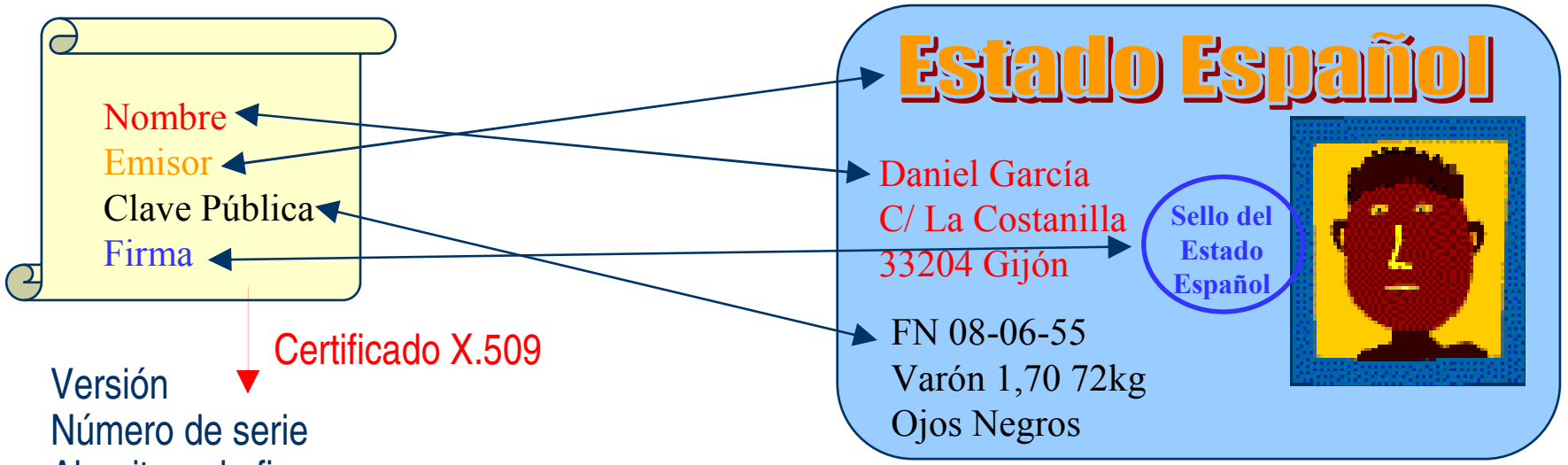
- Utilizan Listas de Revocación de Certificados Certificate Revocation List (CRL)
  Contienen todos los certificados que ya han expirado

Las ACs auto-firman sus propios certificados

Certificados Digitales

Son similares a un pasaporte, documento nacional de identidad, o carnet de conducir

Asocian un nombre con una clave pública 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)

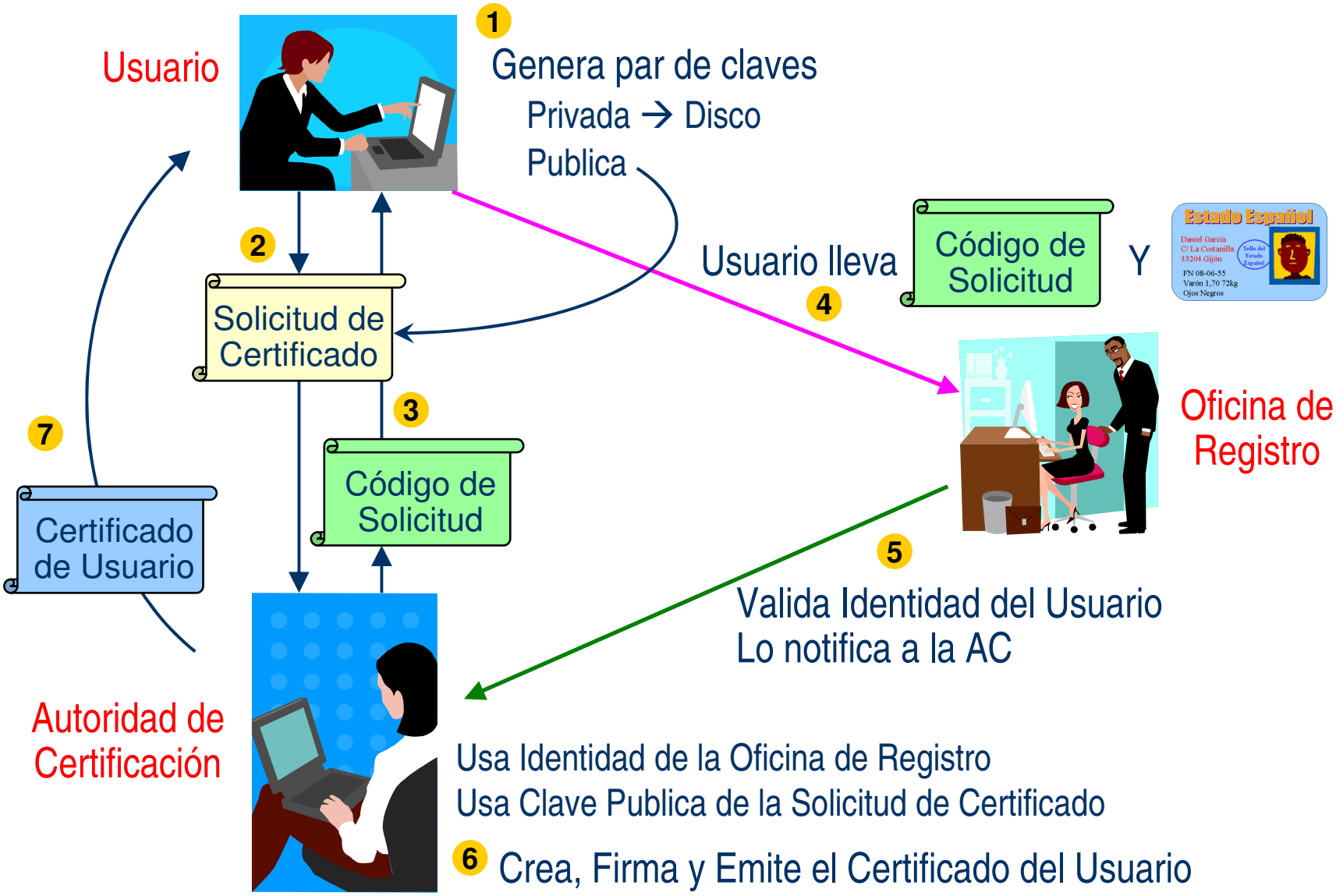
```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
            OU=Certification Services Division,
            CN=Thawte Server CA/emailAddress=server-certs@thawte.com
    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

Solicitud de Certificados



Ejemplo de Autenticación en GSI

Basada en X.509 PKI:

- Cada usuario/host/servicio tiene un certificado X.509
- Los certificados están firmados por autoridades de certificación locales
- Cada transacción Grid requiere autenticación mutua:
  1. A envía su certificado
  2. B verifica la firma en el certificado de A usando el certificado de la AC
  3. B envía a A una cadena de test
  4. A encripta la cadena de test con su clave privada
  5. A envía la cadena encriptada a B
  6. B usa la clave publica de A para desencriptar la cadena de test
  7. B compara la cadena desencriptada con la cadena original
  8. Si coinciden, B ha verificado la identidad de A; No puede repudiarle