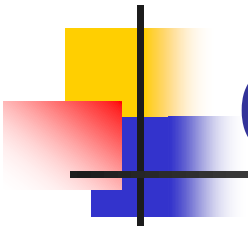
Bloque II
Sistemas de autenticación

Protocolos de autenticación

Seguridad en Redes de Comunicaciones
María Dolores Cano Baños



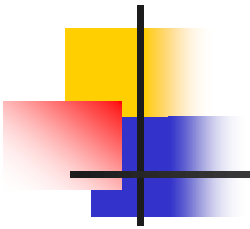
Contenidos

2.1 Sistemas de autenticación

2.2 Kerberos

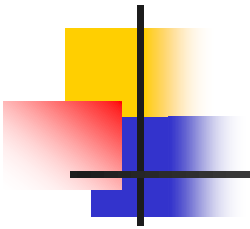
2.3 EAP

2.4 802.1x



Kerberos

- Proyecto Athena, Instituto de Tecnología de Massachussetts (MIT)
- Problema: ambiente distribuido abierto donde usuarios de estaciones de trabajo acceden a servicios de servidores distribuidos en red
- Amenazas:
  - Suplantar identidad
  - Alterar dirección de red
  - Escuchar y hacer *replay*



Kerberos

- Kerberos proporciona un servidor de autenticación centralizado (versiones 4 y 5)
- ¡Emplea sólo cifrado simétrico!
- Requerimientos
  - Seguro, un usuario que escucha no puede obtener información para suplantar a otro
  - Fiable, una falta de disponibilidad de Kerberos supone una falta de disponibilidad para todos los servicios que confían en él
  - Transparente, usuario no es consciente de autenticación más allá de que se le solicita una clave
  - Escalable, capaz de soportar un gran número de clientes y servidores

Kerberos

ARQUITECTURA

- Reino: dominio de administración (hasta 100.000)
- Modelo cliente/servidor
- Principales: son los usuarios, los clientes y los servicios de red ejecutándose en sistemas concretos.
  - Identificador de principal (40 caracteres máx.): nombre de principal, nombre de realización (sistema en el que se proporciona el servicio, papel del usuario, etc.) y nombre de reino (nombre de dominio de Internet en mayúsculas)
- Centro de distribución de claves: (KDC, Key Distribution Center), consta de servidor de autenticación (AS) y servidores de emisión de billetes (TGS)

Kerberos

ARQUITECTURA

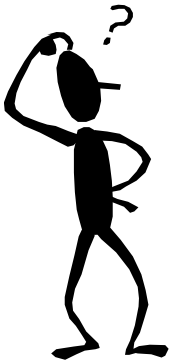
- KDC mantiene base de datos con una entrada por cada principal registrado en el reino.
- Por cada principal:
  - Identificador de principal
  - Clave maestra K de principal (o su clave si es un usuario)
  - Fecha expiración de la identidad
  - Fecha de última modificación del registro
  - Identidad del que modificó el registro por última vez
  - Tiempo de vida máximo de los billetes suministrados por el principal
  - Atributos
  - Datos sobre la implementación

Cifrada con K_{KDC}

Kerberos

Servidor de Autenticación (AS)

Servidor de Concesión de Billetes (TGS)

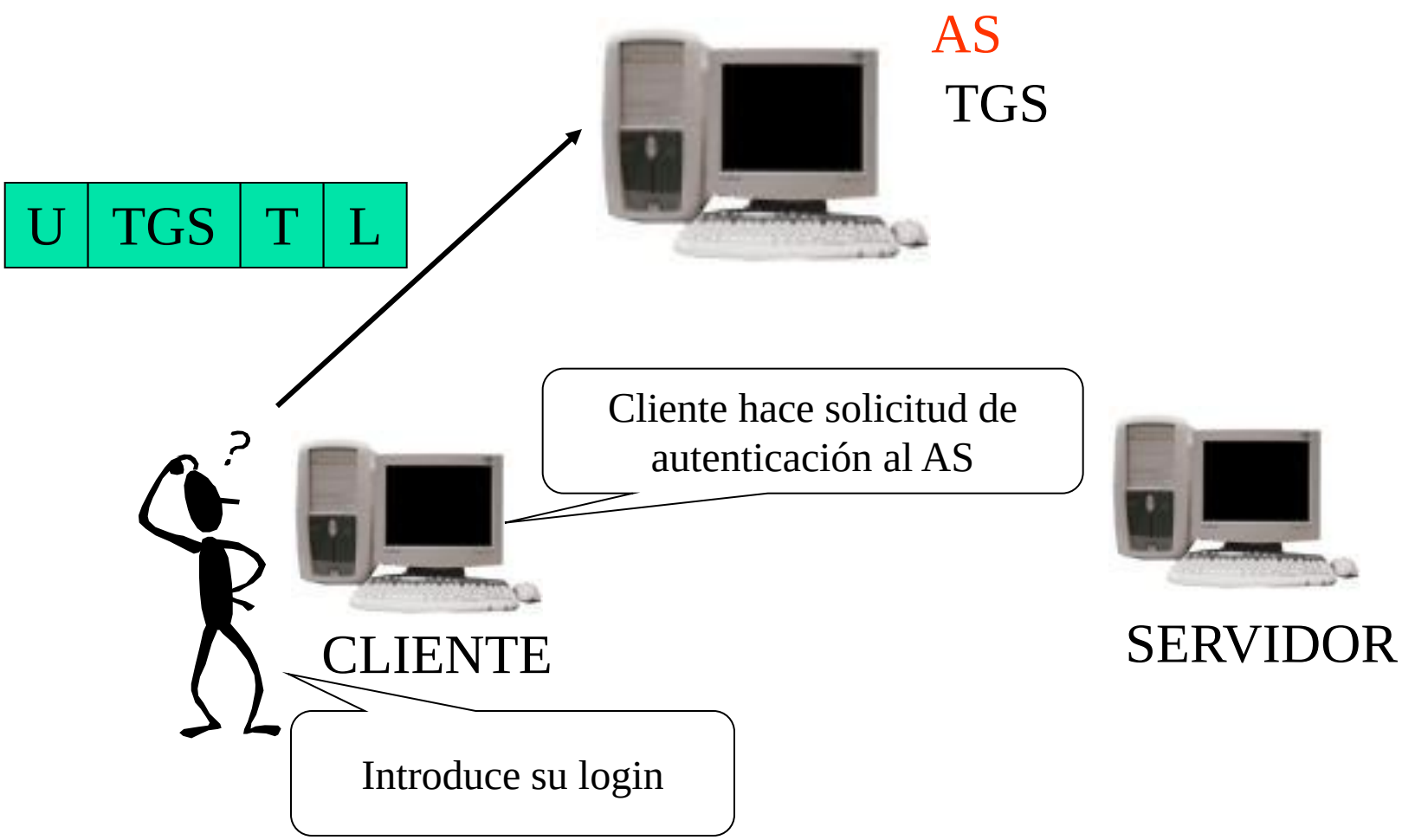


CLIENTE

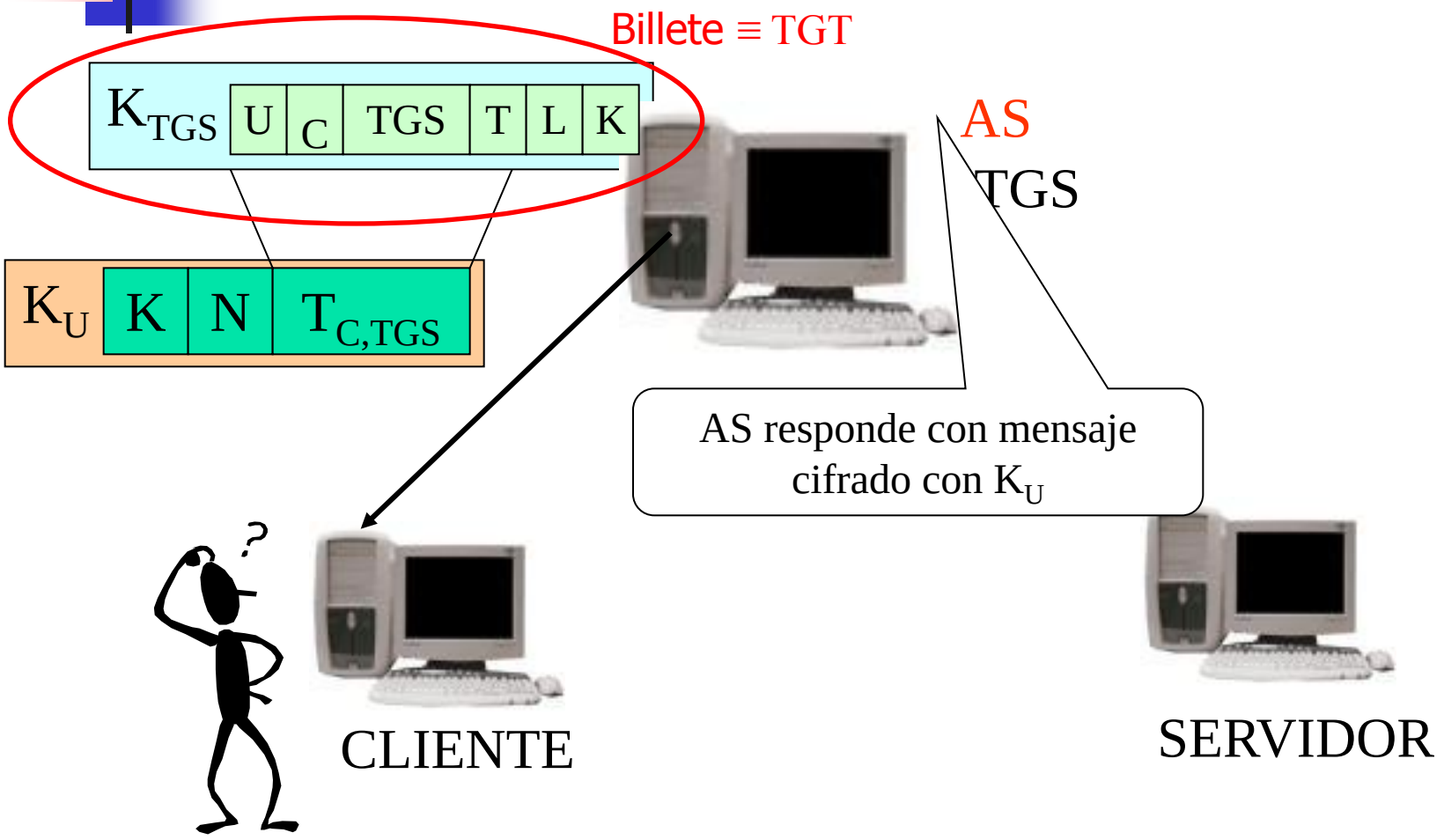
USUARIO

SERVIDOR

Kerberos


AS
TGS
U
TGS
T
L
Cliente hace solicitud de autenticación al AS
CLIENTE
SERVIDOR
Introduce su login

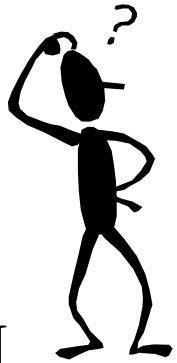
Kerberos


Billete ≡ TGT
K_{TGS} U C TGS T L K
K_U K N $T_{C,TGS}$
AS
TGS
AS responde con mensaje cifrado con K_U
CLIENTE
SERVIDOR

Kerberos

AS

TGS



El cliente conoce K, N y el TGT

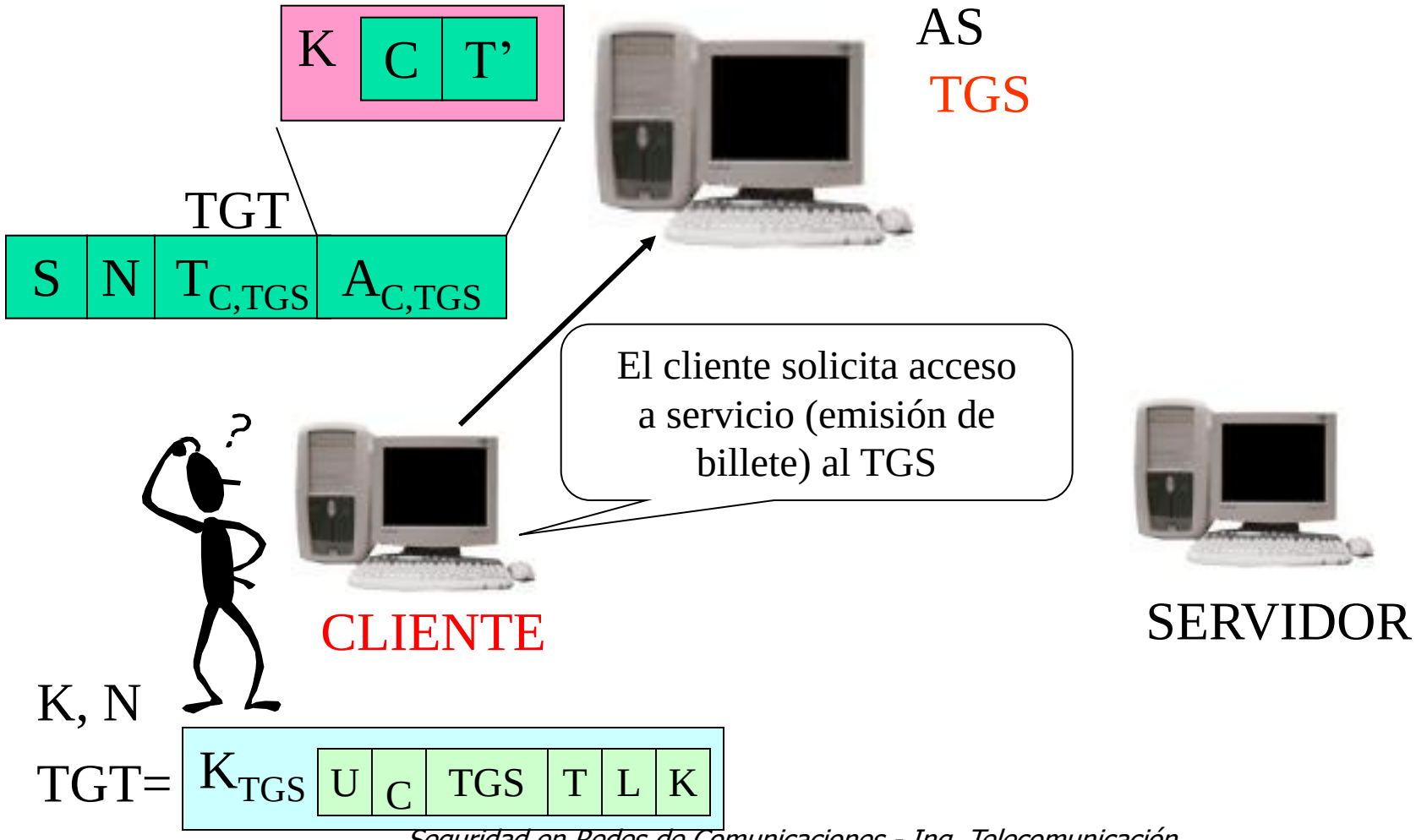
CLIENTE

SERVIDOR

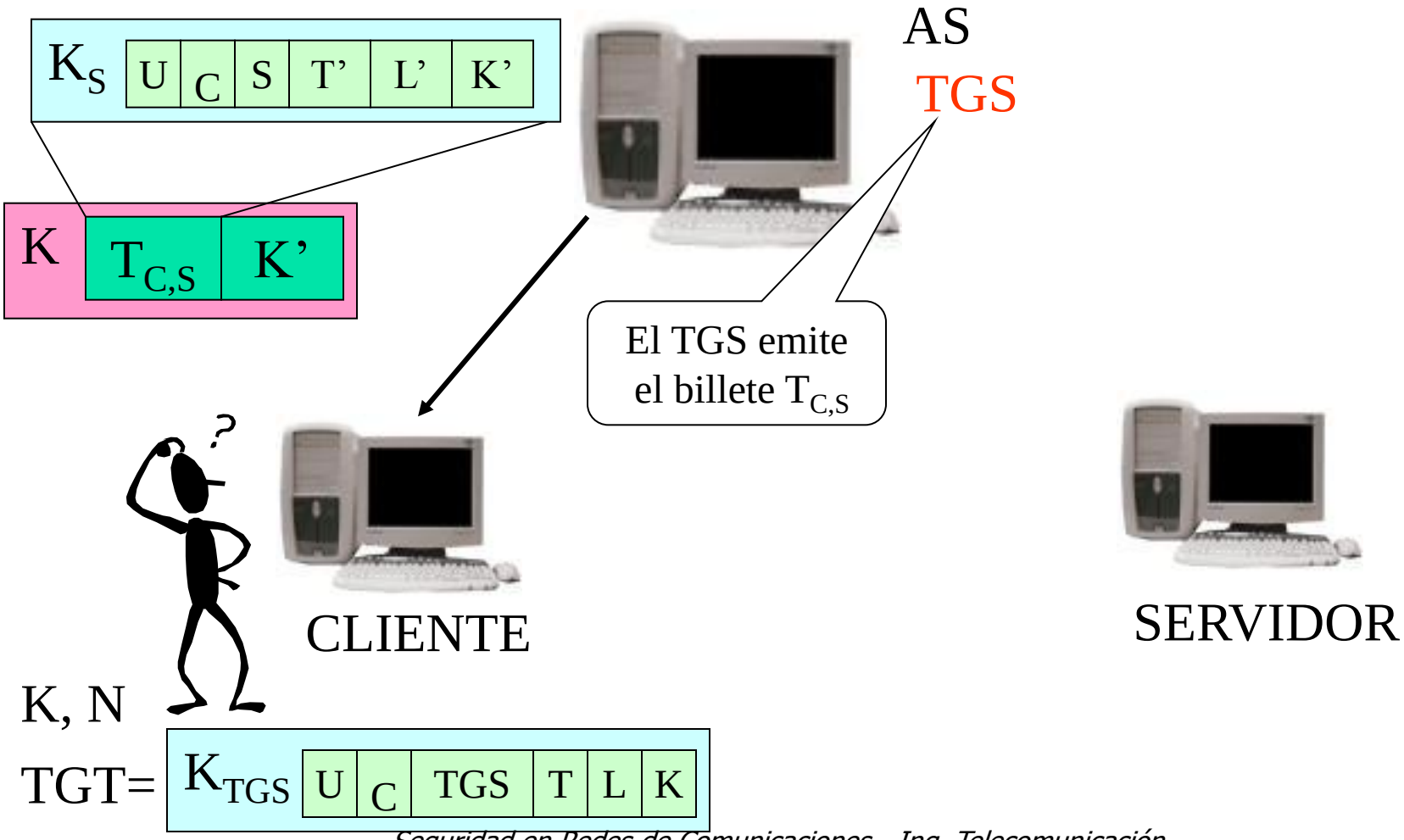
K, N

TGT= K_{TGS} | U | C | TGS | T | L | K |

Kerberos


K
C
T'
AS
TGS
TGT
S
N
$T_{C,TGS}$
$A_{C,TGS}$
El cliente solicita acceso a servicio (emisión de billete) al TGS
CLIENTE
SERVIDOR
K, N
TGT=
K_{TGS}
U
C
TGS
T
L
K

Kerberos


K_S U C S T' L' K'
K $T_{C,S}$ K'
AS
TGS
El TGS emite el billete $T_{C,S}$
CLIENTE
SERVIDOR
K, N
TGT= K_{TGS} U C TGS T L K

Kerberos

AS

TGS

El cliente ya tiene el billete y la clave k'

CLIENTE

SERVIDOR

$T_{C,S}$= K_S | U | C | S | T' | L' | K'

K', K , N

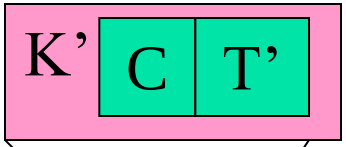
TGT= K_{TGS} | U | C | TGS | T | L | K

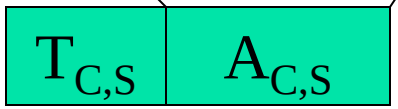
Kerberos

AS

TGS

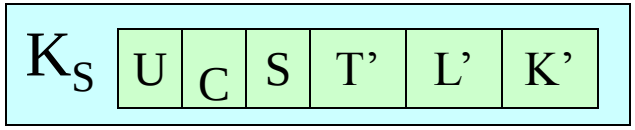
El cliente hace solicitud de aplicación



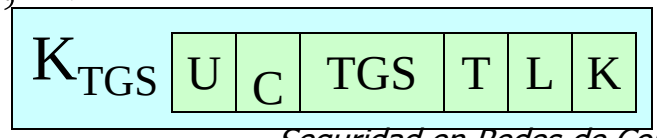


CLIENTE

SERVIDOR

$T_{C,S}$= 

K’, K , N

TGT= 

Kerberos

AS

TGS

CLIENTE

K’ | T’+1

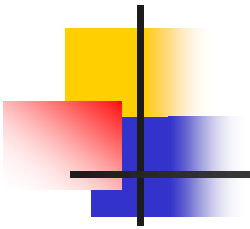
Servidor responde si se requiere autenticación mutua

SERVIDOR

$T_{C,S}$= | K_S | U | C | S | T’ | L’ | K’ |

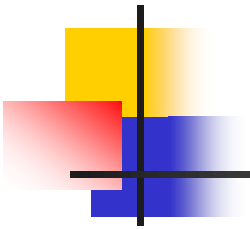
K’, K , N

TGT= | K_{TGS} | U | C | TGS | T | L | K |



Kerberos

- Deficiencias de Kerberos v4:
  - Dependencia del sistema de cifrado (DES)
  - Dependencia del protocolo IP
  - Tiempo de vida de los billetes (21 horas aprox.)
  - Nomenclatura de los principales
  - Autenticación entre reinos
  - Reenvío de la autenticación
  - Limitaciones técnicas: doble cifrado, cifrado PCBC (modo no estándar de DES), …



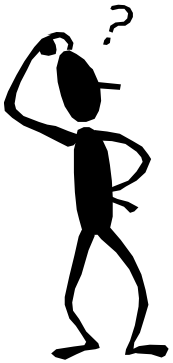
Kerberos

- Mejoras introducidas con Kerberos v5:
  - Identificadores de principales
  - Uso de cifrado
  - Direcciones de red
  - Ordenación de bytes
  - Operación entre reinos
  - Reenvío de autenticación

Kerberos

Servidor de Autenticación (AS)

Servidor de Concesión de Billetes (TGS)

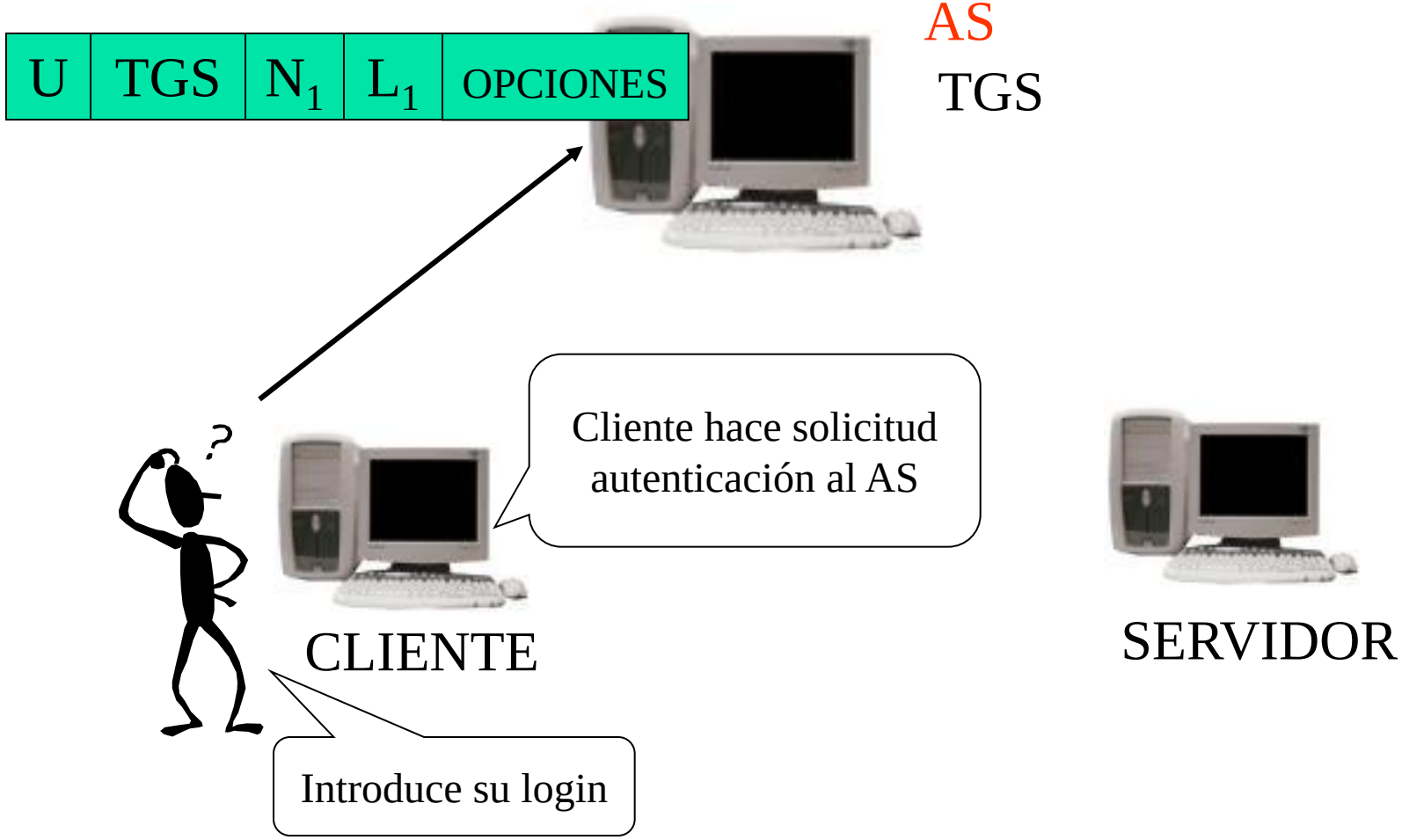


CLIENTE

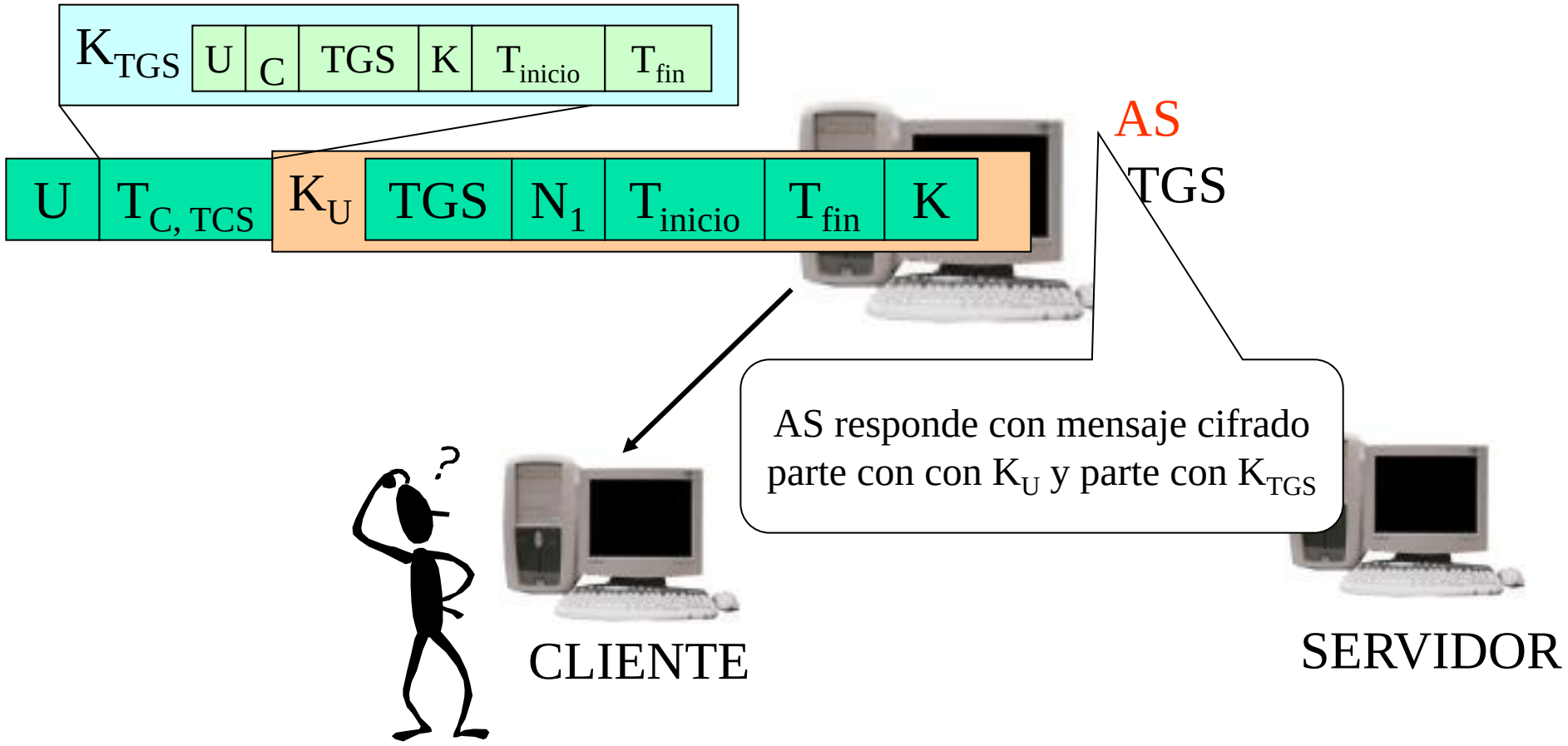
USUARIO

SERVIDOR

Kerberos


U
TGS
N_1
L_1
OPCIONES
AS
TGS
Cliente hace solicitud autenticación al AS
CLIENTE
SERVIDOR
Introduce su login

Kerberos


K_{TGS} U C TGS K T_{inicio} T_{fin}
U $T_{C, TCS}$ K_U TGS N_1 T_{inicio} T_{fin} K
AS
TGS
AS responde con mensaje cifrado parte con con K_U y parte con K_{TGS}
CLIENTE
SERVIDOR

Kerberos

AS
TGS

El cliente conoce K, N_1, T_{inicio}, T_{fin} y el TGT

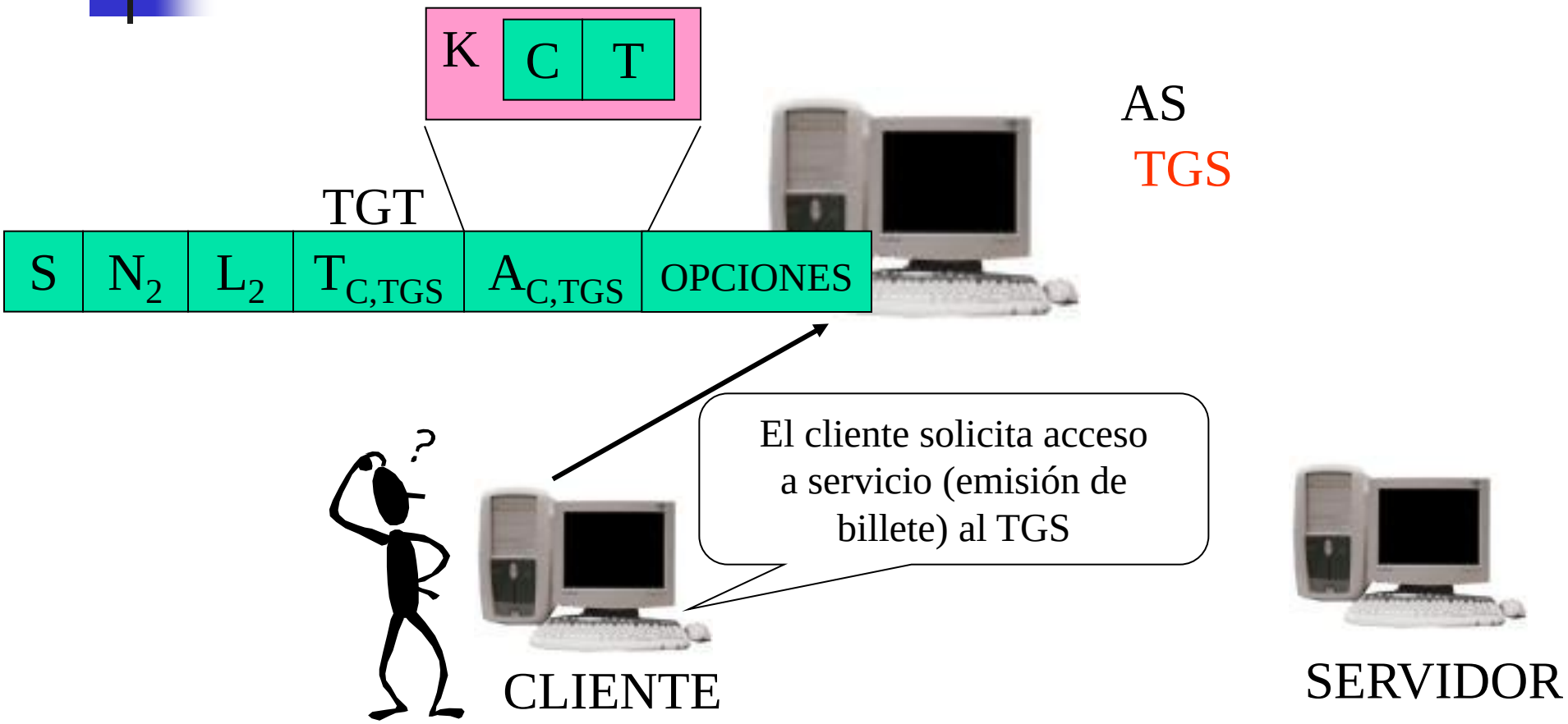
CLIENTE

SERVIDOR

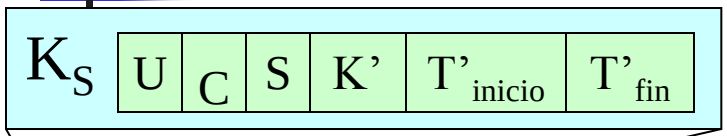
K, N_1, T_{inicio}, T_{fin}

TGT= K_{TGS} | U | C | TGS | K | T_{inicio} | T_{fin}

Kerberos

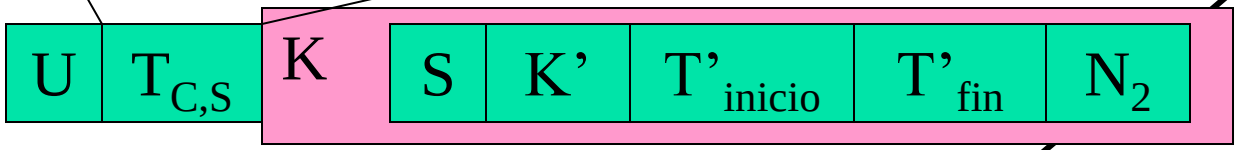

K
C
T
AS
TGS
TGT
S
N_2
L_2
$T_{C,TGS}$
$A_{C,TGS}$
OPCIONES
El cliente solicita acceso a servicio (emisión de billete) al TGS
CLIENTE
SERVIDOR
K, N_1, T_{inicio}, T_{fin}
TGT=
K_{TGS}
U
C
TGS
K
T_{inicio}
T_{fin}

Kerberos



AS

TGS

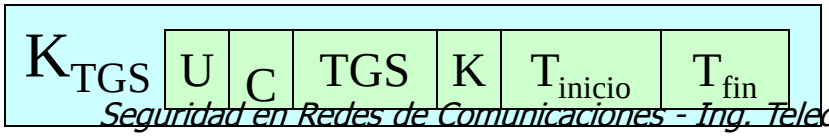


El TGS emite el billete $T_{C,S}$

CLIENTE

SERVIDOR

$K, N_1, T_{inicio}, T_{fin}$

TGT= 

Kerberos

AS
TGS

El cliente ya tiene el billete y la clave k'

CLIENTE

SERVIDOR

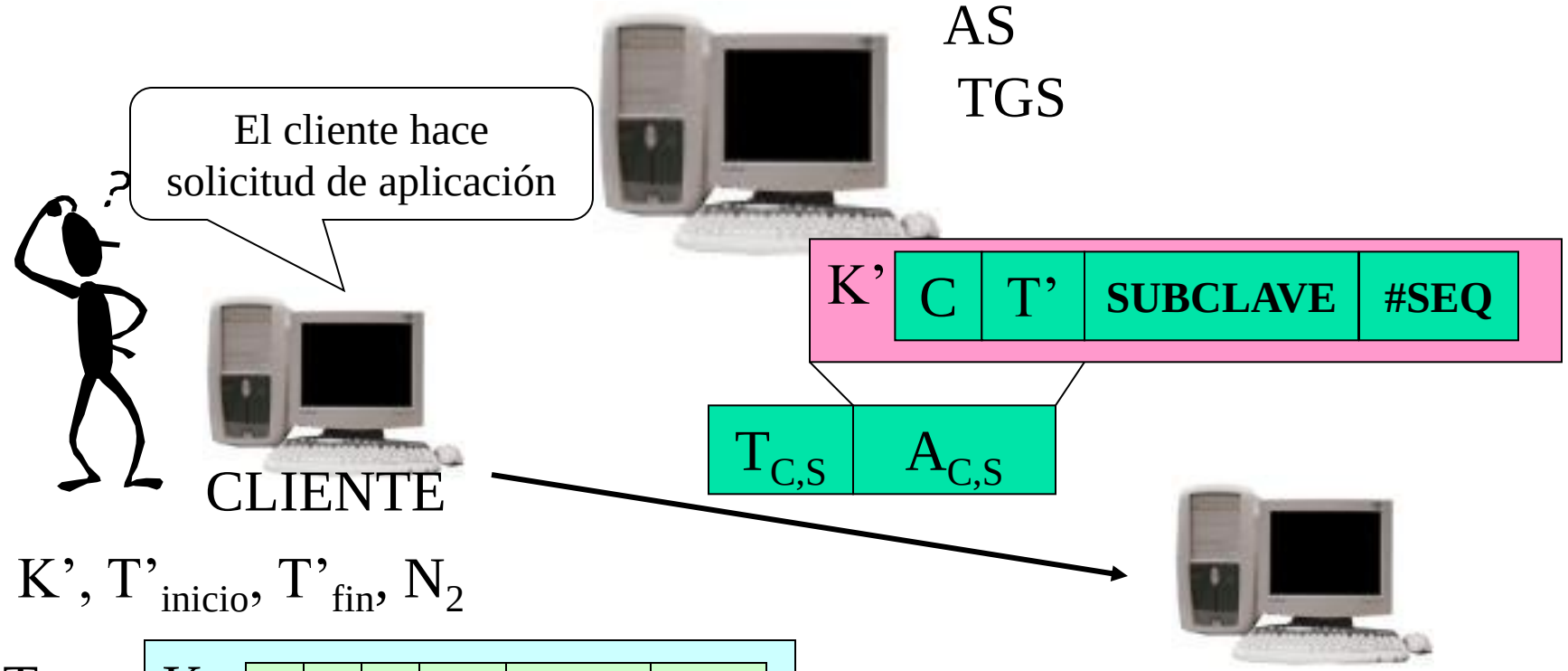
$K', T'_{inicio}, T'_{fin}, N_2$

$T_{C,S} = K_S$ | U | C | S | K' | T'_{inicio} | T'_{fin} |

$K, N_1, T_{inicio}, T_{fin}$

$T_{C,TGS} = K_{TGS}$ | U | C | TGS | K | T_{inicio} | T_{fin} |

Kerberos



$K', T'_{inicio}, T'_{fin}, N_2$

$T_{C,S}=$ K_S | U | C | S | K' | T'_{inicio} | T'_{fin} |

$K, N_1, T_{inicio}, T_{fin}$

$T_{C,\,TGS}=$ K_{TGS} | U | C | TGS | K | T_{inicio} | T_{fin} |

Kerberos

AS
TGS

CLIENTE

K’	T’	SUBCLAVE	#SEQ

Servidor responde si se requiere autenticación mutua

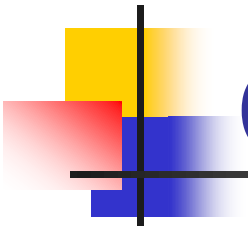
SERVIDOR

$K', T'_{inicio}, T'_{fin}, N_2$

$T_{C,S}$ = | K_S | U | C | S | K’ | T'_{inicio} | T'_{fin} |

$K, N_1, T_{inicio}, T_{fin}$

$T_{C, TGS}$ = | K_{TGS} | U | C | TGS | K | T_{inicio} | T_{fin} |



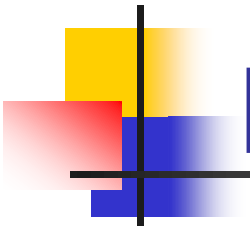
Contenidos

2.1 Sistemas de autenticación

2.2 Kerberos ✓

2.3 EAP

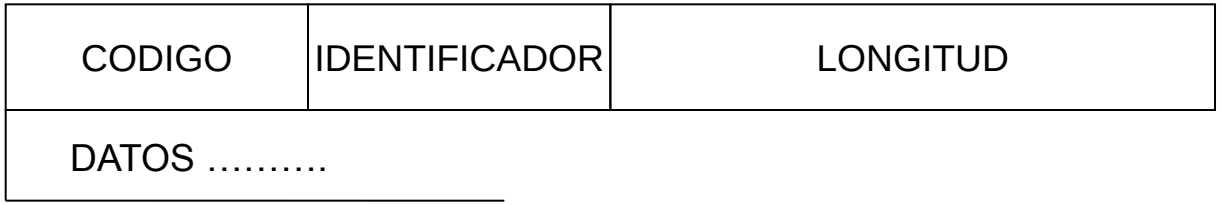
2.4 802.1x



EAP

- **EAP PPP** (Extensible Authentication Protocol) es un protocolo general de autenticación en PPP que soporta múltiples mecanismos de autenticación
- Funcionamiento general (RFC 2284):
  - Tras la fase de establecimiento del enlace, autenticador envía una o más peticiones (*Request*) para autenticar al otro extremo
  - El otro extremo responde a cada petición (*Response*)
  - El autenticador finaliza la fase de autenticación con un paquete de éxito (*Success*) o de fallo (*Failure*)
- Paquete EAP PPP encapsulado en el campo de información de trama PPP
  - (campo protocolo = 0xC227)

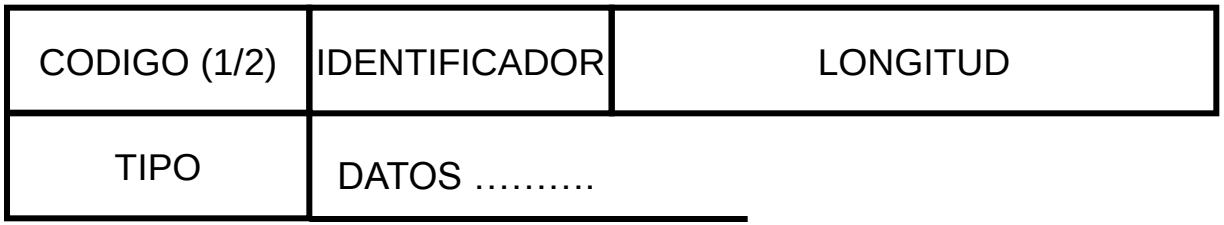
EAP

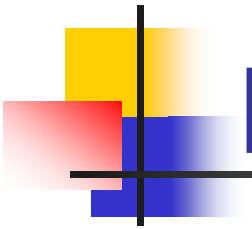


- CÓDIGO (1 byte): Identifica el tipo de paquete
  - 1 -> Request
  - 2 -> Response
  - 3 -> Success
  - 4 -> Failure
- IDENTIFICADOR (1 byte): Empareja respuestas con peticiones
- LONGITUD (2 byte): Longitud del paquete EAP incluyendo todos los campos
- DATOS (0 o más bytes): El formato de este campo viene determinado por el código

EAP

- PAQUETE REQUEST:
  - Paquete Request lo envía el autenticador al otro extremo
  - Cada Request tiene campo TIPO (1 byte) indicando qué se solicita
  - Contenido de campo datos varíable
- PAQUETE RESPONSE:
  - Sólo se envía paquete Response en respuesta a un paquete Request
  - Cada Response tiene campo TIPO que normalmente coincide con el del paquete Request
  - El valor del campo IDENTIFICADOR debe ser el mismo que el del paquete Request





EAP

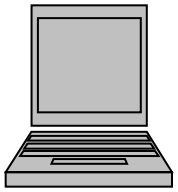
- Tipos de RESQUEST/RESPONSE:
  - 1 -> Identity, solicita identidad del otro extremo
  - 2 -> Notification, mensaje a mostrar en el otro extremo
  - 3 -> NAK, (sólo en Response) tipo de autenticación deseada es inaceptable
  - 4 -> MD5 Challenge
  - 5 -> One time password
  - 6 -> Generic Token Card
  - 13 -> Transport Layer Security

EAP

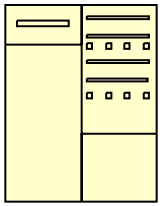
- PAQUETE SUCCESS/FAILURE:
  - El paquete Success reconoce una autenticación satisfactoria
  - Si el autenticador no puede autenticar al otro extremo envía un paquete Failure

CODIGO (3/4)	IDENTIFICADOR	LONGITUD

EAP

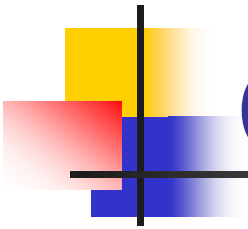


USUARIO SISTEMA FINAL



AUTENTICADOR

1. REQ (Identity)
2. RES (Identity)
3. REQ (Reto-MD5)
4. RES (Nak, Reto-MD5)
5. REQ (Generic Token Card)
6. RES (Generic Token Card)
7. REQ (Generic Token Card)
8. RES (Generic Token Card)
9. SUCCESS



Contenidos

2.1 Sistemas de autenticación

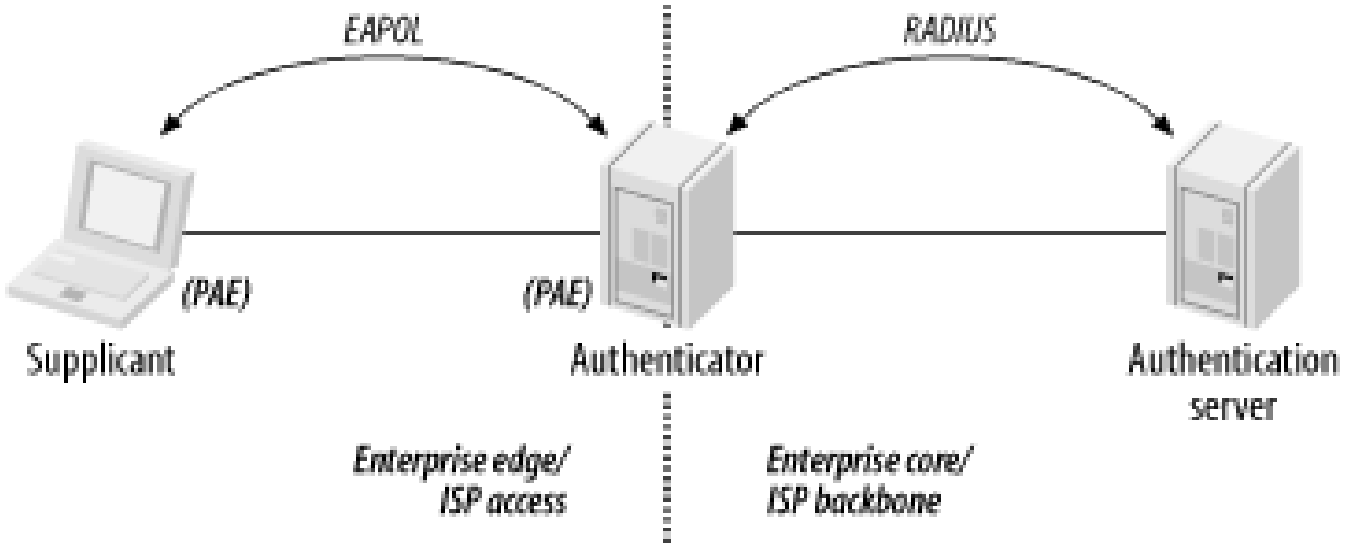
2.2 Kerberos ✓

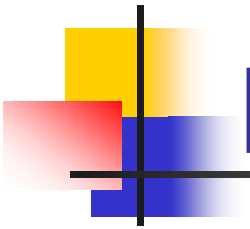
2.3 EAP ✓

2.4 802.1x

EAP: 802.1x

- Estándar de autenticación donde se definen:
  - Suplicante
  - Autenticador
  - Servidor de autenticación
- Entidades de autenticación por puerto (Port Authentication Entities, PAEs)
  - Suplicante y Autenticador



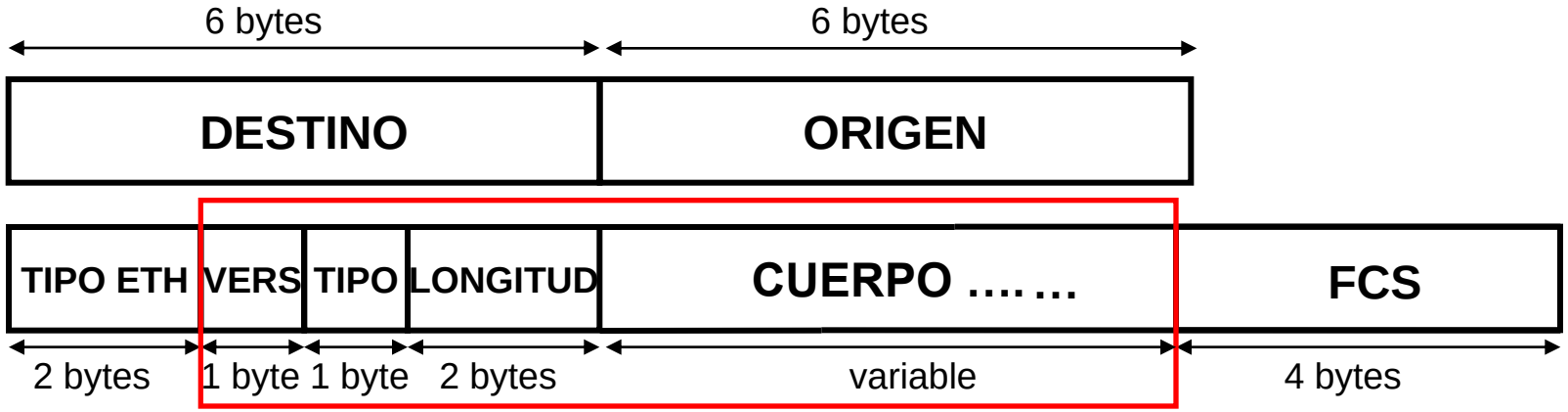


EAP: 802.1x

- Intercambio de autenticación entre suplicante y servidor de autenticación, autenticador hace de puente entre ambos
  - EAPOL (EAP Over LAN) o EAPOW (EAP Over Wireless)
  - RADIUS (Remote Authentication Dial In User Service)
- Ventaja: cambios en el método de autenticación no requieren hacer cambios complejos en el sistema final ni en la infraestructura de la red

EAP: 802.1x

- Formato de trama EAPOL



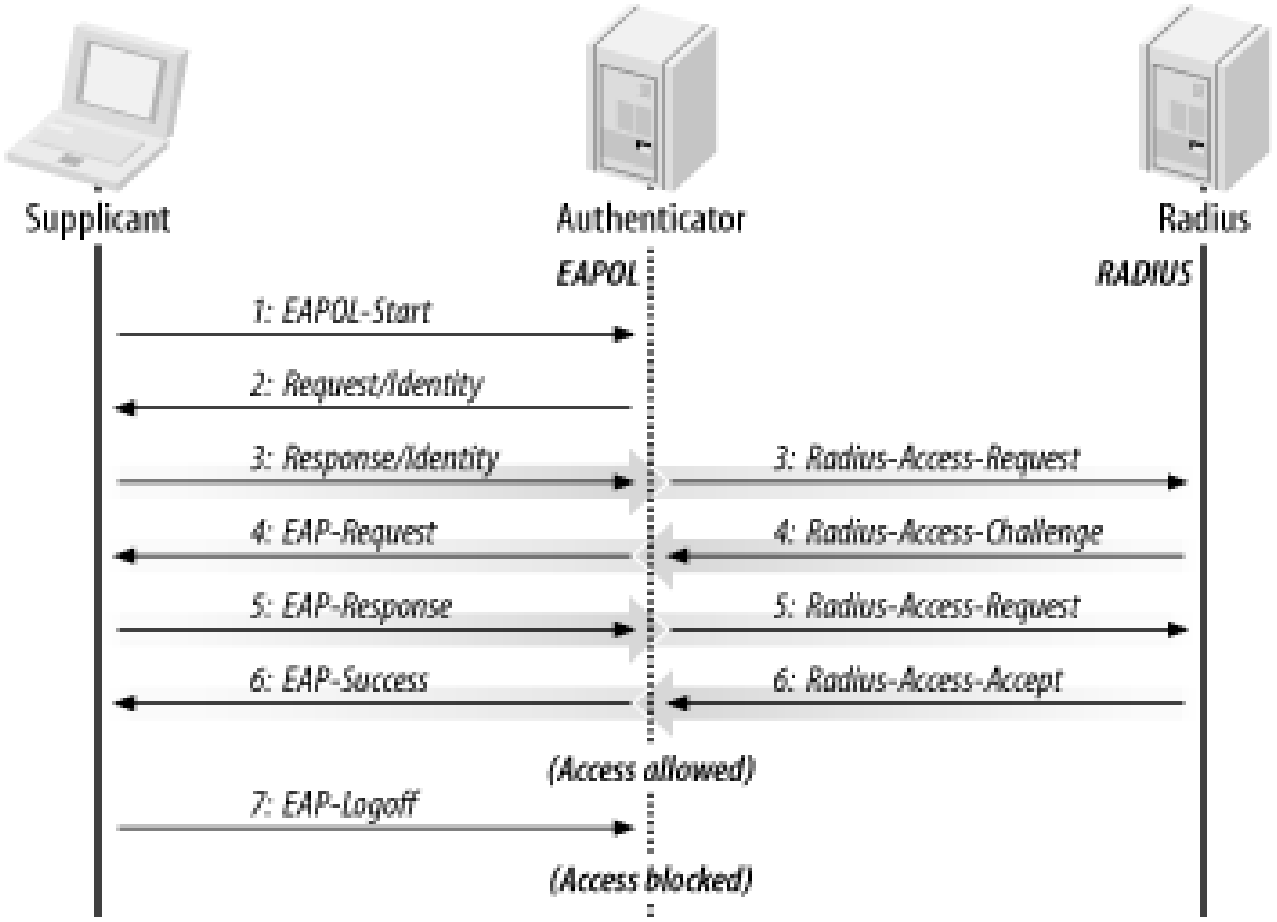
EAP: 802.1x

- DESTINO/ORIGEN (Cabecera MAC)
  - Direcciones MAC origen y destino
  - En LAN de medio compartido los suplicante envían los mensajes a la MAC 01:80:C2:00:00:03
  - En redes 802.11 los puertos no existen como tales, EAPOW se ejecuta tras proceso de asociación entre suplicante y autenticador
- TIPO ETHERNET
  - Código asignado a EAPOL 88:8E
- VERSION
  - Actualmente sólo existe la versión 1

EAP: 802.1x

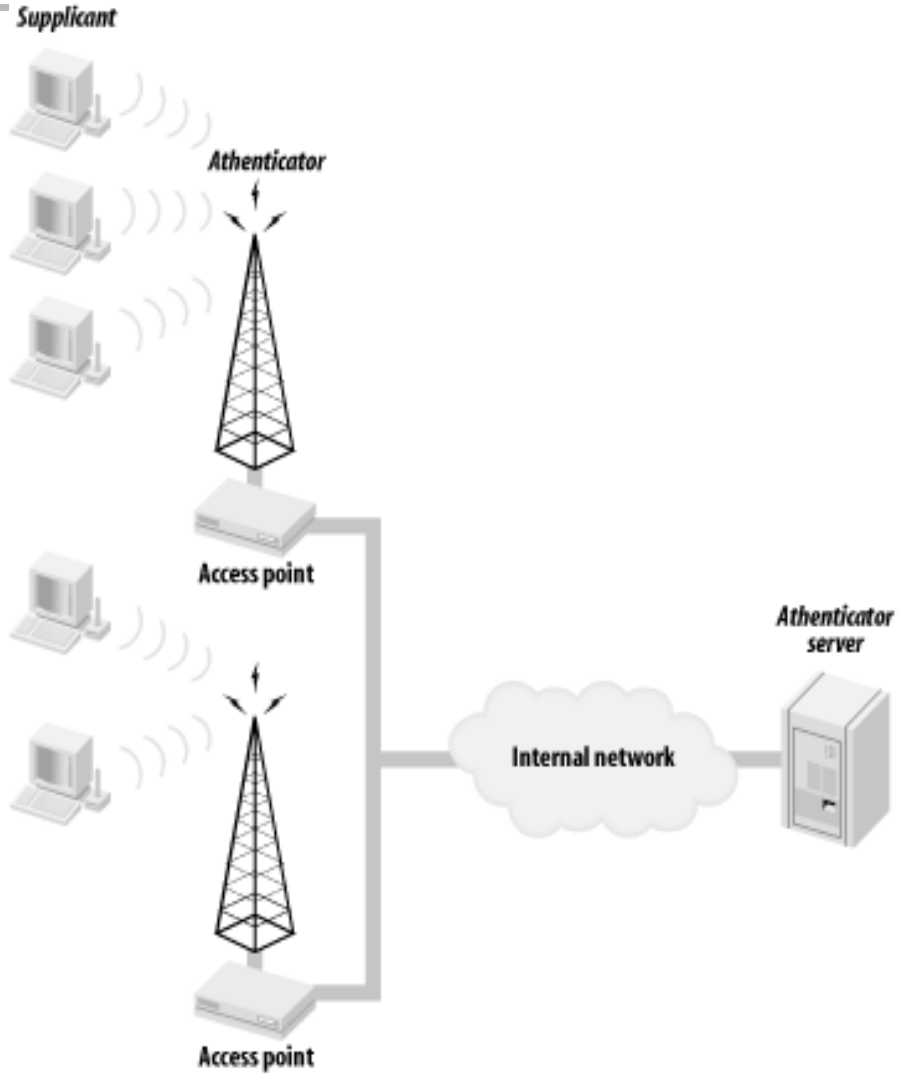
- TIPO PAQUETE
  - 0x00 PAQUETE EAP
  - 0x01 EAPOL START
  - 0x02 EAPOL LOGOFF
  - 0x03 EAPOL KEY
  - 0x04 EAPOL ENCAPSULATED ASF ALERT
- LONGITUD
  - Longitud del campo CUERPO en bytes
- CUERPO
  - Campo que encapsula un paquete EAP, un EAPOL KEY o una alerta EAPOL ENCAPSULATED ASF ALERT

EAP: 802.1x

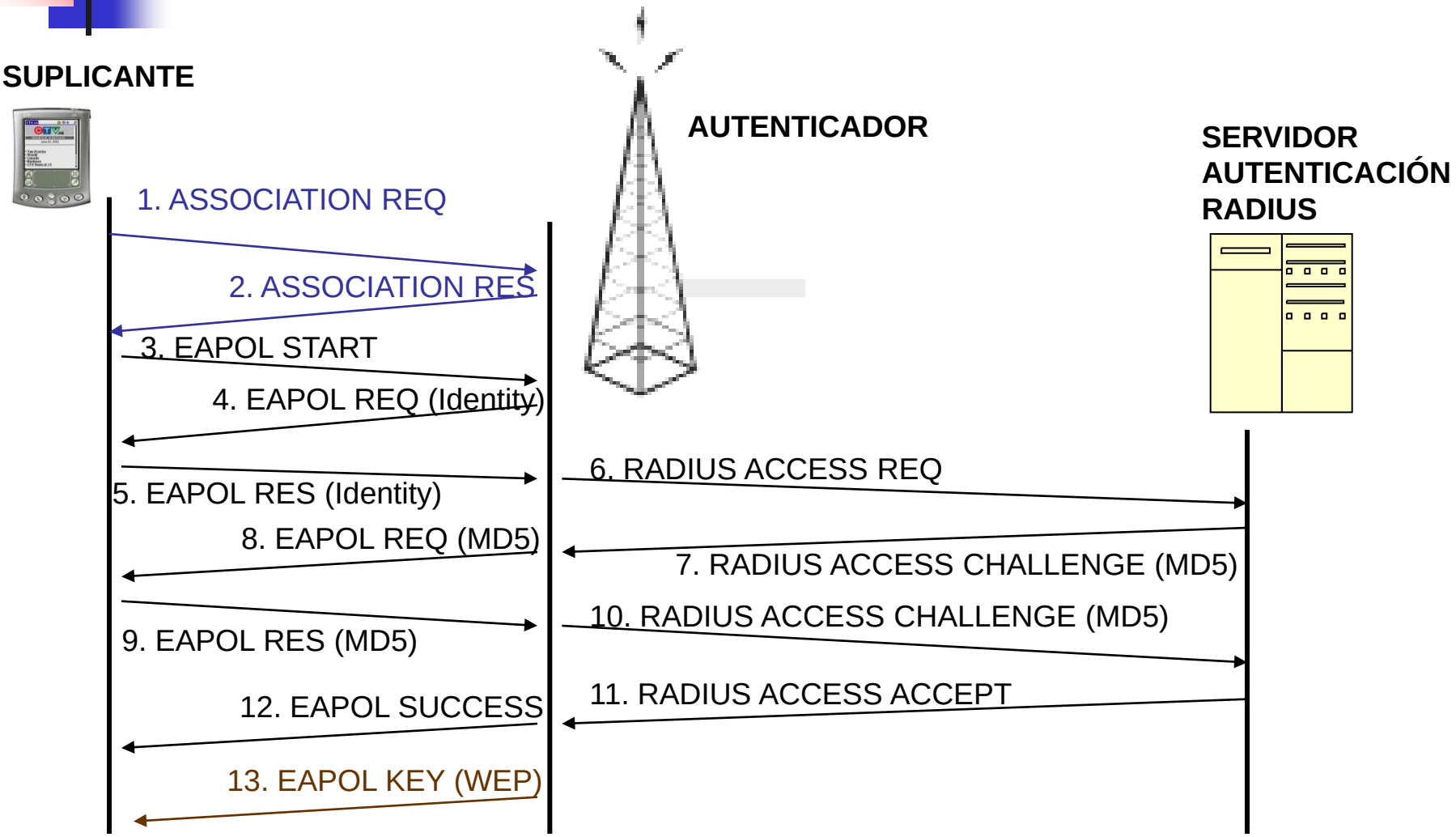

Supplicant
Authenticator
Radius
EAPOL
RADIUS
1: EAPOL-Start
2: Request/Identity
3: Response/Identity
3: Radius-Access-Request
4: EAP-Request
4: Radius-Access-Challenge
5: EAP-Response
5: Radius-Access-Request
6: EAP-Success
6: Radius-Access-Accept
(Access allowed)
7: EAP-Logoff
(Access blocked)

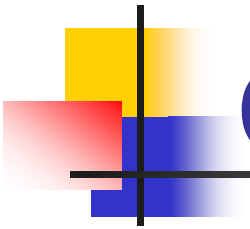
EAP: 802.1x

- En redes inalámbricas "asociación entre estación móvil y punto de acceso" ≅ "puerto lógico"
- El punto de acceso descarta todo el tráfico hasta que autenticación satisfactoria
- Trama EAPOL KEY se puede emplear para distribución de claves dinámicas en WEP (Wired Equivalent Privacy)



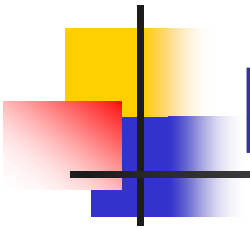
EAP: 802.1x


SUPLICANTE
AUTENTICADOR
SERVIDOR AUTENTICACIÓN RADIUS
1. ASSOCIATION REQ
2. ASSOCIATION RES
3. EAPOL START
4. EAPOL REQ (Identity)
5. EAPOL RES (Identity)
6. RADIUS ACCESS REQ
7. RADIUS ACCESS CHALLENGE (MD5)
8. EAPOL REQ (MD5)
9. EAPOL RES (MD5)
10. RADIUS ACCESS CHALLENGE (MD5)
11. RADIUS ACCESS ACCEPT
12. EAPOL SUCCESS
13. EAPOL KEY (WEP)



Contenidos

- 3.1 Introducción ✓
- 3.2 Funciones HASH y MAC
  - 3.2.1 MD5 ✓
  - 3.2.2 SHA ✓
  - 3.2.3 HMAC ✓
- 3.3 Sistemas de autenticación
  - 3.3.1 Kerberos ✓
  - 3.3.2 EAP ✓
    - -802.1x ✓
- 3.4 Firma digital
  - 3.4.1 Certificados



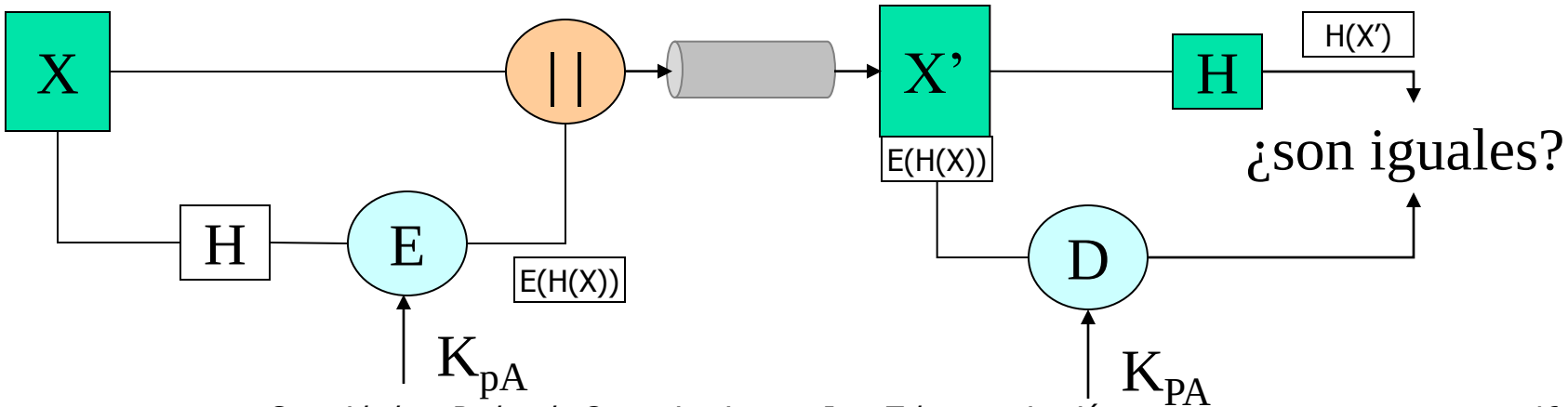
Firma digital

- Criptografía de clave pública (asimétrica)

- La FIRMA DIGITAL debe tener las siguiente propiedades:
  - Poder verificar autor, fecha y hora de la firma
  - Poder autenticar el contenido del mensaje a la hora en la que se firmó
  - Debe estar verificada por un tercero para evitar disputas

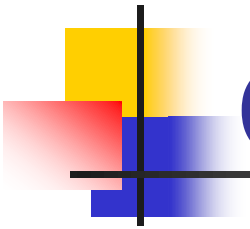
Firma digital

- Cualquier FIRMA DIGITAL:
  - Firma ≡ patrón de bits dependientes del mensaje firmado
  - Utilizará información única del emisor para evitar denegación y falsificación
  - Sencilla de crear
  - Sencilla de reconocer y verificar
  - Falsificarla debe ser computacionalmente no factible



Firma Digital

- Firma Digital Directa
  - Sólo intervienen los dos comunicantes
  - Destino conoce clave pública de emisor
  - Firmamos mensaje completo ó hash del mensaje con K_p emisor
  - Problema: seguridad de la clave secreta
- Firma Digital Arbitrada
  - Una tercera entidad actúa como árbitro
  - Funcionamiento general: todos los mensajes pasan por el árbitro que comprueba la validez de origen y contenido
  - Confiabilidad total en el árbitro

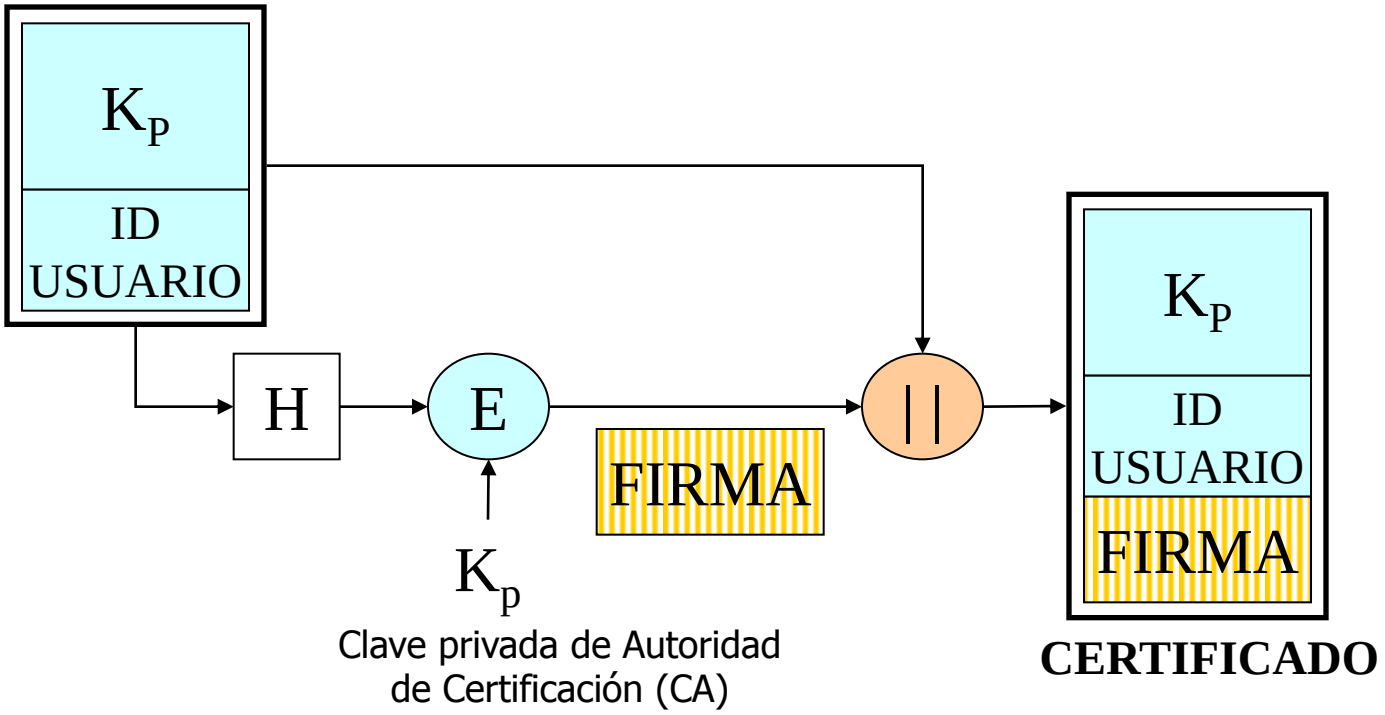


Contenidos

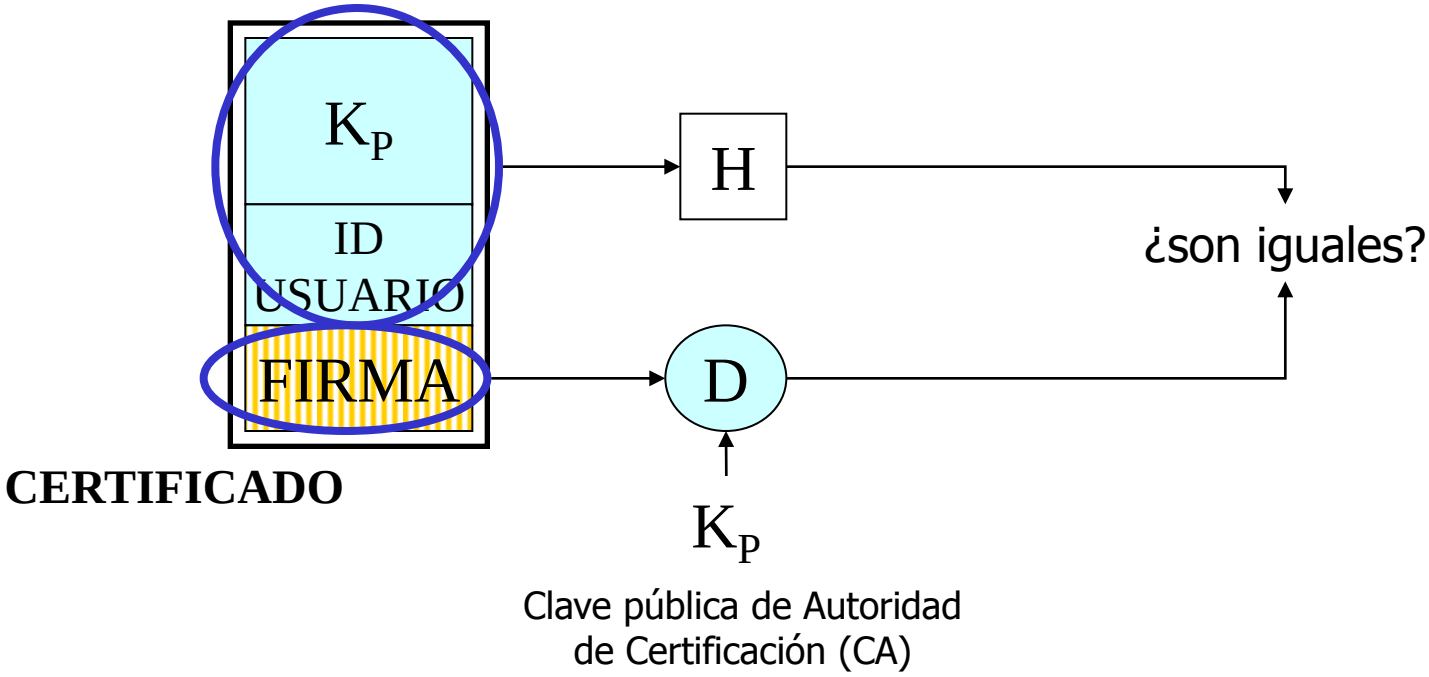
- 3.1 Introducción ✓
- 3.2 Funciones HASH y MAC
  - 3.2.1 MD5 ✓
  - 3.2.2 SHA ✓
  - 3.2.3 HMAC ✓
- 3.3 Sistemas de autenticación
  - 3.3.1 Kerberos ✓
  - 3.3.2 EAP ✓
    - -802.1x ✓
- 3.4 Firma digital
  - 3.4.1 Certificados

Certificados

- Las claves públicas han de ser "públicas" ⇨ ¿problema de suplantación?
  - Solución: certificados de clave pública

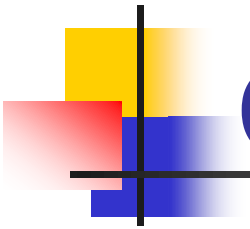


Certificados



Certificados

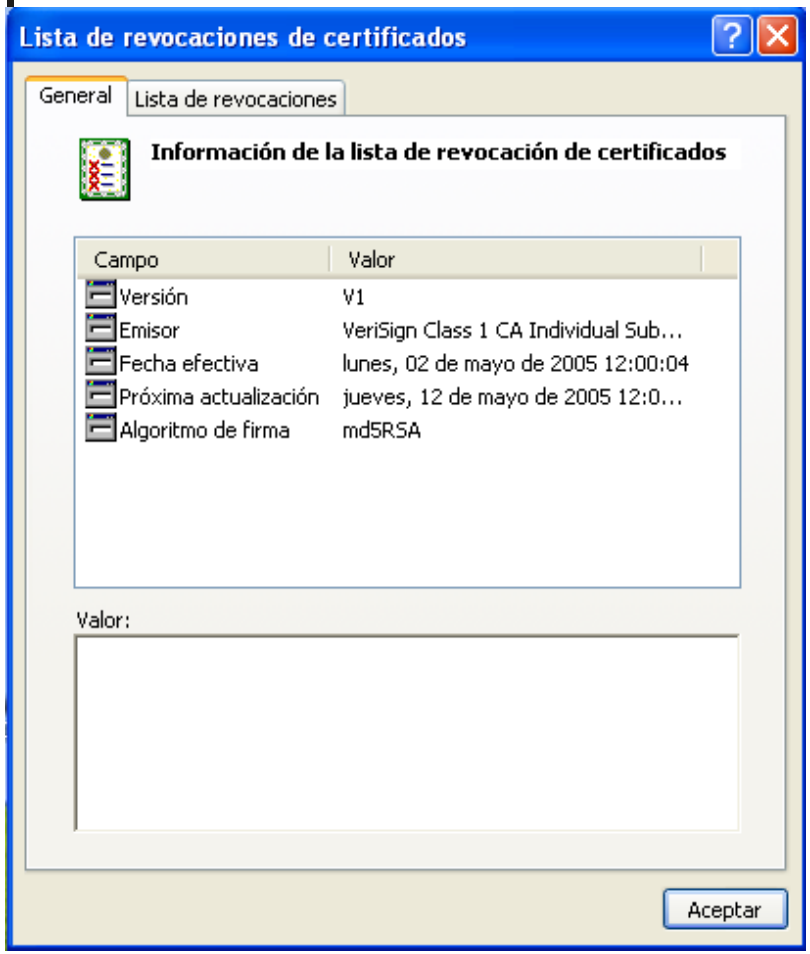
- Servicio ofrecido por las CA
  - CA interna, certificar a sus propios empleados, puestos y niveles de autoridad
  - CA externa de empleados, empresa contrata a otra para certificar a sus empleados
  - CA externa de clientes, empresa contrata a otra para que certifique a sus clientes
  - **CA confiable de terceros**, compañía o gobierno opera una CA que relaciona claves públicas con nombres legales de individuos o empresas

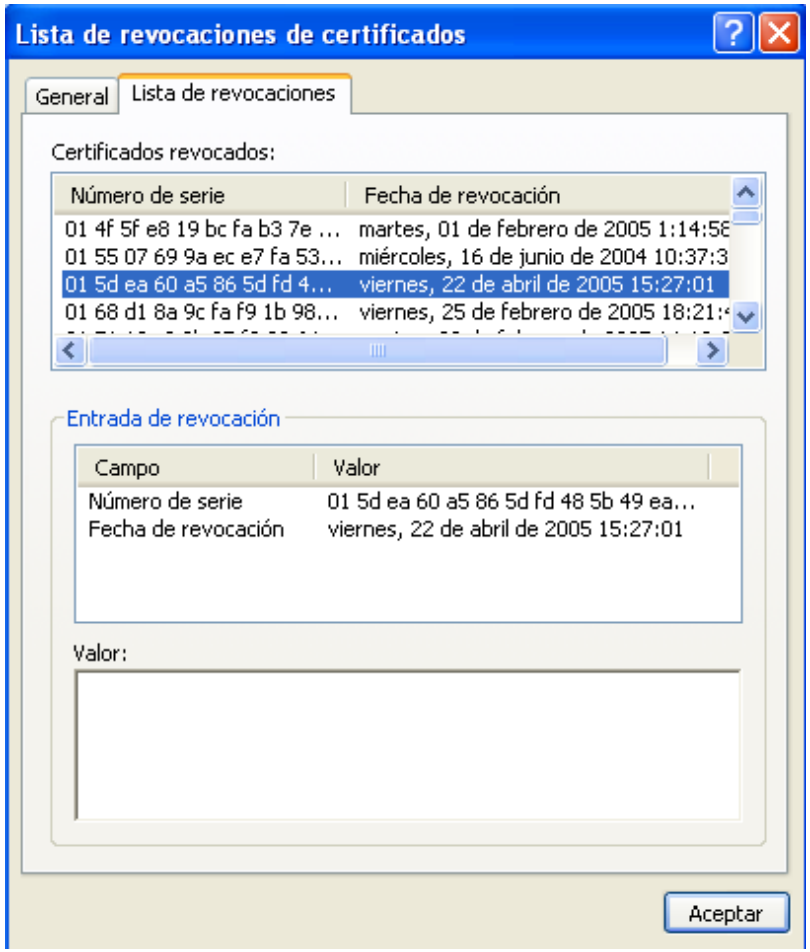


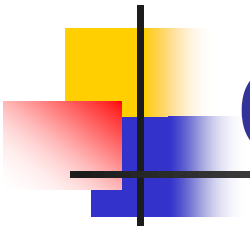
Certificados

- Revocación de certificados:
  - Clave privada de usuario comprometida
  - CA emite certificado a entidad incorrecta
  - El usuario cambia de CA
  - Violación de la seguridad de la CA
- **Lista de revocación de certificados** (CRL, Certification Revocation List)
  - Ejemplo: http://crl.verisign.com/

Certificados







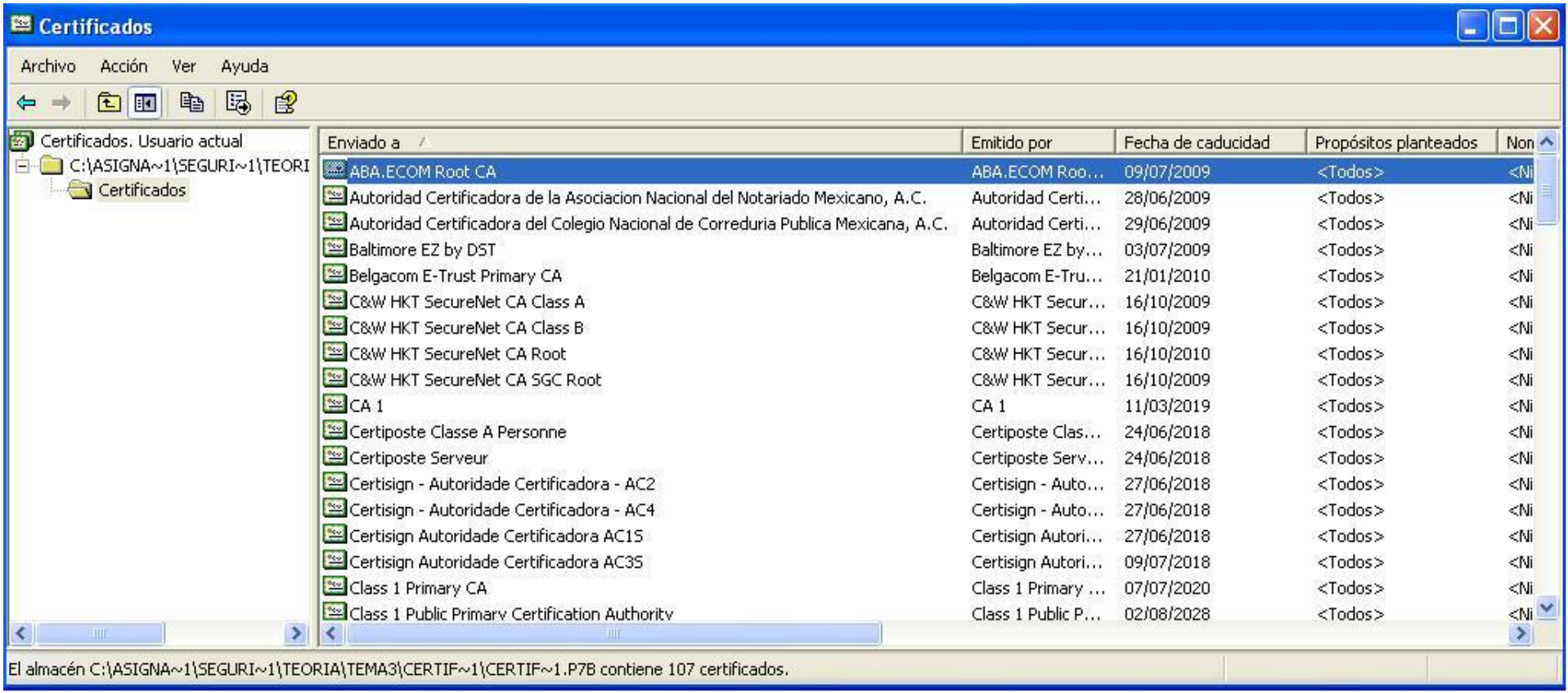
Certificados

- Certificados de autoridades certificadoras
- Certificados de servidores
- Certificados personales
- Certificados de editor de software

Certificados

CERTIFICADO DE AUTORIDAD CERTIFICADORA

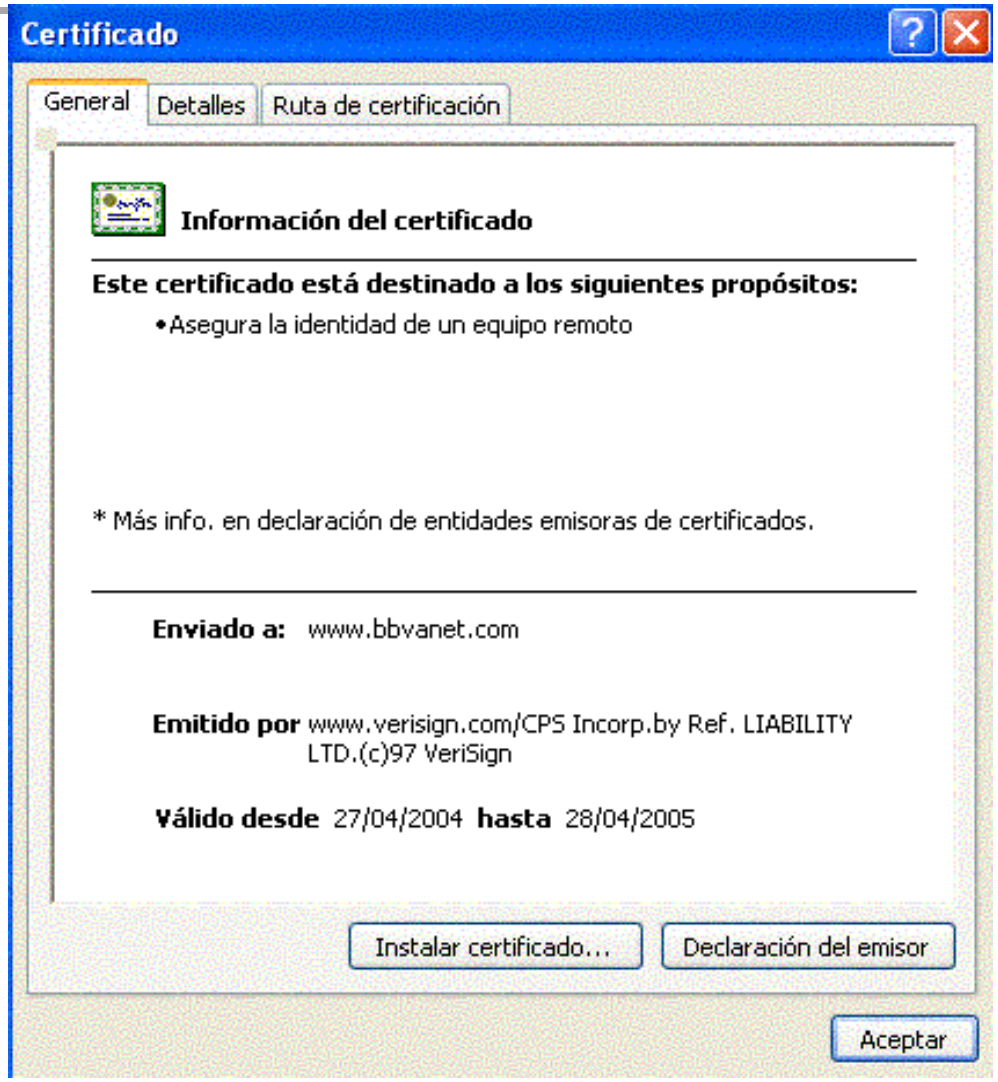
- Nombre y clave pública de la CA
- Pueden ser autofirmados
- PKI (Public Key Infrastructure)



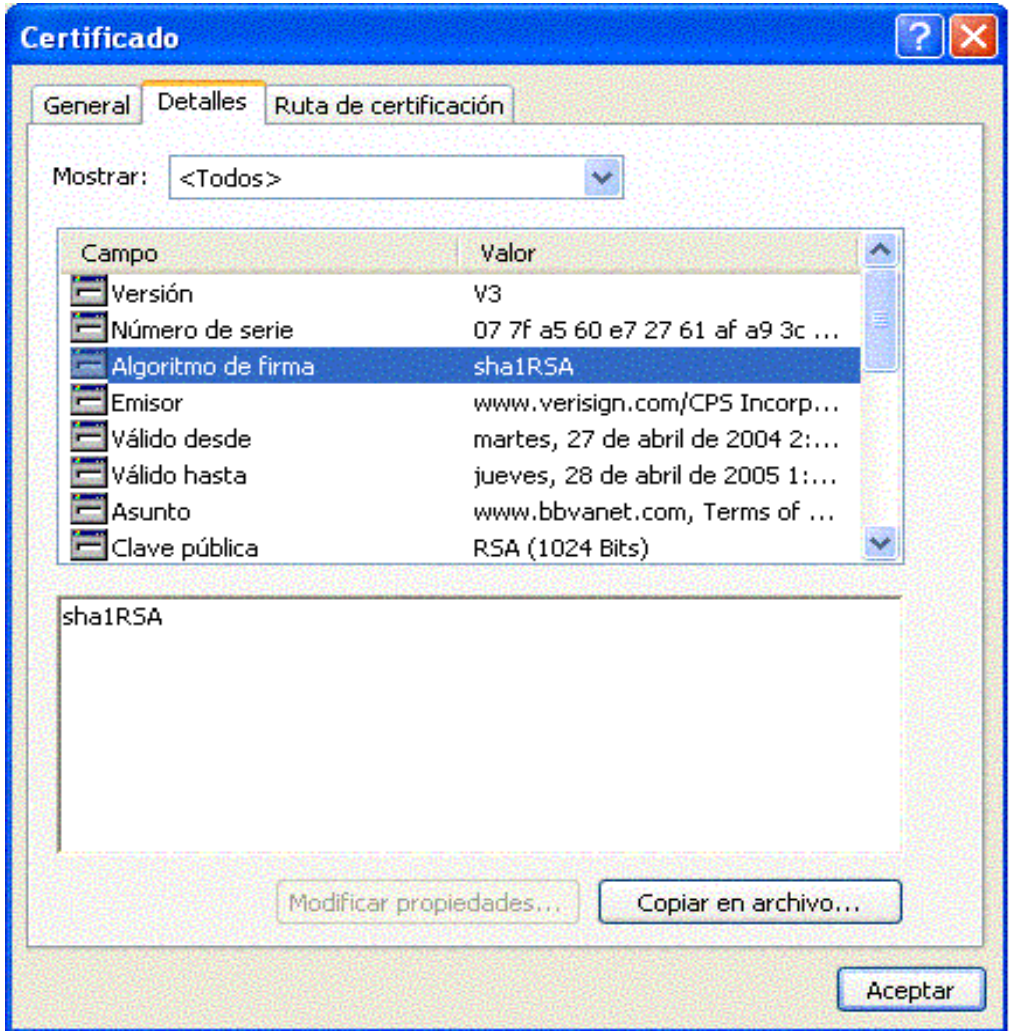
Certificados

CERTIFICADO DE SERVIDOR

- Cada servidor SSL -> un certificado de servidor SSL
- Debe contener:
  - longitud de clave firmada
  - nº serie del certificado
  - algoritmo de firma
  - nombre del servidor
- Ejemplo



Certificados



Certificados

CERTIFICADO PERSONAL

- Diseñado para comprobar la identidad de un individuo emitido por una CA
- Beneficios:
  - Eliminar necesidad de recordar login y password
  - Prueba de pertenecer a una organización
  - Comunicaciones cifradas
  - Restringir acceso a sitios web

Certificados

CERTIFICADO PERSONAL

- A partir de la v.3 de Navigator Netscape e Internet Explorer
  - Creación de claves
  - Obtención de certificados
  - Reto/respuesta
  - Almacenamiento seguro
- En España:
  - Fabrica Nacional de Moneda y Timbre (www.cert.fnmt.es)
  - ANF Autoridad de Certificación (www.anf.es)
  - AC Camerfirma (www.camerfirma.com)
  - Autoridad de Certificación de la Abogacía (www.acabogacia.org)
  - Firma Profesional S.A. (www.firmaprofesional.com)

Certificados

- **Servicios a los que se puede acceder con un certificado de usuario en España**

Administración Central

Agencia Estatal de Administración Tributaria
Comisión del Mercado de las Telecomunicaciones
Instituto de Crédito Oficial
Instituto Nacional de Estadística
Ministerio de Economía
Presidencia de Gobierno
Seguridad Social
Dirección General del Catastro
Dirección General de Costes de Personal y Pensiones Públicas
Ministerio de Trabajo y Asuntos Sociales

Administración Autonómica

Comunidad de Madrid
Gobierno de Canarias
Gobierno de Navarra
Gobierno de la Rioja
Junta de Andalucía
Xunta de Galicia

Administración Local

Ayuntamiento de Alboraya
Ayuntamiento de Laredo
Ayuntamiento de Catarroja
Ayuntamiento de Madrid
Ayuntamiento de Paterna
Ayuntamiento de Totana
Ayuntamiento de Valencia
Diputación de Barcelona

Otros

Asociación de Asesores de Empresa en Internet
Consejo General del Notariado
Gestor de Infraestructuras S.A.
Paradores Nacionales de Turismo
Saniline
SegurosBroker
Sociedad Digital de Autores y Editores

Certificados

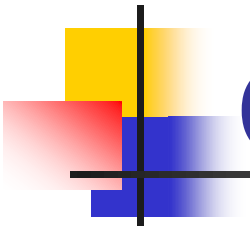
CERTIFICADO PERSONAL

- Cómo solicitarlo de modo gratuito por dos meses:
  1) Ir a https://digitalid.verising.com
  2) Seleccionar PersonalID -> Buy Now -> Enroll Now
  3) Completar el formulario de *enrollment*
     - Nombre y Apellidos
     - Dirección de correo electrónico
  4) Aceptar el acuerdo
  5) Comprobar el correo electrónico, *verisign* enviará un correo con un identificador y la URL de una página web
  6) Ir a la página web indicada e introducir el identificador
  7) El navegador obtendrá el certificado
  8) Para instalarlo seguir las indicaciones del navegador
  9) Comprobación en Internet Explorer: ir a Herramientas -> Opciones de Internet -> Contenido -> Certificados -> Personal

Certificados

CERTIFICADO DE EDITOR DE SOFTWARE

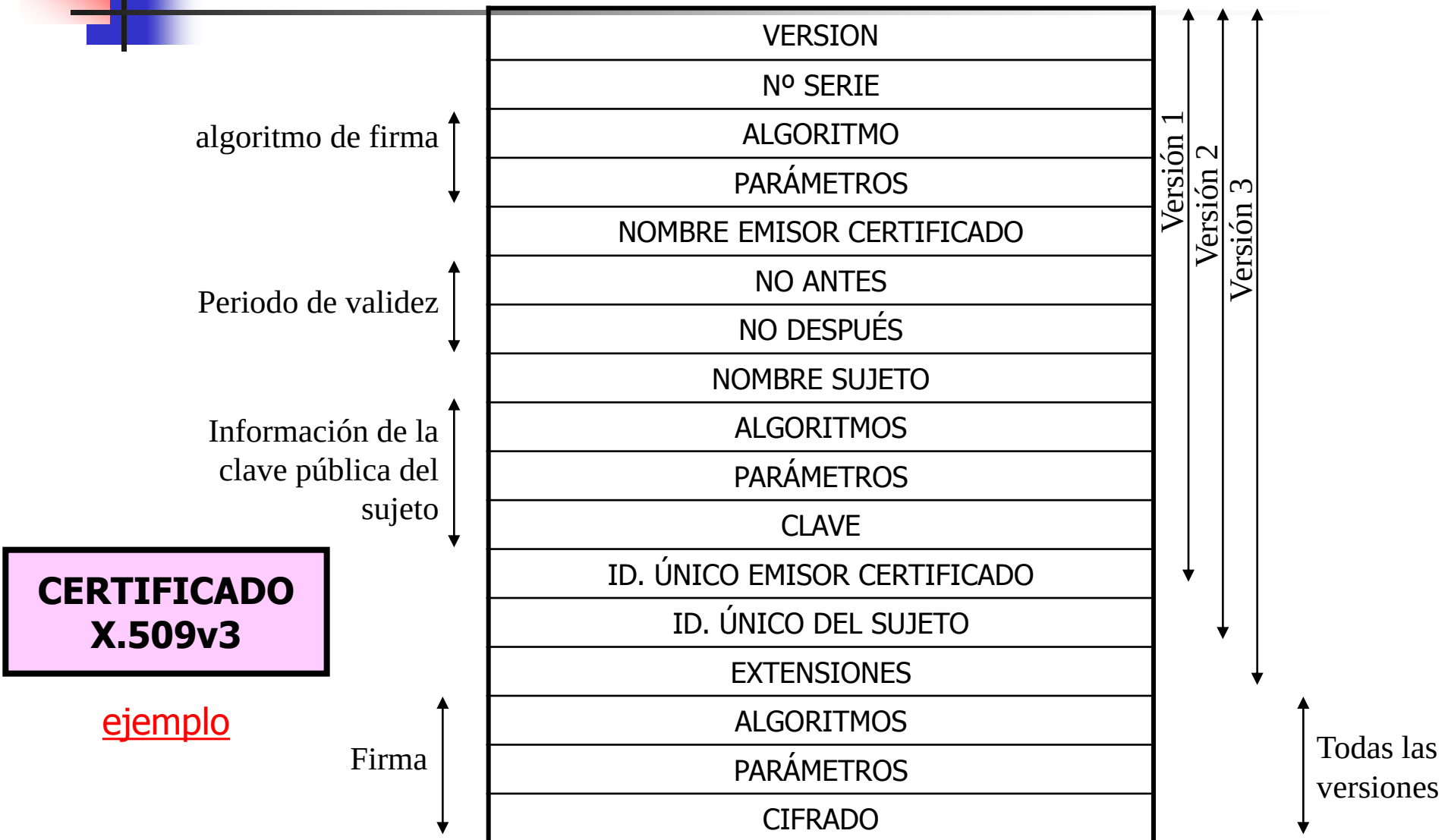
- Firmar programas ejecutables mediante firma electrónica
- Mejora la confiabilidad del software distribuido por Internet
- Propuestas:
  - Authenticode (Microsoft)
  - JAR, formato de archivo java que permite uso de firma digital



Certificados

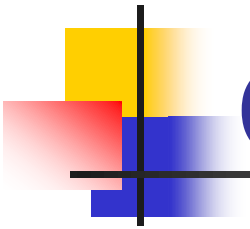
- **Certificado X.509**
  - Parte de la serie de recomendaciones X.500
  - X.509 permite servicio de autenticación
  - Estructura de certificado X.509 empleada en muchos contextos (S/MIME, seguridad IP, SSL/TLS, SET, …)
  - Versión 3 revisada en 2000

Certificados



CERTIFICADO X.509v3

ejemplo



Certificados

- Las claves privadas no son personas
- Los nombres distinguidos no son personas
- Existen demasiados nombres de personas iguales
- Los certificados digitales no dicen lo suficiente
- X.509 v.3 no permite la divulgación selectiva
- Los certificados digitales permiten la combinación fácil de datos
- ¿Cuántas CA necesita la sociedad?
- ¿Cómo prestar una clave?
- ¿Existen mejores opciones a las firmas digitales de clave pública?