

## *2. Conceptos básicos en seguridad informática*



# Estado de protección (I)



- 
- Un *Sistema de protección* describe condiciones bajo las que un sistema es seguro -> *máquina de estados*
  - Siendo P un conjunto de todos los posibles estados seguros del sistema, y Q el conjunto que define los estados de protección en el cual el sistema está autorizado a residir, tal que  $Q \subseteq P$ .
    - *Política de seguridad* : caracterización de los estados del conjunto Q.
  - *Transición de estados*: operaciones permitidas *sobre elementos* de Q.
    - *Mecanismo de seguridad* : prevención para que el sistema no entre en ningún estado que no pertenezca a Q.



# Estado de protección (II)

---

- Supongamos que el conjunto de mecanismos de seguridad restringen al sistema a un conjunto de estados  $M$ , entonces dicho conjunto de mecanismos se dice que es :
  - *Seguro*, si  $M \subseteq Q$ .
  - *Preciso*, Si  $M = Q$ .
  - *Amplio*, si existen estados  $m \in M$  que  $m \notin Q$ .
- El *objetivo* es buscar la *precisión* en los resultados de la aplicación de mecanismos de seguridad.



# Matriz de control de accesos

---

- Modelo simple y preciso para describir, especificar los estados de protección de un sistema (SO, BD).
- Definimos :
  - Conjunto  $O$  de objetos protegidos (ficheros, máquinas,..)
  - Conjunto  $S$  de objetos activos (sujetos). Inician transiciones (procesos, usuarios,....)
  - Conjunto  $P$  de permisos para operaciones sobre elementos de  $O$ .
  - Matriz  $A$ , donde cada entrada  $a[s,o] \subseteq P$ , donde  $s \in S$ , y  $o \in O$ .
  - Finalmente, la tupla  $(S,O,A)$  define el conjunto de estados protegidos del sistema.



# Ejemplo matriz de control de accesos

---

- Acceso de procesos a ficheros o a otros procesos.
  - $O = \{\text{proceso1}, \text{proceso2}, \text{fichero1}, \text{fichero2}, \text{fichero3}\}$
  - $S = \{\text{proceso1}, \text{proceso2}\}$
  - $P = \{\text{read}(r), \text{write}(w), \text{execute}(x), \text{append}(a), \text{own}(o)\}$
  - $A :$

	<u>fichero1</u>	<u>fichero2</u>	<u>fichero3</u>	<u>proceso1</u>
<u>proceso1</u>	r,w,o	r	r,w,x,o	w
<u>proceso2</u>	a	r,o	r	r,w,x,o



# Transición de estados de protección

---

- Teniendo un estado de protección de sistema  $X_i=(S_i,O_i,A_i)$  y queremos pasar a una estado  $X_{i+1}$ , se aplicará una transición  $t_{i+1}$  para pasar del estado  $X_i$  al  $X_{i+1}$ .
- Se puede representar una transición de estado por un comando o procedimiento de transformación al cual se le instancian determinados parámetros.
  - CreaciónSujeto (s)                      SuprimirSujeto(s)
  - CreaciónObjeto(o)                      SuprimirObjeto(o)
  - AñadirPermiso(p,a[s,o])              QuitarPermiso(p,a[s,o])



# Atenuación de privilegios: copia y propiedad

---



- *Principio de atenuación de privilegios* : Un sujeto no puede asignar a otro sujeto, permisos que no posee.
- *Permiso de copia* : da derecho a asignar permisos para otros (permiso “P” en Windows NT)
- *Permiso de propiedad*: da derecho a añadir o suprimir privilegios para si mismo.



# Decidibilidad en seguridad

---

- ¿ Existe un algoritmo genérico que pueda determinar si un estado es seguro o no ?
- En el caso abstracto no es decidable.
- Se buscan modelos más restringidos que permitan determinar mediante un algoritmo si un estado es seguro o no --> aplicación en políticas y composición de políticas de seguridad :
  - Modelo de protección de Adquirir-Asignar (Take-Grant).



# Planificación de la seguridad

---

- Planificar las necesidades en seguridad (nivel de confianza).
- Evaluación de riesgos (identificación de los problemas).
- Análisis de costes/beneficios y mejores prácticas.
- Creación de políticas adaptadas a las necesidades.
- Implementación.



# Planificar las necesidades



---

## ■ Tipos :

- Confidencialidad (Privacidad) : ley de protección de datos.
- Integridad de datos.
- Disponibilidad.
- Consistencia.
- Control.
- Auditoría.

## ■ Ejemplos :

- Banco: integridad, control y capacidades de auditoría
- Defensa : confidencialidad
- Universidad : integridad y disponibilidad



# Evaluación de riesgos

---

- Qué proteger. Frente a qué proteger. Tiempo, esfuerzo y dinero que se está dispuesto a invertir para una adecuada protección.
  
- 3 pasos :
  - Identificación de bienes y su valor
  - Identificación de peligros
  - Cálculo de riesgos



# Evaluación de riesgos (II)



- 
- Problemas físicos :
    - Fenómenos naturales : incendios, inundaciones, terremotos,...
    - Acceso a componentes físicos del sistema : ordenador, consola de administración, cables de red, etc.
  - Problemas lógicos :
    - Errores en los programas.
    - Utilización incorrecta por parte de los usuarios (educación).
    - Utilización fraudulenta del sistema.
    - Principalmente, tres categorías :
      - Contraseñas.
      - Sistemas de ficheros.
      - La red.
  - Herramientas para la detección de riesgos informáticos.



# Análisis de costes/beneficios y mejores prácticas

---



- Coste de la pérdida.
- Probabilidad de pérdida.
- Coste de la prevención. Comparativas. Prioridades.
- Seguro =  $1 / \text{Utilizable}$ .
- Mejores prácticas : recomendaciones, procedimientos y políticas generalmente aceptadas por los expertos en seguridad informática



# Políticas de seguridad

---



- Roles :
  - Identificación y justificación de los elementos a proteger.
  - Definir responsabilidades para dicha protección.
  - Base de interpretación y resolución de conflictos.
- Procedimientos estándares.
- Guías.



# Políticas de seguridad (II)



- Ideas en el desarrollo de políticas :
  - Asignar propiedad a información y equipos.
  - Enfocar la expresión de políticas de forma positiva.
  - Recordar que empleados y usuarios son personas
  - Educar a los usuarios.
  - Responsabilidad debe conllevar autoridad.
  - Conocer el perímetro de seguridad (portátiles, PDAs, redes inalámbricas, ordenadores utilizados en casa, DVDs, discos extraíbles, visitas, impresoras, copiadoras, fax,..)
  - Decidir filosofía básica (permitido lo no especificado o inversa)
  - Niveles independientes y redundantes de defensa a varios niveles, con auditoría y monitorización



# Implementación

---



- Gestión de riesgos supone sentido común.
  - Uso de tecnologías y educación de personas
  - Múltiples niveles de defensa
  - Priorizar
- Auditorías de cumplimiento de políticas. Problemas :
  - Personal insuficiente (falta formación, sobrecarga,..)
  - Material insuficiente (inadecuación de recursos, sobrecarga,..)
  - Organización insuficiente (asignación responsabilidades y autoridades, conflictos de responsabilidades, tareas poco claras)
  - Política insuficiente (riesgos no previstos, incompletud, conflictos de políticas, discordancia entre políticas y contexto)



# Implementación (II)

---

- Respuestas a incidentes
- Definir que elementos se subcontratan y cómo.
- No se recomienda seguridad mediante oscuridad.
- Tecnología : sistemas y criptografía.



# Principios básicos prevención

---



- Modelos de Protección
- Autenticación (quién es quién) : identificación digital
- Autorización (a quién se le deja hacer qué) :
  - Permisos y privilegios
  - Control de accesos



# Elementos básicos de implementación técnica

---



- Representación de identidades :
  - Máquinas (IP, DNS,...)
  - Ficheros, procesos, objetos (identificadores numéricos de sistema)
  - Usuarios (cuentas de usuario)
  - Grupos y roles (accesos basados en roles)
  - Certificados (confianza aportada por terceros)
  - Estado y cookies (identidad en conexión web)
  - Anonimidad (en conexiones web)



# Elementos básicos de implementación técnica (II)

---



- Mecanismos de control de acceso
  - Listas de control de acceso (ACLs sistemas de ficheros)
  - Capabilities (llaves de permiso de acceso sin identificación directa)
  - Cerraduras y llaves
- Criptografía
- Controles del flujo de información :
  - Mecanismos basados en compiladores
  - Mecanismos basados en ejecución
- Confinamiento :
  - Máquinas virtuales
  - Cubos de arena (sandboxes)



# Ejemplo: Conceptos de seguridad en Java



- 
- Evolución :
    - 1.0 : modelo de *cubo de arena* (sandbox) :
      - Acceso total a todo el código local
      - Acceso completamente restringido a código Java applets
    - 1.1 : Introducción de applets firmados :
      - Acceso total a código local y applets firmados
      - Acceso completamente restringido a applets no firmados.
    - Java 2 : servicio de seguridad en ejecución genérico para código local y applets. Concepto básico de *Dominios de Protección*.



# Seguridad Java 2 (I)



- Seguridad del núcleo (java runtime) :
  - *Verificador de Byte-Code*
  - *Cargador de clases*
  - *Origen del código* (URL, certificados digitales,..)
  - *Clases de permisos* : definen acceso a recursos de sistema (ficheros, sockets, .....
  - *Dominios de protección* : relaciones de grupos de clases con dominios de protección donde se definen permisos..
  - *Políticas* : conjunto de permisos y dominios.
  - *Gestor de seguridad* :verifica autorizaciones (permisos)
  - *Controlador de accesos* : control a aspectos de más bajo nivel.



# Seguridad Java 2 (II)

---



- Almacén de claves : contiene claves privadas y certificados, cifrados.
- Extensiones de seguridad :
  - Servicio de autenticación y autorización de Java (JAAS)
    - Definición de permisos de ejecución de código Java específico
  - Extensión de sockets seguros de Java (JSSE)
    - Utilización de SSL/TSL
  - Extensión de servicios genéricos de seguridad de Java (GSS-API )



# Políticas

---

- Un *modelo de seguridad* representa un conjunto de políticas de seguridad.
- Una *política de seguridad* define los estados de sistema autorizados, o seguros, en contraposición a aquellos que no lo son.
- Un *sistema es seguro* si no entra en estados no autorizados.
- Un *agujero de seguridad* ocurre cuando un sistema entra en un estado no autorizado.
- La información I tiene *propiedad de confidencialidad* con respecto al conjunto de entidades X, si ningún miembro de X puede obtener ninguna información de I.
- La información I tiene *propiedad de integridad* con respecto a X, si todos los miembros de X confían en I.
- El recurso R tiene *propiedad de disponibilidad* con respecto a X, si todos los miembros de X tienen acceso a R.
- Un *mecanismo de seguridad* es una entidad o procedimiento que obliga a cumplir alguna parte de la política de seguridad.
- Papel central de la noción de *confianza* y de las *hipótesis* de seguridad.



# Tipos de control de acceso

---

- Dos tipos utilizados aisladamente o en combinación :
  - *Control de accesos discrecional* (DAC), o control de accesos basado en identidad : cada usuario individual puede utilizar un mecanismo de control de accesos para permitir o denegar accesos a recursos.
  - *Control de accesos obligatorio* (MAC), o control de accesos basado en reglas : un mecanismo de sistema controla los accesos y los usuarios individuales no pueden modificar ese acceso.
- Existe un tercero, *control de accesos controlado por el creador*, donde el creador del recurso define los controles de acceso.



# Lenguajes de políticas de seguridad

---

- Lenguajes que representan políticas de seguridad :
  - Lenguajes de alto nivel : restricciones de políticas mediante abstracciones.
  - Lenguajes de bajo nivel : restricciones expresadas mediante entradas o invocaciones a programas existentes en el sistema.



# Lenguaje de alto nivel

---

- Ejemplo :
  - A cada objeto corresponde a un *tipo*.
  - A cada sujeto corresponde un *dominio*.
  - Las construcciones del lenguaje definen restricciones de miembros de dominio con objetos de un tipo.
    - Política que solo permite escribir en binarios de sistema a administradores en Unix :
      - Domain d\_user = (/usr/bin/sh, /usr/bin/ksh)  
(crwxd -> t\_generic)  
(rxd -> t\_sysbin)  
(crwd -> t\_writeable)  
(rd -> t\_readable)  
(exec -> d\_admin) ????



# Lenguaje bajo nivel

---

- Ejemplos :

xhost +hendrix -merlin

/usr/local/tripwire +gimnpsu012345678-a

/etc/pac 0755 1 root root 16384 sept 17 22:08



# Componentes de sistema

---

- Representacion de identidad
- Mecanismos de control de accesos
- Flujo de informacion
- El problema de confinamiento.



# Elementos adicionales

---

- Código Malicioso
- Análisis de vulnerabilidades
- Auditoria
- Detección de intrusiones



# Seguridad práctica en sistemas :servidor web

---



- Política
  - Red : Cortafuegos, DMZs, etc
  - Usuarios
  - Autenticación
  - Procesos
  - Ficheros



# Sistema de autenticación Kerberos



- Sistema de autenticación de única firma en red con posibilidad de privacidad. Versión 5: MIT, Heimdal, Windows 200X
- Abstracción de acceso de programas a servicios Kerberos :
  - Estándar IETF: Generic Security Services API (GSS-API). disponible en C, C++, JAVA.
  - Windows : Security Support Provider Interface (SSPI)
  - Negociación de mecanismos de autenticación : Simple and Protected Negotiation Mechanism (SPNEGO)
- Tecnología de clave simétrica (3DES en V. 5))

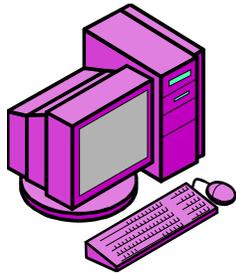
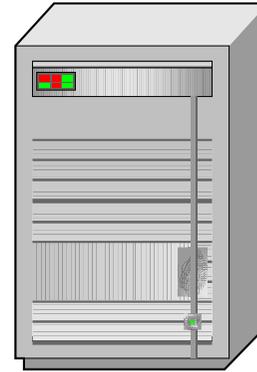


# Elementos de Kerberos

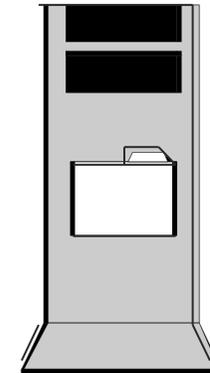
---



Servidor Kerberos  
Centro de distribución de claves  
(KDC)



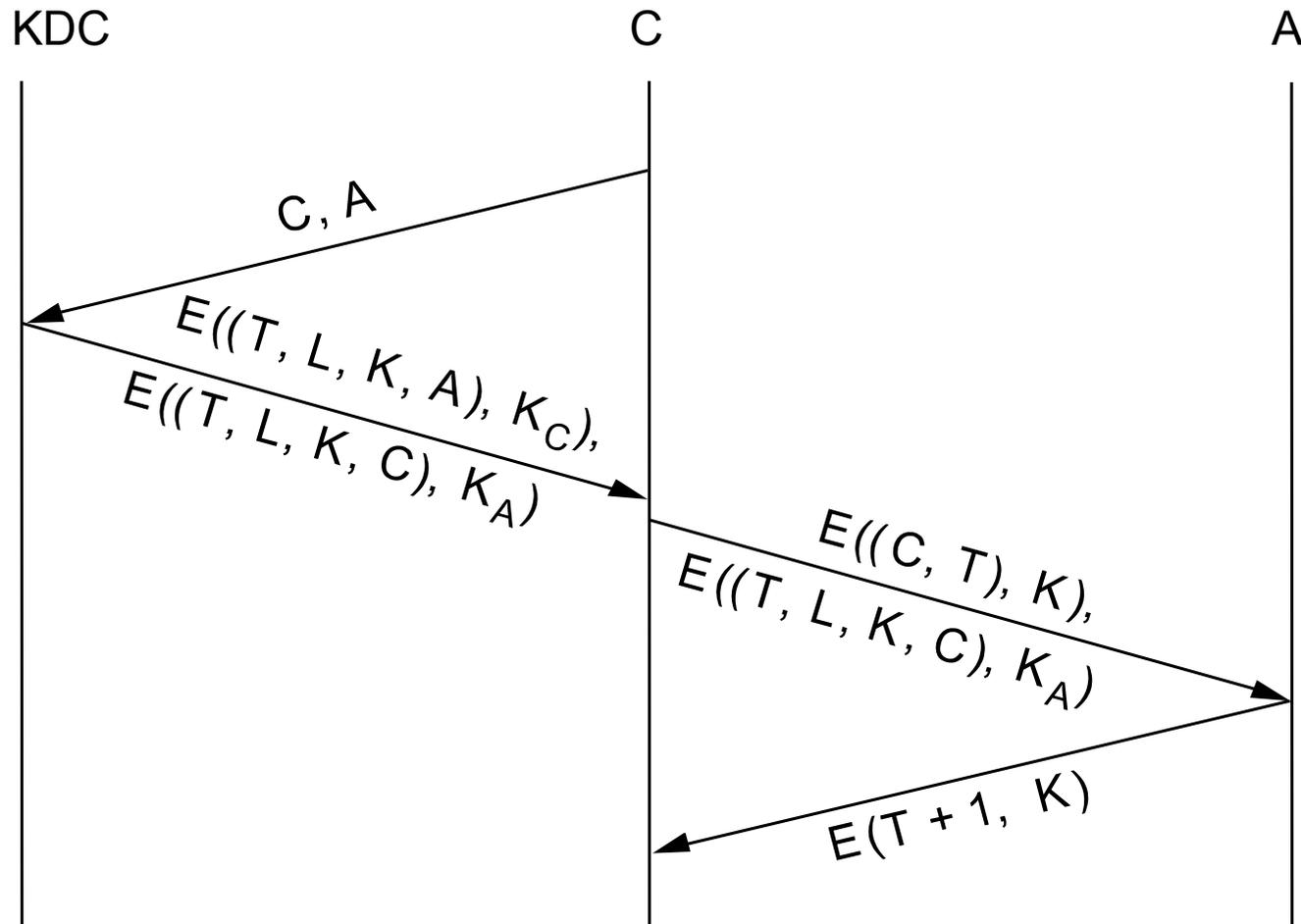
Cliente (C)



Aplicación de red (A)

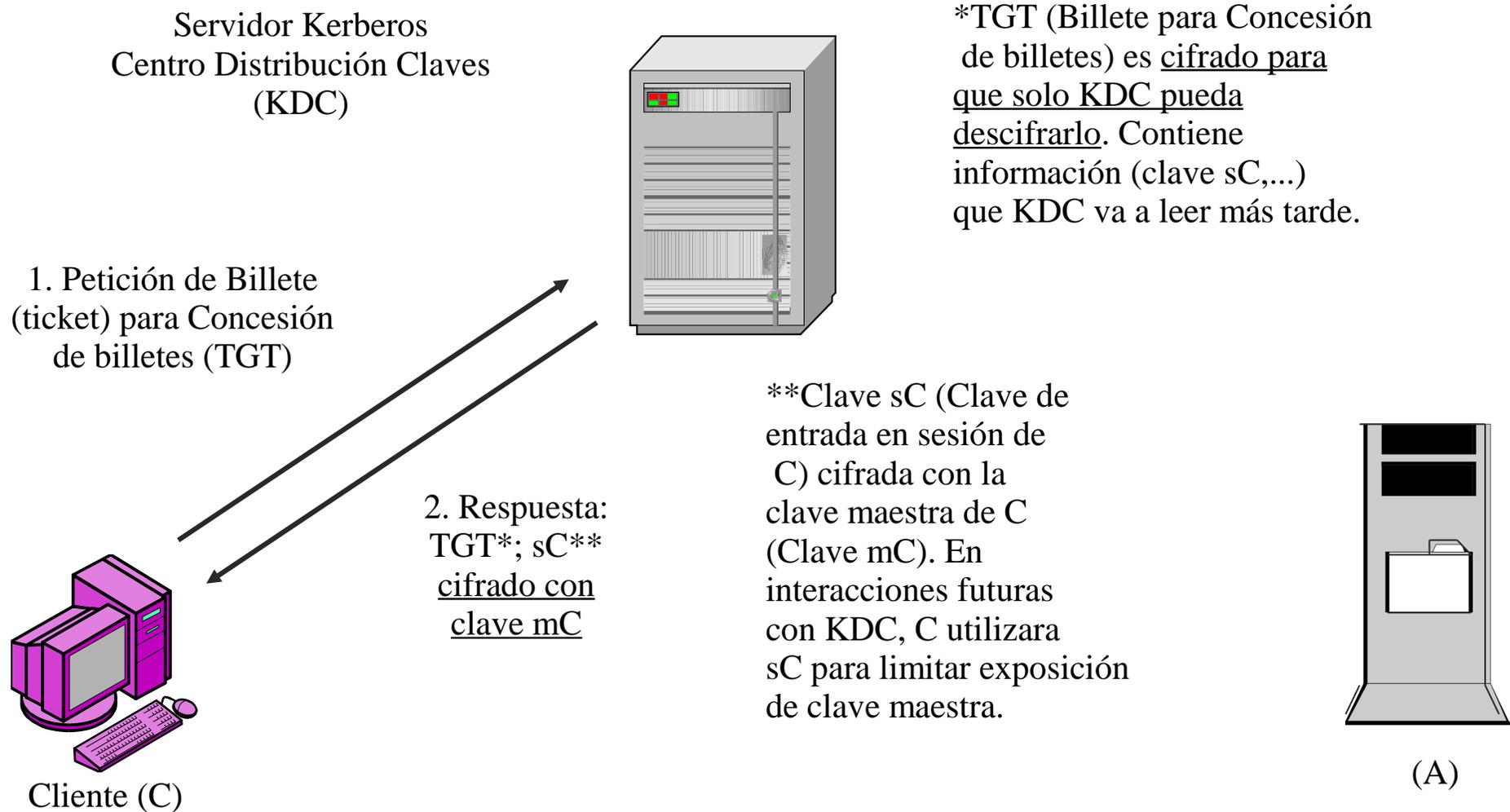


# Concepto protocolo autenticación Kerberos



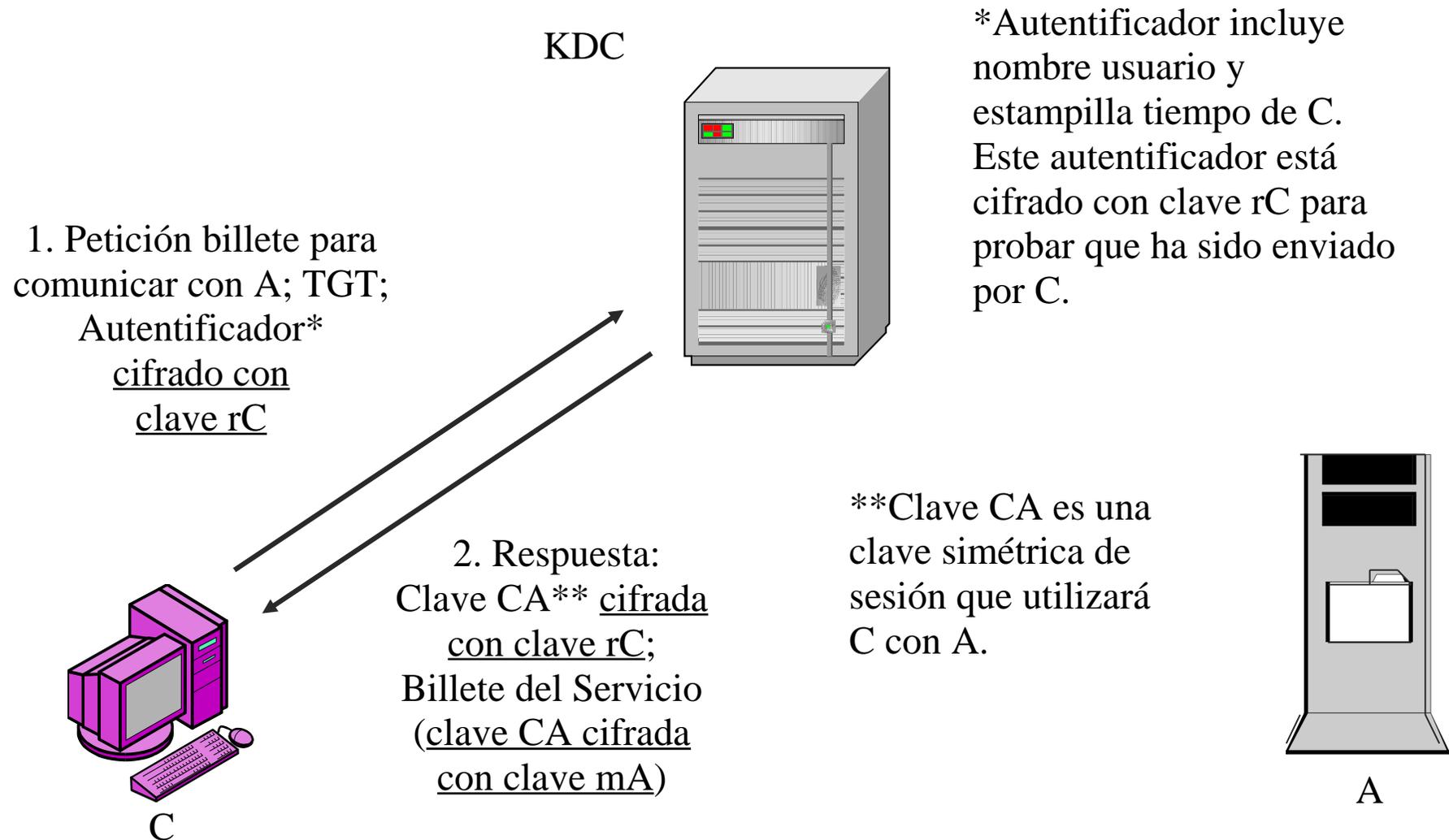


# Implementación protocolo Kerberos (I)



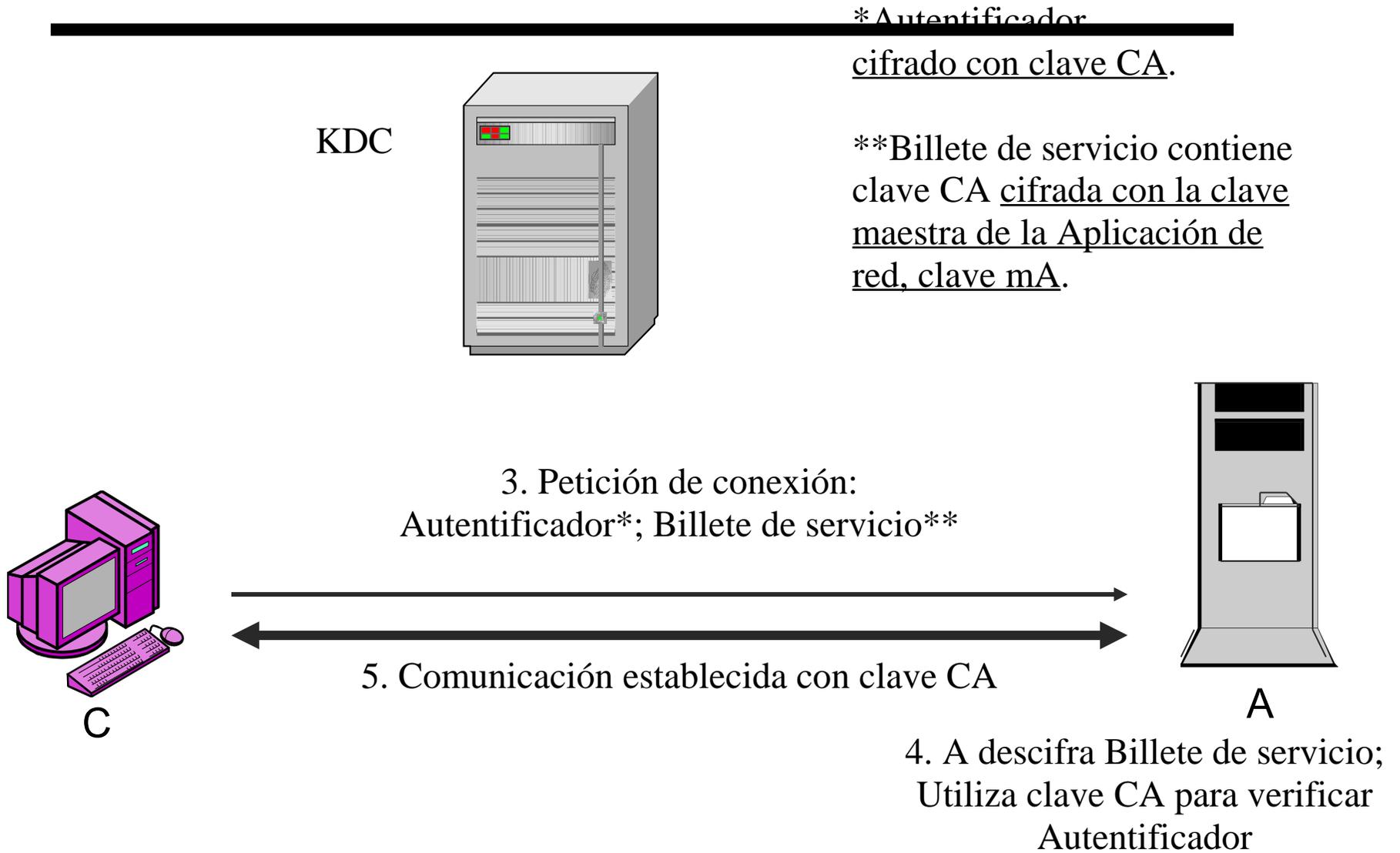


# Implementación protocolo Kerberos (II)





# Implementación protocolo Kerberos (III)





# SSH (1)



- 
- SSH (Secure Shell) es un protocolo de entrada en sesión y/o ejecución de comandos en una máquina remota (V. 2.0).  
Implementación standard: Openssh
  - Es un sustituto completo de rsh, rlogin, rcp, telnet, rexec, rcp y ftp, pero que provee, además, comunicaciones cifradas seguras entre 2 máquinas sobre una red insegura.
  - Autenticación usuarios y máquinas (clave pública: RSA, DSA).  
Privacidad (simétrico: 3DES, Blowfish). Integridad/Autenticación mensajes (HMAC-SHA1, HMAC-MD5). Compresión de datos.



# SSH (2)

---



- Conexión : `ssh pepe@hendrix.cps.unizar.es`
- Claves de autenticación automática (RSA, DSA) :
  - Creación : `ssh-keygen`
  - Almacenamiento autenticación usuario:
    - Clave privada (identificación): `$HOME/.ssh/{identity, id_rsa,...}`
    - Clave pública (autenticación): `$HOME/.ssh/{identity,...}.pub`
    - Claves externas: `$HOME/.ssh/{authorized_keys}`
  - Almacenamiento autenticación máquina :
    - Clave privada (identificación): `/etc/ssh/ssh_host_{key, rsa_key,...}`
    - Clave pública (autorización): `$HOME/.ssh/known_hosts`



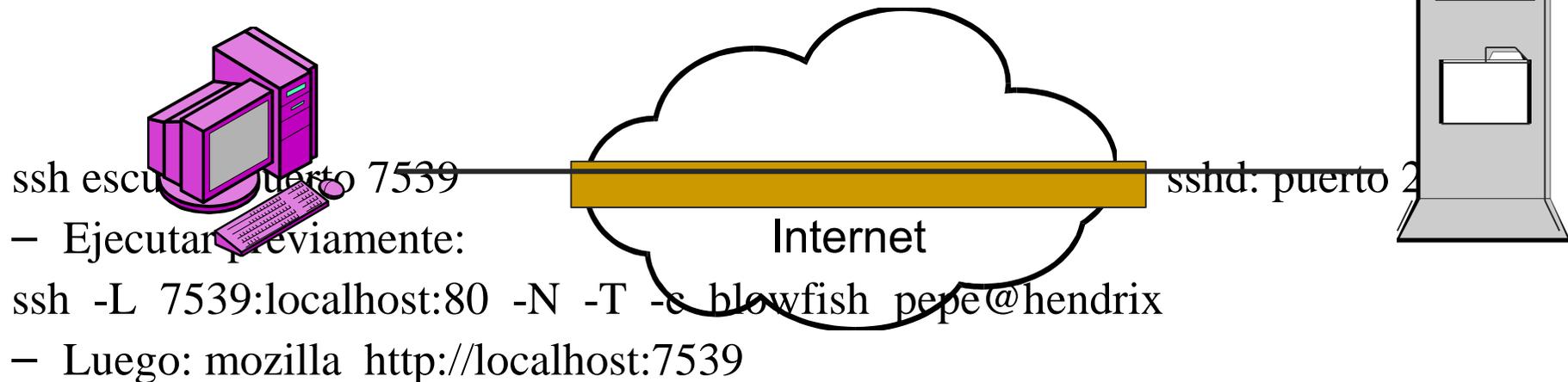
# SSH (2)



- Tuneles SSH (no es necesario programación específica):

Apache: puerto 80

Mozilla





# Certificados digitales

---



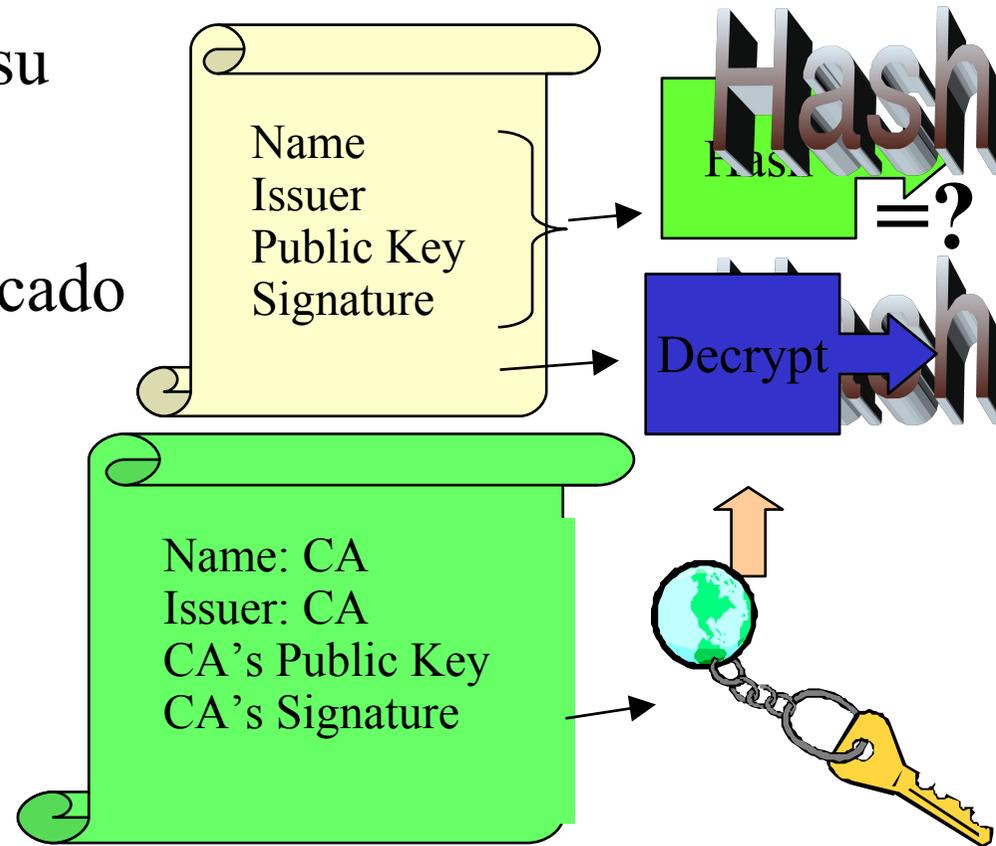
- Utilización de criptografía pública como un tipo de sistema de identificación de propósito general.
- Un certificado es una credencial que relaciona un nombre con una clave pública (entre otros datos) en un paquete firmado por una tercera parte confiable, con un tiempo de validez.
- Como un pasaporte o un carnet de conducir.
- Estándar: certificados *X.509 v3*.
- Comprobando la firma, uno puede verificar que una clave pública pertenece a un determinado usuario.



# Autoridades de certificación (CAs)



- Un pequeño conjunto de entidades confiables (tercera parte confiable) que establecen certificados firmados.
- La autoridad de certificación firma su propio certificado que puede ser distribuido de forma confiable.
- Entonces la clave pública del certificado de la CA puede ser utilizado para verificar otros certificados

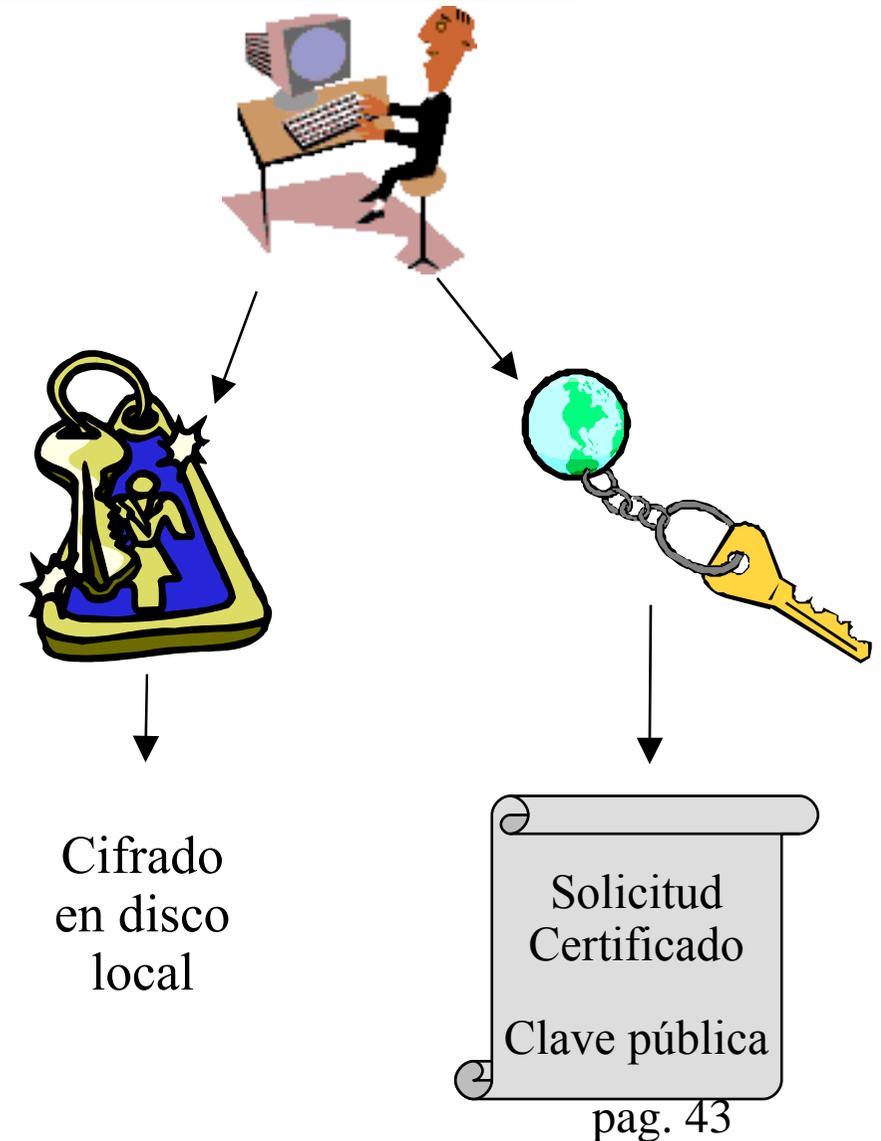




# Solicitud de creación y producción de certificados



- Usuario crea un pareja clave pública/privada.
- Clave privada es almacenada cifrada (passphrase) con una contraseña del usuario.
- La clave pública se coloca en una petición de creación de certificado, que es enviada a una autoridad de certificación.



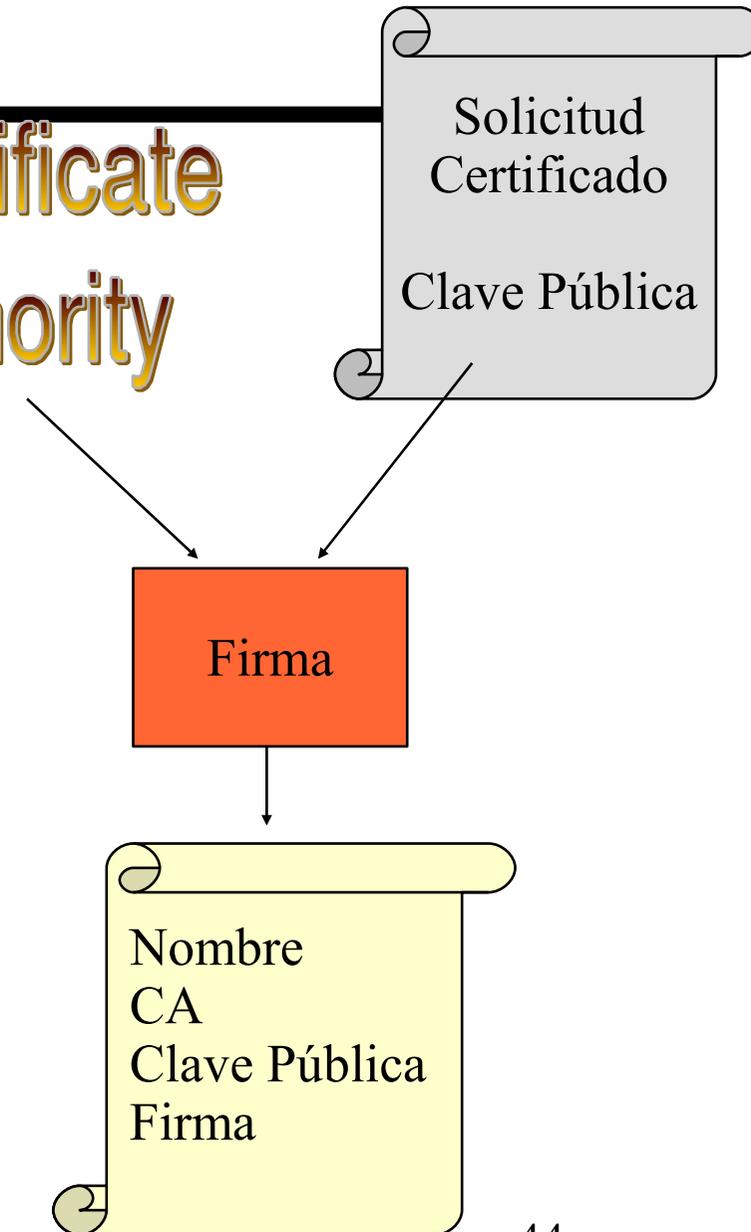


# Producción de certificados



- La CA normalmente incluye una Autoridad de Registro (RA) que verifica la petición
  - El nombre del usuario debe ser único en el contexto del CA
  - Es el nombre real del usuario, etc
- El CA firma, entonces, la petición y produce un certificado para el usuario.

Certificate  
Authority





# Revocación de certificados



- CAs necesitan revocar certificados si :
  - La clave privada del usuario ha sido comprometida.
  - CA descubre haber entregado certificado a usuario erróneo.
  - Certificado producido para permitir acceso usuario a un servicio, usuario ha perdido autorización de acceso a él.
  - Sistema del CA comprometido de tal forma que otro puede emitir certificados falsos de esa CA.
- Métodos de gestionar revocaciones :
  - Lista de revocación de certificados (CRL).
    - Búsqueda regular de CRLs en los CAs: Campo Punto Distribución CRL en X.509 v3 (CDP).
  - Validación de certificados en tiempo real.



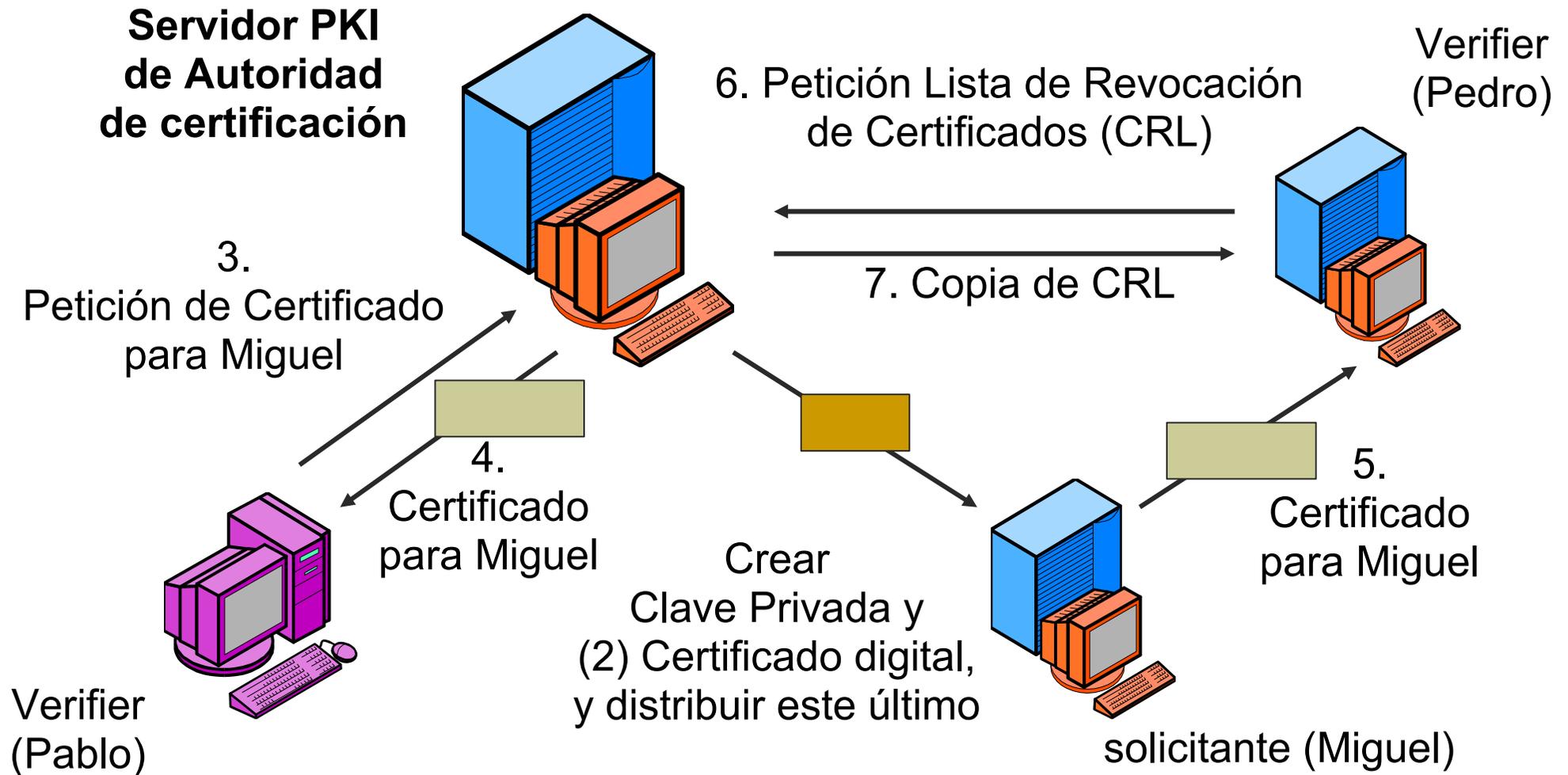
# Infraestructura de clave pública (PKI) (I)



- 
- *Infraestructura de clave pública* es el sistema de certificados digitales (X.509 v3), autoridades de certificación, sistemas y hardware utilizado para distribuir claves públicas.
  - Espacio de nombres de usuarios.
  - Certificados en navegadores (Mozilla, Explorer,...)
  - Necesidad de más información en los certificados.
  - ¿ Cuantas autoridades de certificación ?



# Infraestructura de clave pública (PKI) (II)

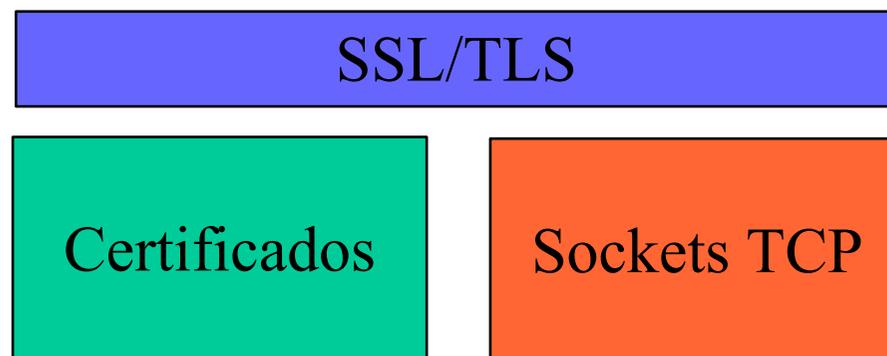




# SSL/TLS



- El Nivel Seguro de Sockets (SSL) o Seguridad de Nivel Transporte (TLS -SSL 3.1-), Estándar IETF, utiliza certificados (opcional, pero usual) y sockets TCP para proveer conexiones seguras, con las siguientes opciones :
  - Autenticación de una o ambas partes utilizando certificados
  - Protección de mensajes :
    - Confidencialidad (criptografía simétrica)
    - Integridad (MD5, SHA)
- Librerías Java, C, C++
- Librería OpenSSL.





# Arquitectura SSL/TLS

---



- Dos niveles :
  - Protocolo registro SSL provee servicios básicos de seguridad.
  - 3 protocolos de nivel superior:
    - Apretón de manos, cambio de especificación de cifrado, alertas
- Conexión
  - Transporte con algún servicio, asociado a una sesión
- Sesión
  - Creada por apretón de manos, define parámetros criptográficos de seguridad para múltiples conexiones.



# Sesión y conexión

---



- Parámetros de sesión
  - ID, certificado del otro, método de compresión, especificación del cifrado, clave secreta maestra, se puede reanudar.
- Parámetros de conexión:
  - Calor aleatorio de cliente y servidor, servidor escribe clave secreta MAC, cliente escribe clave secreta MAC, cliente escribe clave, IV, número de secuencia.



# Protocolo de registro de SSL

---



- 2 servicios
  - Confidencialidad e integridad de mensajes
- Protocolo por niveles:
  - Fragmentar datos de aplicación en bloques
  - Comprimir datos
  - Aplicar código de autenticación de mensaje (MAC) =  $h(m|s)$  para mensaje  $m$  y clave secreta  $s$
  - Cifrar con la clave del cliente o del servidor
  - Transmitir sobre TCP
- Especificar tipo de contenido para protocolos superiores



# IP Security (IPSEC)

---



- Objetivos :
  - IP4 no diseñado para seguridad
  - Mecanismo seguridad en nivel de red para IP4 e IP6
  - Puede ser transparente para usuarios



# Arquitectura y conceptos IPSEC

---

- Implementación en estaciones y encaminadores
- Modo túnel vs. Modo transporte
- Asociación de seguridad (SA)
  - Índice de parámetros de seguridad (SPI)
  - Base de datos de la política de seguridad (SPD)
  - Base de datos de la asociación de seguridad (SAD)
- Protocolo Seguridad Encapsulada (ESP), encapsulación de la carga de seguridad
- Cabecera Autenticada (AH)



# Estaciones y Encaminadores

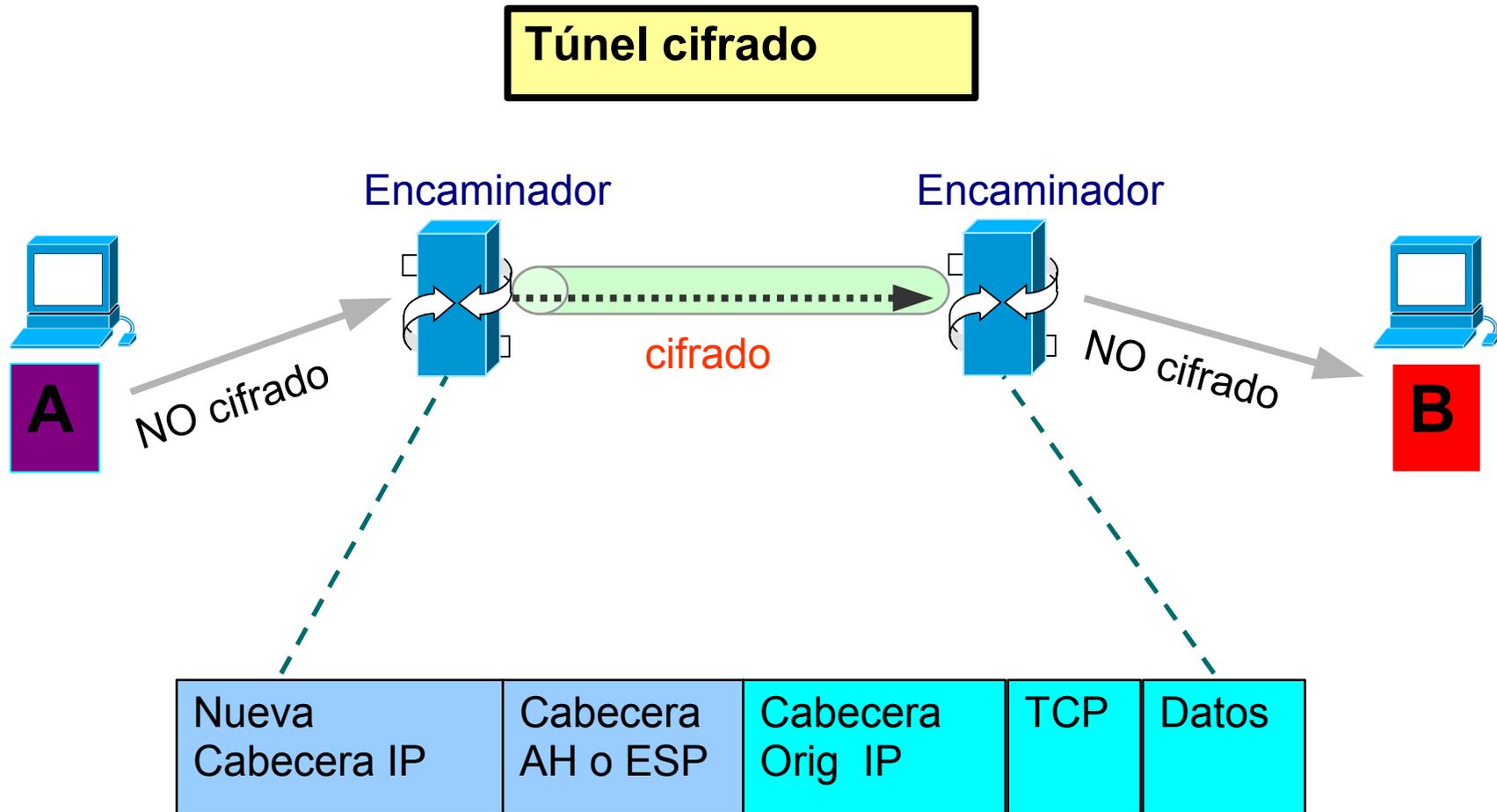
---



- Estaciones pueden implementar IPSec a :
  - Otras estaciones en modo transporte o túnel
  - Encaminadores en modo túnel
- Encaminadores a encaminadores : modo túnel:

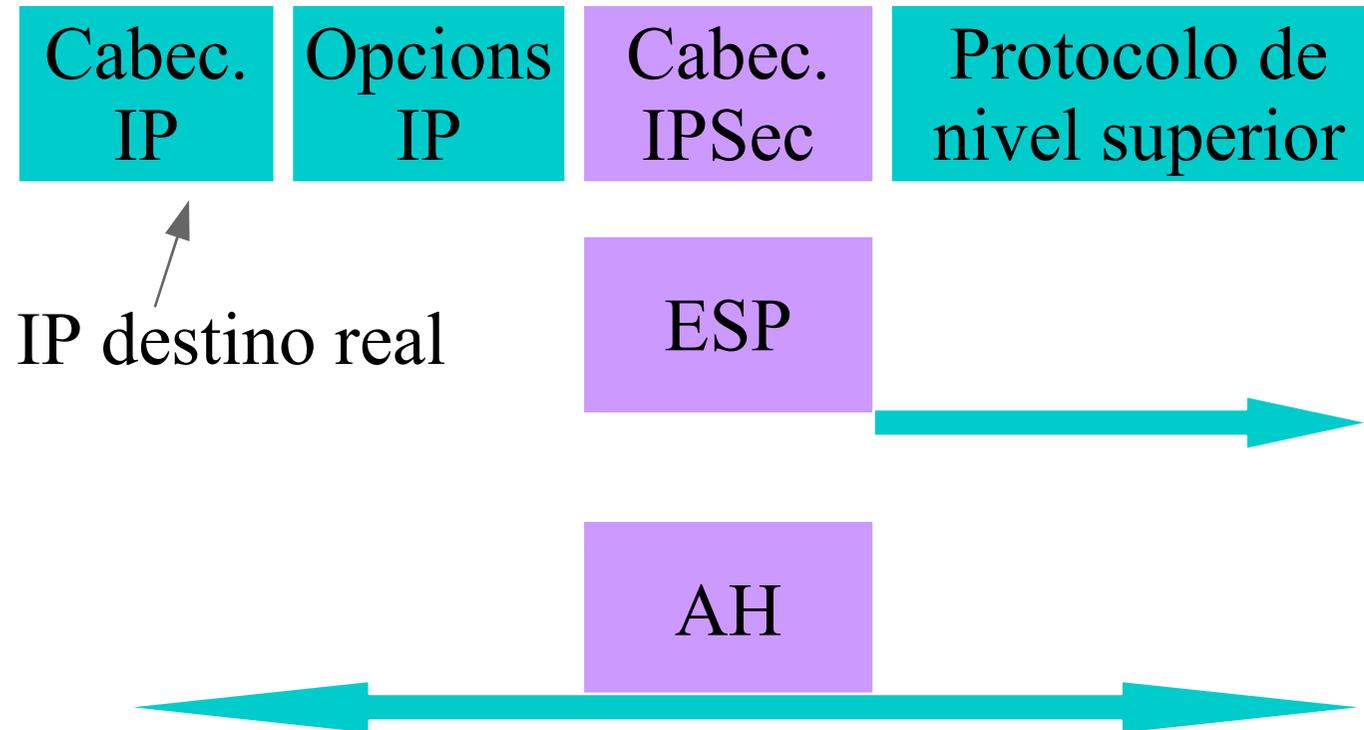


# Modo túnel





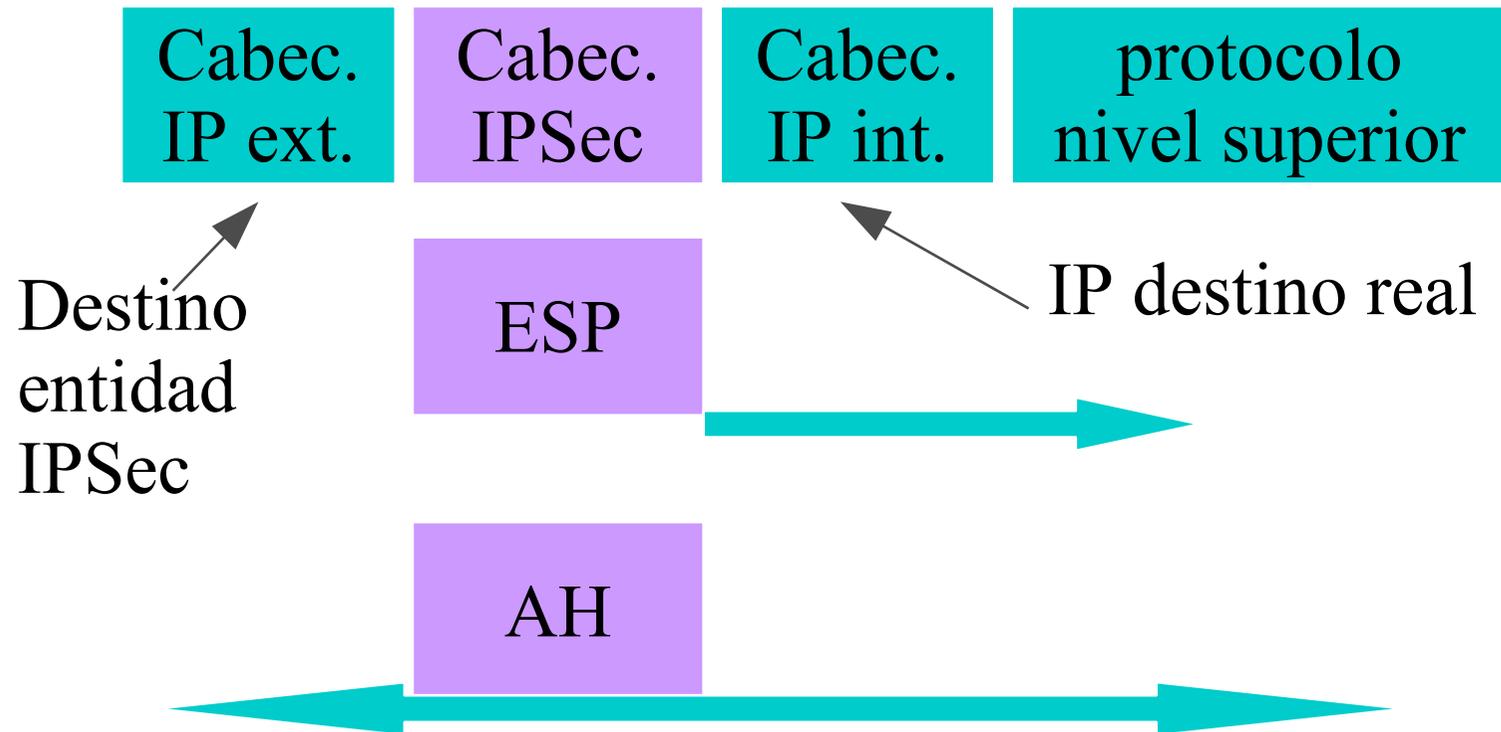
# Modo transporte



- ESP protege sólo la carga de niveles superiores
- AH puede proteger tanto cabeceras IP como la carga de niveles superiores



# Modo túnel



- ESP se aplica sólo al paquete interior
- AH puede ser aplicado al porciones de la cabecera exterior



# Asociación de seguridad (SA)

---



- Determina procesamiento IPSec para *emisores*
- Determina procesamiento IPSec para *receptores*
- *SAs no son fijos.....* Son generados y personalizados para cada flujo de tráfico.
- Índice de parámetros de seguridad (SPI):
  - Valor de hasta 32 bits
  - SPI enviado con el paquete por el emisor
  - SPI permite al receptor seleccionar SA correcto->determina correcto procesamiento de seguridad (dado previo acuerdo con emisor)
  - SPI + dirección IP destino + Protocolo IPSec (AH o ESP) identifica de forma única al SA



# Base de datos de SAs (SAD)

---

- Mantiene parámetros para cada SA
  - Tiempo de vida del SA
  - Información de AH y ESP
  - Modo túnel o transporte
- Cada estación o encaminador que participa en IPSec tiene su propia base de datos de SAs.
- Puede ser aplicada más de 1 SA a un paquete IP.
- Ejemplo : ESP no autentifica nueva cabecera IP. ¿ Cómo autentificarla ?
  - Utilizar una SA para aplicar ESP con/sin autenticación sobre paquete original
  - Utilizar 2ª SA para aplicar AH.



# Base de datos de Políticas de seguridad (SPD)

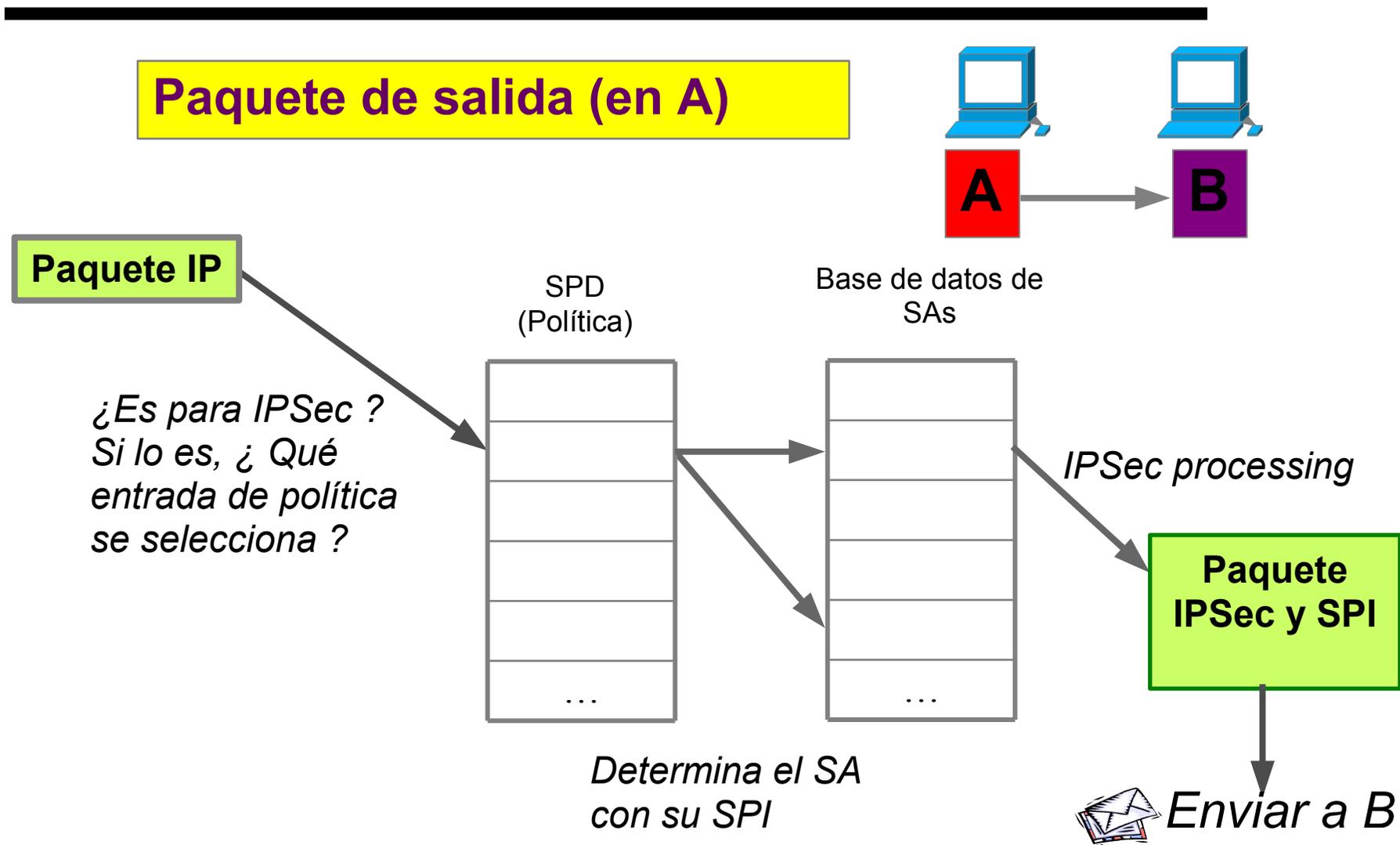
---



- ¿ Qué tráfico proteger ?
- Las entradas de la base de datos definen que SA o conjuntos de SA se utilizan en el tráfico IP
- Cada estación o encaminador tiene su propio SPD
- Indices del SPD acceden a campos de selección:
  - IP Destino, IP origen, Protocolo transporte, Protocolo IPSec (índices SA,...), Puestos origen y Destino,.....
- Acciones ligadas a entradas del SPD
  - No dejar entrar o salir, no aplicar o no esperar IPSec,
  - Proteger (aplicar o chequear seguridad). Si no existe SA:
    - Entrada : descartar paquete
    - Salida generar dinámicamente SA

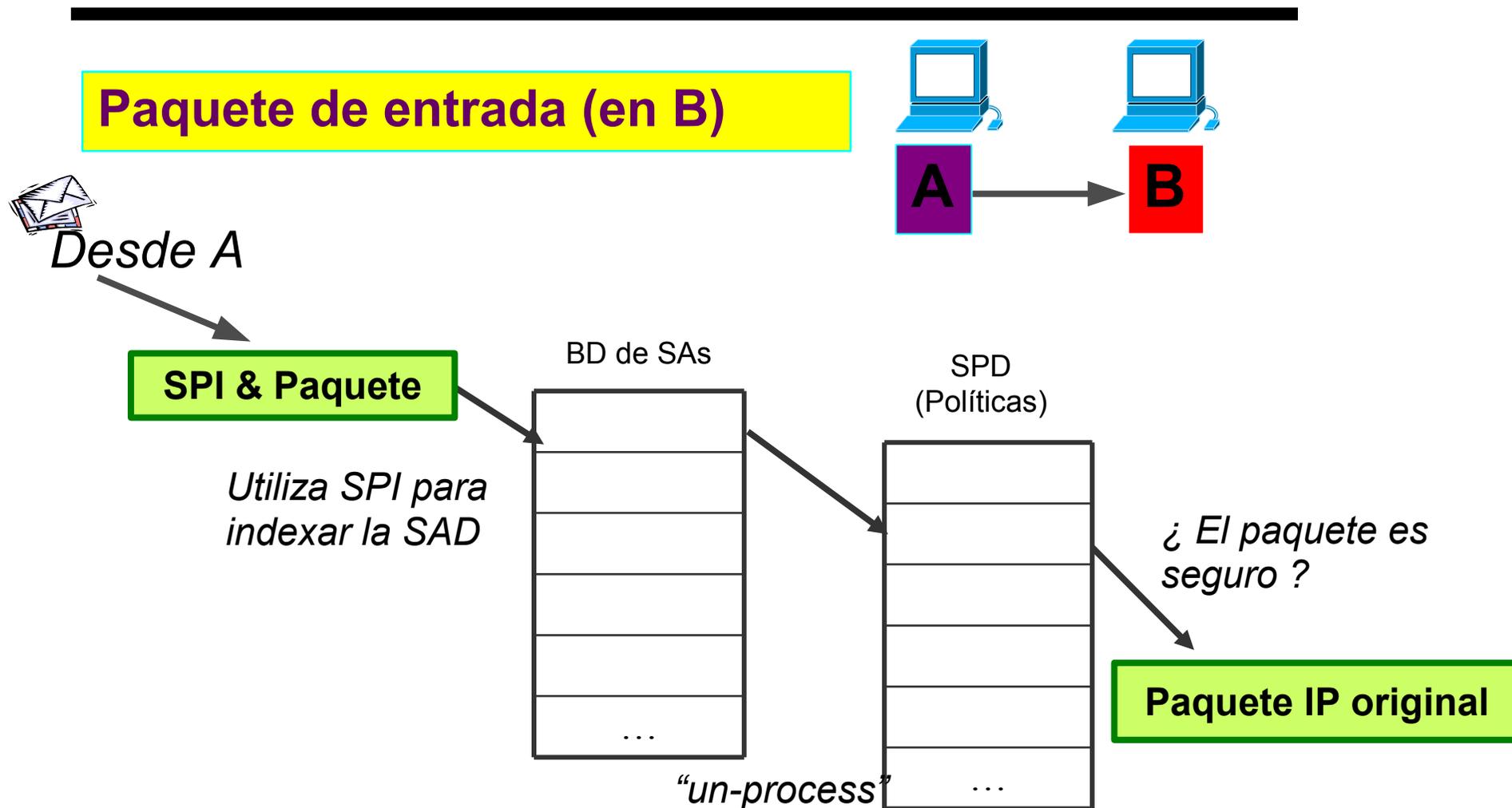


# Procesamiento de envío





# Procesamiento en entrada





# Gestión de claves

---

- AH y ESP requieren claves de cifrado y autenticación
- Proceso para negociar y establecer SAs de IPSec entre 2 entidades.
- Secretismo perfecto hacia adelante (PFS):
  - La captura de una clave no debe de dar acceso a todos los datos, solo a los datos protegidos por esa clave.
  - Claves no derivadas de sus predecesoras.
- Nonces : Números pseudo aleatorios generados localmente.
- Gestión manual de claves :



# Gestión manual de claves

---



- Obligatorio
- Util cuando desarrolladores de IPSec están depurando
- Intercambio de claves offline (teléfono, email, etc)
- Puesta en marcha de SPI y negociación de parámetros



# Intercambio de Claves en Internet (IKE)



- 
- Utilizado cuando un paquete de salida no tiene una SA
  - Dos fases:
    - Establecer una SA de IKE
    - Utilizar esa SA para negociar SAs en IPSec
  - El SA de IKE es utilizado para definir cifrado y autenticación del tráfico IKE
  - Múltiples SAs de IPSec pueden ser establecidas con una SA de IKE