# T2.1.2 Servicio de Protección de Acceso

Garantía y Seguridad de la Información.

# Protección de acceso: Concepto

- Reunión de los servicios de
  - Autenticación
    - Servicio que permite verificar que un agente es quién dice ser
  - Autorización (Control de acceso)
    - Servicio que permite ofrecer o denegar un acceso a un recurso
  - Auditoría
    - Servicio que permite seguir la pista a los intentos (exitosos o no) de autenticación.

# RFC-2903: El estándar de la triple-A (AAA)

- El estándar AAA (RFC 2903)
  - marco arquitectónico para configurar un sistema de seguridad de red, que proporciona:
  - Autenticación, Autorización y Auditoria.

- Existe un familia de protocolos que ofrecen los tres servicios citados.
  - Son mecanismos de control de acceso remoto y provisión de servicios de red

# AAA: Una familia de estándares

- RFC 2194 Review of Roaming Implementations
- RFC 2477 Criteria for Evaluating Roaming Protocols
- RFC 2881 Network Access Server Requirements Next Generation (NASREQNG) NAS Model
- RFC 2903 Generic AAA Architecture
- RFC 2904 AAA Authorization Framework
- RFC 2905 AAA Authorization Application Examples
- RFC 2906 AAA Authorization Requirements
- RFC 3169 Criteria for Evaluating Network Access Server Protocols
- RFC 3539 AAA Transport Profile
- RFC 1234 AAA Transport Profile

# Estándar AAA: Características

- Un protocolo AAA debe ser capaz de
  - autenticar a los usuarios,
  - dar una respuesta correcta a las solicitudes de autorización de los mismos,
  - recolectar datos que permitan hacer una auditoría sobre los recursos a los que se ha tenido acceso

# AAA. Autenticación

- **Autenticación**. Validar la identidad de los *principales*.
  - Probar que el *principal* posee una pieza única de información.
  - Probar que el *principal* posee una credencial de identificación no ambigua

- Un *principal* en seguridad informática es **cualquier entidad que puede ser autenticada por el sistema informática**.

- Variantes (ver más adelante):
  - Autenticación de usuarios
  - Autenticación de sistemas

# AAA. Autorización

- **Autorización**. Concesión de privilegios al *principal* basándose en su identidad (autenticada), los privilegios que solicita y el estado actual del sistema.
  - El privilegio concedido suele ser el uso de un determinado tipo de servicio.
    - Discrecional, Basado en roles, Obligatorio.

# AAA. Auditoría

- **Auditoría**. Seguir la pista y revisar los eventos, errores, accesos e intentos de autenticación de un sistema.

- A veces se denomina también **responsabilidad**

# AAA: Mecanismos de Control de acceso

- **Control de acceso**. Permite controlar el acceso a un recurso.
  - permite al usuario/administrador de un sistema restringir las posibles actividades, usos y contenidos del sistema.
- **Mecanismo**: Componente hardware o software que puede usarse para ofrecer o denegar el acceso a un recurso, en base a la autenticación del recurso y del accedente.
  - Ejemplos
    - una SmartCard, un dispositivo biométrico o un hardware de acceso a red (router,...)
    - los permisos de un archivo o un recurso compartido.

T2.1.2 Servicio de Protección de Acceso

# Autenticación de Usuarios

# Autenticación de usuarios

- La autenticación de un usuario, alguna característica que sólo posee el auténtico usuario y el sistema puede comprobar:
  - un secreto cuyo conocimiento tiene que comprobarse: PIN, clave, frase de paso, o información personal privada.
  - un ítem que sólo posee el usuario: llave, una tarjeta, una insignia, un sello,...
  - alguna característica única del usuario que es capaz de reconocer el sistema (biometría): una huella dactilar u ocular, una forma de teclear, de andar, la voz, el propio rostro,...

# Autenticación de usuarios

- Se basa en los denominados **factores**
  - Qué tiene?
    - llaves de la puerta, tarjetas de crédito, abre-puertas de garaje portátiles,…
  - Qué sabe?
    - Palabra contraseñas o frases de contraseña
  - Quién es?
    - Características biométricas
  - Dónde está?
- Cualquier factor sirve para autenticar
- Cuantos más factores se requieran, más sólida será la autenticación.

# AU: Factor "Qué tiene?"

- Forma muy débil de autenticación.
  - Las llaves de las puertas, los abre-puertas de garaje portátiles o las tarjetas de crédito se pueden perder o robar fácilmente.
- Solo se deben usar para realizar la autenticación multifactor.

# AU: Factor "Qué sabe?"

- Números de identificación personal (PIN)
  - A usar en dispositivos optimizados para la entrada de información numérica.
- Contraseñas (cadenas de caracteres ASCII imprimibles)
  - acceder a un sistema informático o acceder a una cuenta en un sistema remoto.

- El factor de conocimiento es superior al factor de posesión cuando se usa solo.

# Utilización de passwords: fortalecimiento

- El método más habitual para fortalecer el mecanismo de passwords es obligar a utilizar passwords difíciles de romper.
- También es posible utilizar autenticación multifactorial
  - limitando el uso del sistema a condiciones temporales y espaciales.
  - disparando alarmas sobre el uso del sistema en condiciones no habituales (horario, host-ip, protocolos, aplicaciones...).
  - obligando al usuario a probar su identidad mediante otros mecanismos durante la operación (introducción repetida de la clave, petición de datos personales del usuario,...)

# Ataques sobre passwords

- Por orden de complejidad son:
  - Ataque por fuerza bruta.
  - Ataque por diccionario (passwords frecuentes).
  - Ataque por probabilidad de usuario (passwords posibles del usuario).
  - Ataque dirigido por lista de passwords (cuando alguien proporciona la lista de passwords del sistema).
  - Ataque por trampa o amenaza al usuario.
    - Ocultando información de por qué falla un par <nombre, password>
    - Aumentando el tiempo de demora entre peticiones

# Passwords de un solo uso

- Utilizan un dispositivo de tamaño de una tarjeta de crédito
  - muestra una contraseña variable en el tiempo (llamado un código de acceso)
  - devuelve un código de acceso cuando se introduce un reto en un teclado.
- Los mecanismos desafío-respuesta son menos vulnerables a los ataques por repetición.
- El usuario o el sistema que se autentica proporcionan una clave distinta cada vez al servicio de autenticación.

# Passwords de un solo uso. Generación

- El usuario utiliza claves de una lista previamente acordada en base a un índice de una, dos o más dimensiones. Los valores de la lista deben ser aleatorios.
- El usuario utiliza un token proporcionado por el servicio de autenticación y un valor dependiente del tiempo para alimentar una función que aleatoriza el resultado.
  - Se confía en la sincronía de tiempos entre usuario y servicio de autenticación.
- El usuario encripta el token de desafío con una clave secreta previamente acordada.

# Passwords de un solo uso. Obtención

- Software: en computadoras y también mediante apps sobre smartphones.
- Mediante telefonía móvil (vía SMS, o similar)
- Mediante aplicaciones on-chip sobre smartcard con desafío-respuesta.
- Mediante aplicaciones on-chip sobre smartcard o tokens-USB que proporcionan consecutivamente claves de una lista (necesitan memoria para almacenar los passwords usados).
- Métodos vía Web basados en información privada del usuario.

# AU: Factor "Quién es?"

- Se utiliza algún atributo físico para autenticar.
  - Tasa de "falsos positivos"
  - Tasa de "falsos negativos"
  - Tasa de engaños
  - Intrusividad

- Este es el factor más fuerte cuando se usa solo.

# AU: Factor "Dónde está?"

- La ubicación se usa junto con el factor "saber algo".
  - Unix o Linux solo permitirán que un sujeto inicie sesión en la cuenta "root" (superusuario) desde la consola del sistema.

- No se debe confiar en este factor por sí solo para la autenticación

# AU: Combinación de factores (I)

- Acceso de cajero automático
  - el cliente tiene que usar una tarjeta de cajero automático (*algo que tiene*) y conocer el PIN asociado (*algo que sabe*).
- Acceso web a la cuenta bancaria
  - el cliente debe ingresar una contraseña (*algo que sabe*) y un número de un cryptotoken (*algo que tiene*).
    - El token criptográfico es un pequeño dispositivo que muestra un número durante un período corto de tiempo y luego lo cambia a un nuevo número no predecible.
      - Passwords de un solo uso

# AU: Combinación de factores (II)

- Acceso remoto a una red corporativa
  - Se usa una firma digital (credencial) creada con una clave privada almacenada de forma segura en una "tarjeta inteligente".
    - tarjeta de plástico con una unidad de procesamiento incorporada y un almacenamiento no volátil.
    - La clave privada se almacena en el almacenamiento no volátil de la tarjeta.
  - Hay que insertar la tarjeta inteligente (*algo que tiene*) en un lector de tarjetas inteligentes e introducir una contraseña (*algo que sabe*) para que el procesador de la tarjeta descifre la clave privada.
  - El procesador de la tarjeta utiliza la clave privada para crear una firma digital del mensaje de autenticación y exportar la firma fuera de la tarjeta.

# AU: Combinación de factores (III)

- Un sujeto debe usar una firma digital creada usando una clave privada almacenada en una tarjeta inteligente
  - El sujeto inserta la tarjeta (*algo que tiene*) en un lector de tarjetas
  - Introduce su contraseña (*algo que sabe*)
  - Coloca un dedo en un lector de huellas digitales (*algo que es*)
  - Y el procesador de la tarjeta descifra la clave privada y la usa para crear una firma digital de un mensaje de autenticación y exportar la firma fuera de la tarjeta.

# AU: Autenticación de doble factor

- Nos autenticamos con una contraseña o con PIN, *algo que sé*
- Y además con *algo que tengo* (un token USB o una tarjeta de coordenadas)
- Y con *algo que soy* (la huella, el iris, la voz o el rostro)

# AU: Autenticación en dos pasos

- Se utilizan dos factores del tipo "*algo que sé*"
  - por ejemplo contraseña y un código que nos envían por SMS o email.

- Dos mejor que uno: doble factor para acceder a servicios críticos INCIBE, 22/02/2017

T2.1.2 Servicio de Protección de Acceso

# Autenticación de Sistemas. Kerberos

# Autenticación de sistemas

- Actividad crítica para garantizar la seguridad de los sistemas informáticos en una red.
  - Sin el conocimiento de la identidad de un *principal* (persona o sistema) que solicita una operación, la operación no se debe permitir.
- Los métodos de autenticación tradicionales no son adecuados para su uso en redes de ordenadores
  - los atacantes supervisan el tráfico de la red para interceptar contraseñas y romper la seguridad del sistema.
- Los mecanismos al uso realizan autenticación y autorización en el mismo momento.

# Autenticación de Sistemas: Esquema general AAA

- El usuario final se conecta al dispositivo de entrada y solicita el acceso a la red.
- La funcionalidad "cliente *AAA*" recoge y envía las credenciales del usuario final al servidor *AAA*.
- El servidor *AAA* procesa los datos y devuelve una respuesta de aceptación o rechazo y otros datos relevantes al cliente *AAA*.
- El cliente *AAA* notifica al usuario final que le ha sido concedido o denegado el acceso para los recursos especificados.

# Autenticación de Sistemas: Enfoques

- Los enfoques utilizados para asegurar las autenticaciones a través de redes entre las máquinas están basados en:
  - el concepto de centros de distribución de claves (KDC)
  - el uso combinado de las firmas digitales, certificados digitales e infraestructuras de clave pública (PKI).

# Kerberos. ¿Qué es?

- El sistema Kerberos se desarrolló en la década de 1980 en el Instituto de Tecnología de Massachusetts (MIT) para el Proyecto Athena
- La versión 4 se lanzó a fines de la década de 1980.

# Kerberos. ¿Qué es?

- Es un protocolo de autenticación en redes de ordenadores.
- Permite a dos ordenadores en una red insegura demostrar su identidad mutuamente de manera segura.
- Se basa en criptografía de clave simétrica y requiere un tercero de confianza.

# Kerberos

- Los clientes quieren utilizar servicios de los servidores.
- Existe un mecanismo central, llamado centro de distribución de claves (KDC)
  - un servidor de autenticación (AS)
  - un servidor de autorización, servidor emisor de tickets (TGS)
    - Los tickets sirven para demostrar la identidad de los usuarios.

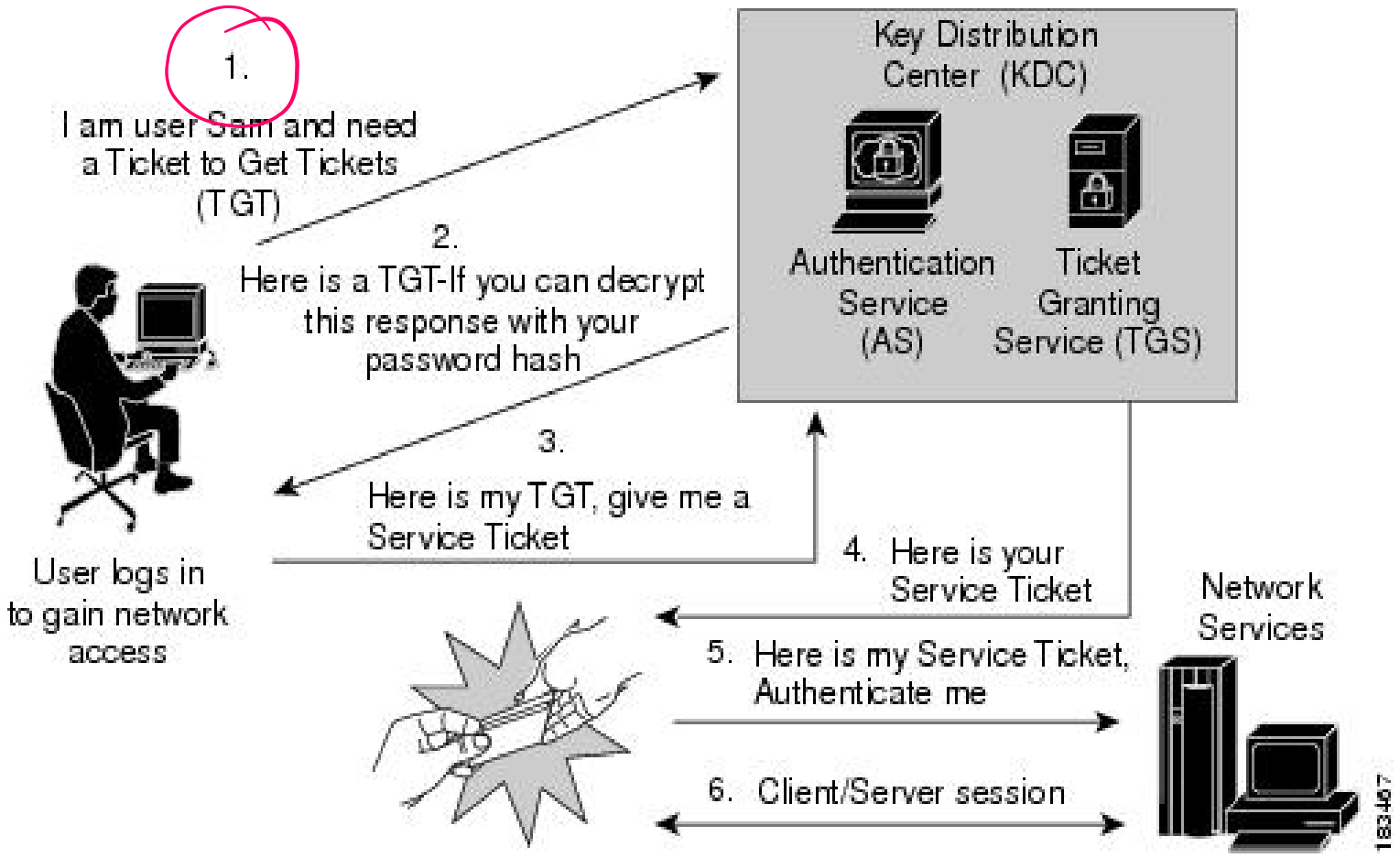
# Kerberos. ¿Cómo funciona?

- Cada cliente (o *principal*) tiene una clave secreta única que sólo conoce el cliente y el KDC.
  - El conocimiento de esta clave sirve para probar la identidad del cliente.
- El AS tiene registrados a todos los usuarios
  - guarda una clave que obtiene de su password.
  - Cada servidor de servicios (SS) comparte una clave con el AS.

# Kerberos. ¿Cómo funciona?

- El cliente se autentica ante el AS
- Demuestra al TGS que está autorizado a recibir un ticket de servicio (y lo recibe)
- Ya puede demostrar al SS que ha sido aprobado para hacer uso del servicio.
- Hace uso de él.

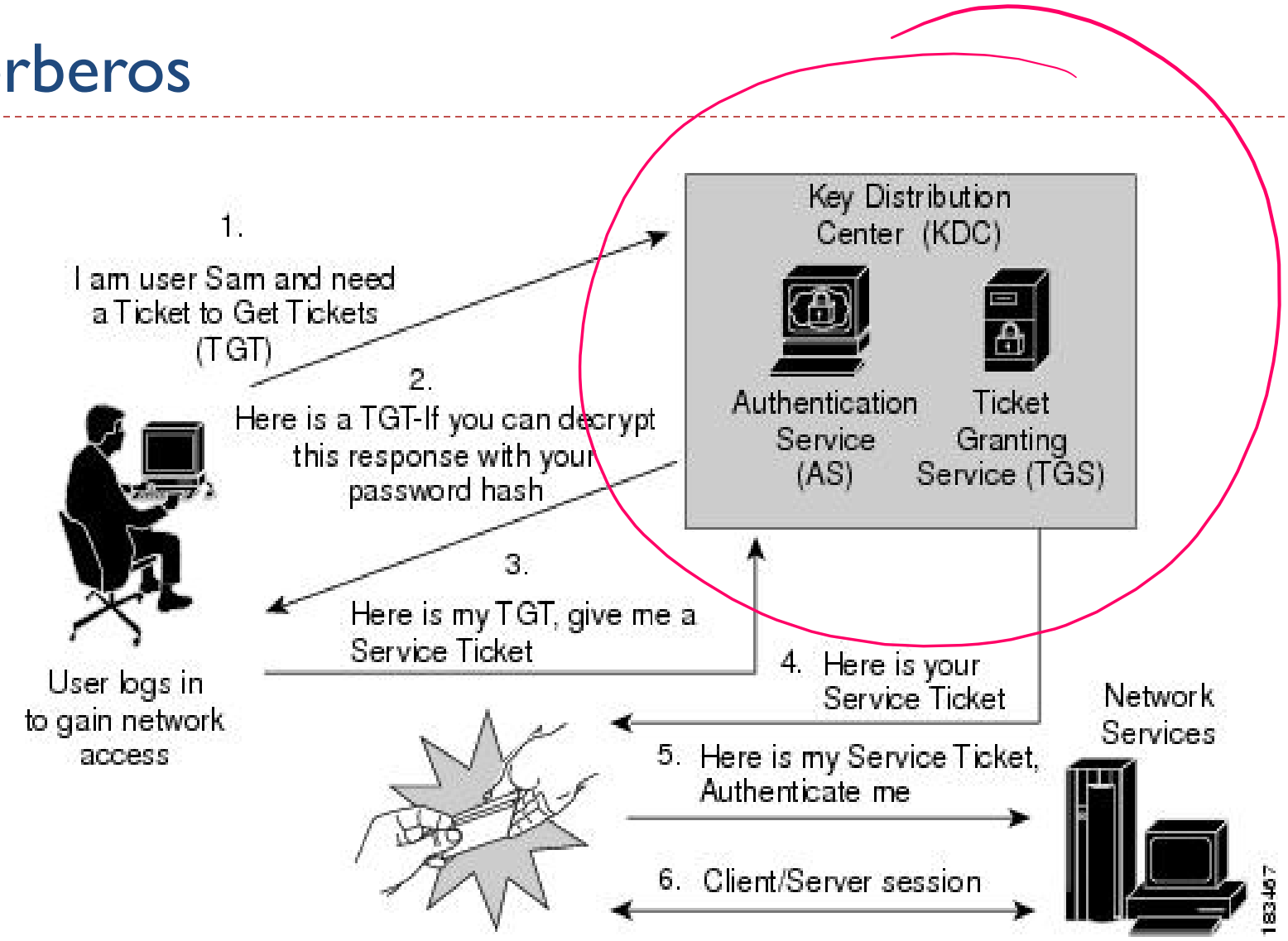
# Kerberos


1.
I am user Sam and need a Ticket to Get Tickets (TGT)
2.
Here is a TGT-If you can decrypt this response with your password hash
3.
Here is my TGT, give me a Service Ticket
4. Here is your Service Ticket
5. Here is my Service Ticket, Authenticate me
6. Client/Server session
Key Distribution Center (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
User logs in to gain network access
Network Services
183467

# Kerberos

1. Un usuario introduce su nombre de usuario y password en el cliente.
2. El cliente genera una clave hash a partir del password y la usa como la clave secreta del cliente.
3. El cliente envía un mensaje en texto plano al AS solicitando servicio en nombre del usuario.
   - Un mensaje de ejemplo podría ser "El usuario XYZ solicita servicios".
   - Ni la clave secreta ni el password son enviados, sólo la petición del servicio.

# Kerberos


Key Distribution Center (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
1. I am user Sam and need a Ticket to Get Tickets (TGT)
2. Here is a TGT-If you can decrypt this response with your password hash
3. Here is my TGT, give me a Service Ticket
4. Here is your Service Ticket
5. Here is my Service Ticket, Authenticate me
6. Client/Server session
User logs in to gain network access
Network Services
183467

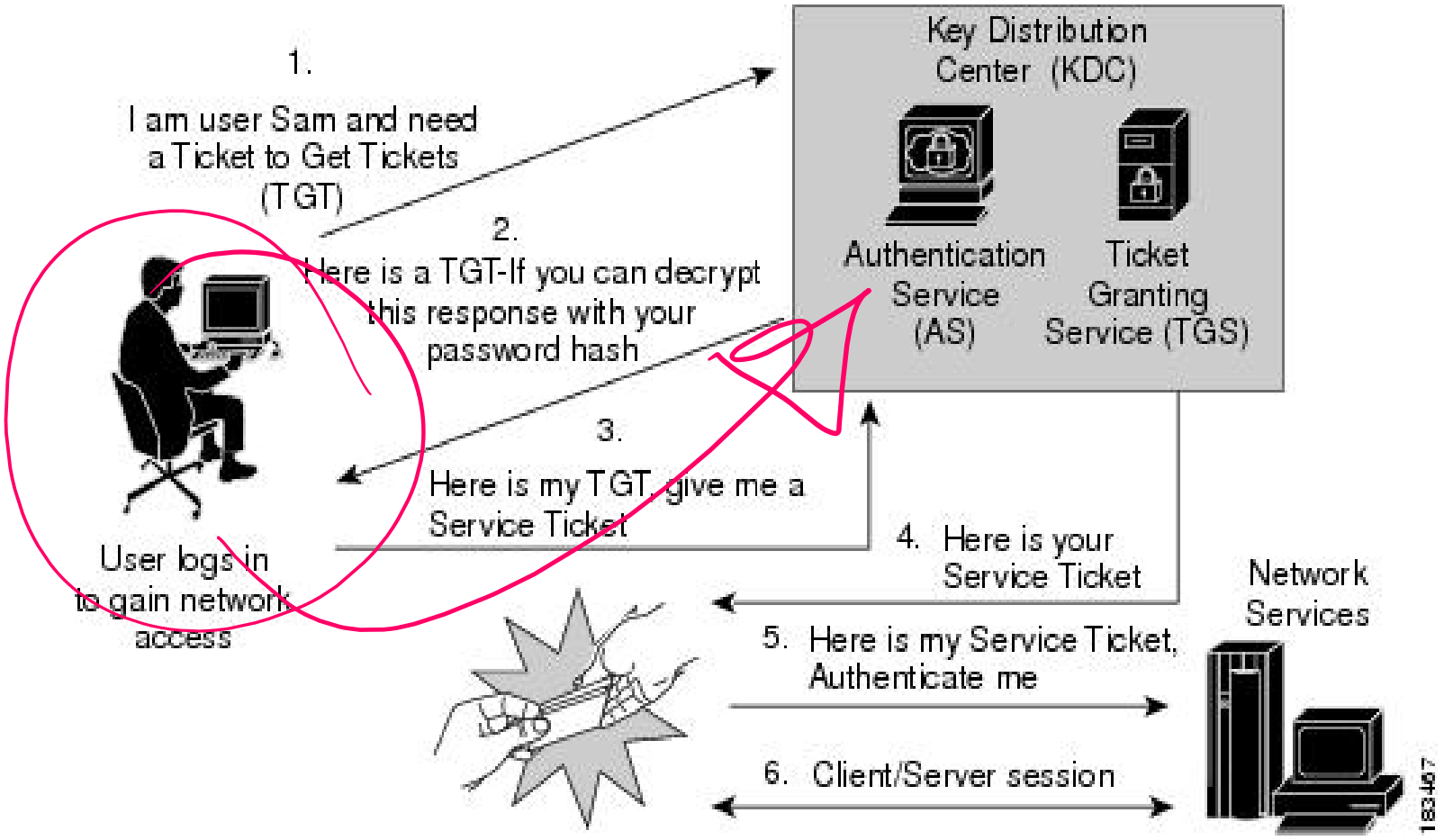
# Kerberos

4. El AS comprueba si el cliente está en su base de datos. Si es así, envía dos mensajes al cliente:

   Mensaje A: Clave secreta TGS/cliente usando la clave secreta AS/cliente.

   Mensaje B: Ticket (con ID de cliente, dirección de red del cliente, período de validez y clave secreta compartida por TGS con el cliente) cifrado usando la clave secreta TGS/AS.

# Kerberos


Key Distribution Center (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
1. I am user Sam and need a Ticket to Get Tickets (TGT)
2. Here is a TGT-If you can decrypt this response with your password hash
3. Here is my TGT, give me a Service Ticket
4. Here is your Service Ticket
5. Here is my Service Ticket, Authenticate me
6. Client/Server session
User logs in to gain network access
Network Services
183467

# Kerberos

5. Una vez que el cliente ha recibido los mensajes, descifra A para obtener la clave secreta TGS/cliente.

   Esta clave se usa para las posteriores comunicaciones con el TGS.

   El cliente no puede descifrar el mensaje B.

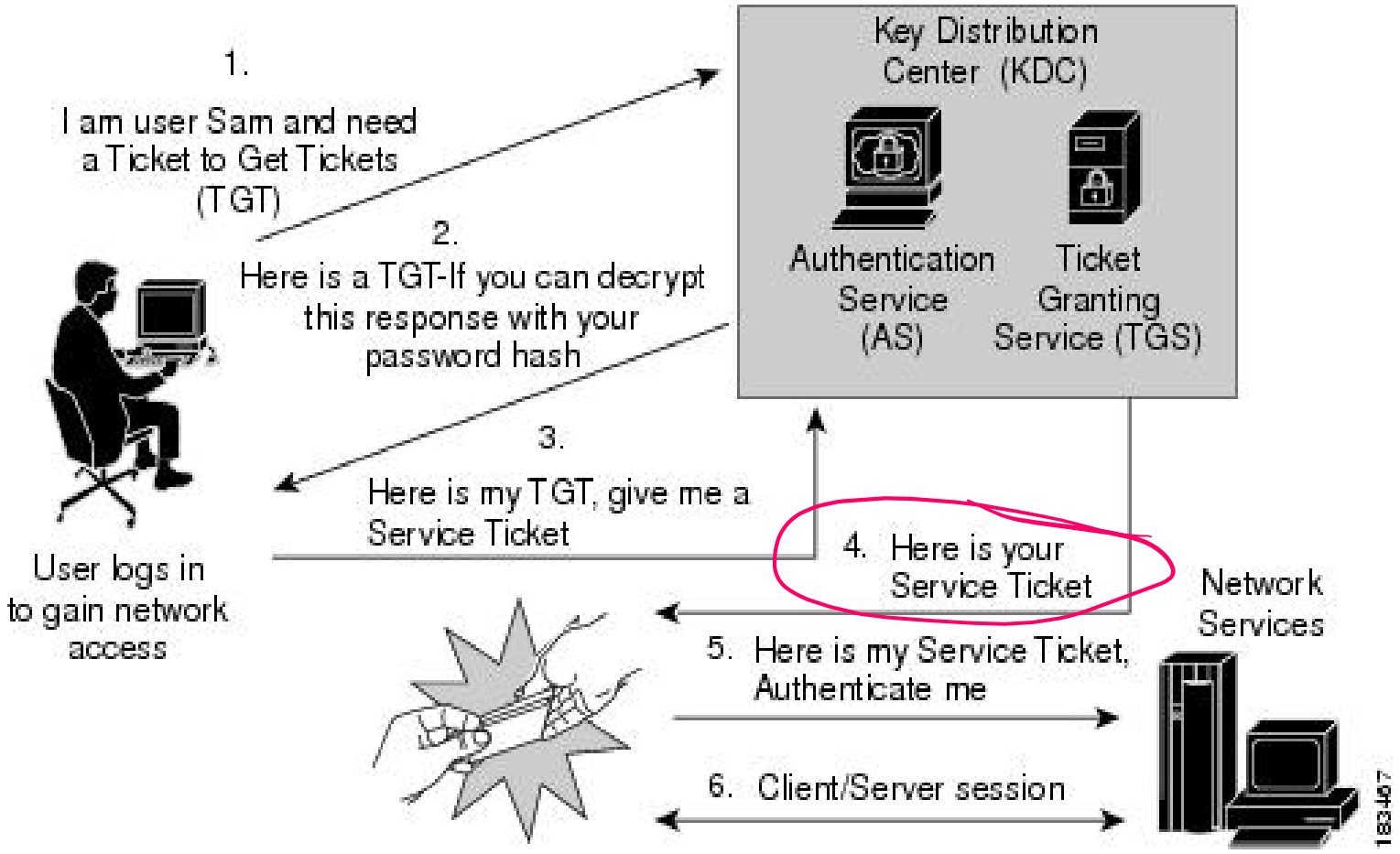
   En este momento el cliente ya se puede autenticar ante el TGS.

6. Entonces el cliente envía los siguientes mensajes al TGS:

   Mensaje C: Compuesto del ticket del mensaje B y el ID del servicio solicitado.

   Mensaje D: Autenticador (compuesto por el ID de usuario en el cliente y una marca de tiempo), cifrado usando la clave secreta TGS/cliente.

# Kerberos


Key Distribution Center (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
1. I am user Sam and need a Ticket to Get Tickets (TGT)
2. Here is a TGT-If you can decrypt this response with your password hash
3. Here is my TGT, give me a Service Ticket
4. Here is your Service Ticket
5. Here is my Service Ticket, Authenticate me
6. Client/Server session
User logs in to gain network access
Network Services
183467

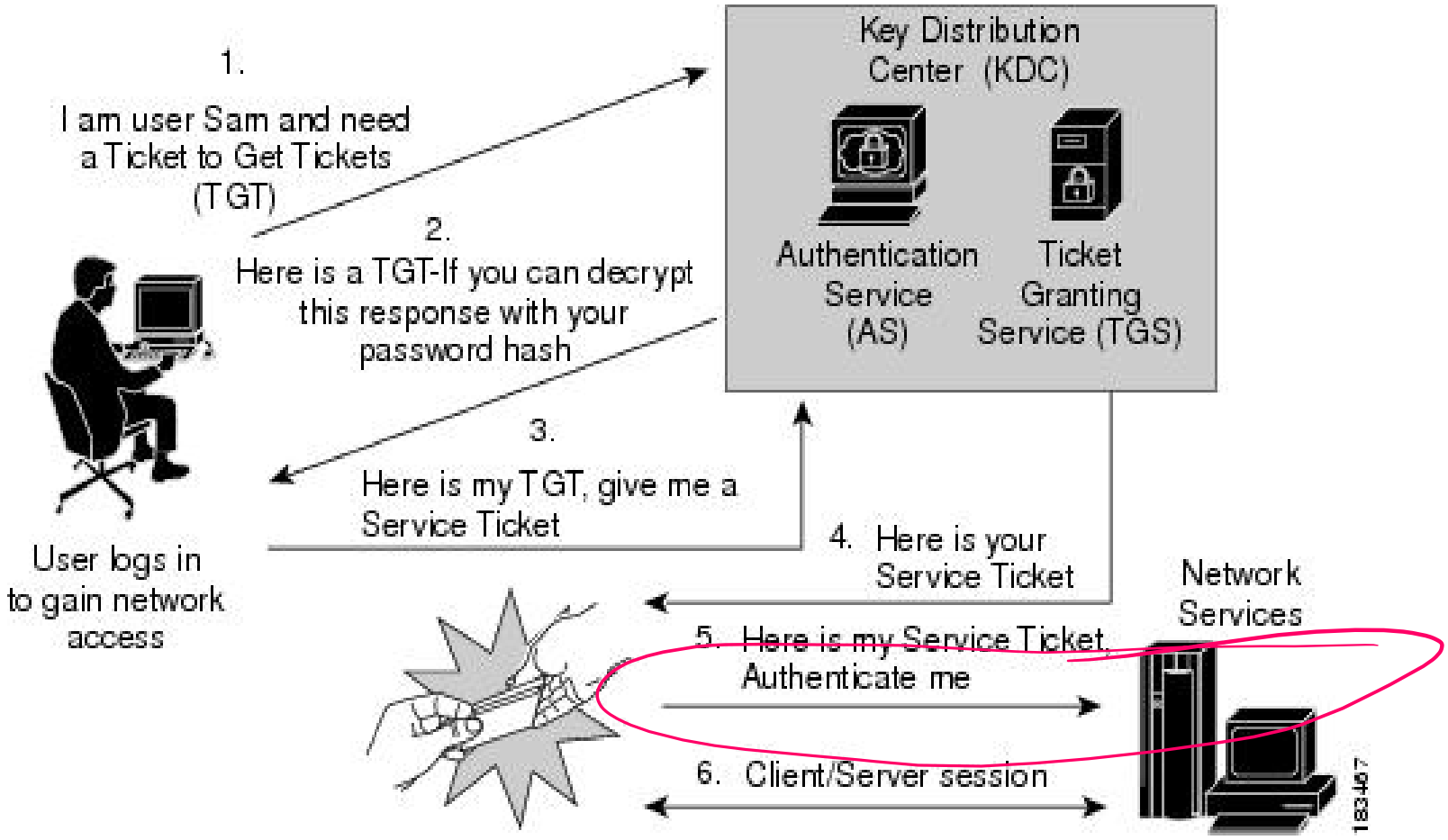
# Kerberos

7. Cuando recibe los mensajes anteriores, el TGS descifra el mensaje D (autenticador) usando la clave secreta TGS/cliente y envía los siguientes mensajes al cliente:

   Mensaje E: Ticket cliente-a-servidor (que incluye el ID de usuario en el cliente, la dirección de red del cliente, el período de validez y una clave secreta cliente/servidor para la duración de la sesión) cifrado usando la clave secreta TGS/servidor.

   Mensaje F: La clave secreta cliente/servidor para la duración de la sesión cifrada usando la clave secreta TGS/cliente.

# Kerberos


1.
I am user Sam and need a Ticket to Get Tickets (TGT)
2.
Here is a TGT-If you can decrypt this response with your password hash
3.
Here is my TGT, give me a Service Ticket
4. Here is your Service Ticket
5. Here is my Service Ticket, Authenticate me
6. Client/Server session
Key Distribution Center (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
User logs in to gain network access
Network Services
183467

# Kerberos

8. Cuando el cliente recibe los mensajes E y F, ya tiene suficiente información para autenticarse ante el servidor de aplicaciones. El cliente se conecta al servidor y envía los siguientes mensajes:

   Mensaje E del paso anterior.

   Mensaje G: un nuevo autenticador que incluye el ID del usuario en el cliente, una marca de tiempo, cifrado usando la clave secreta cliente/servidor para la duración de la sesión.

# Kerberos

9. El servidor descifra el ticket usando su propia clave secreta (compartida con TGS) y envía el siguiente mensaje al cliente para confirmar su identidad:

   Mensaje H: la marca de tiempo encontrada en el último autenticador recibido del cliente más 1, cifrado usando la clave secreta cliente/servidor para la duración de la sesión.

# Kerberos

10. El cliente descifra la confirmación usando la clave y comprueba si la marca de tiempo está correctamente actualizada. Si es así, el cliente confiará en el servidor y podrá comenzar a usar el servicio que este ofrece.
11. El servidor provee del servicio al cliente.

# Problemas de Kerberos

- Cualquier programa que lo utilice ha de ser modificado para poder funcionar correctamente
  - *"kerberización"*
    - Modificación para comunicación con el KDC.
- Gran centralización del sistema.
  - Se ha de disponer en todo momento del servidor Kerberos.
- Casi toda la seguridad reside en el servidor que mantiene la base de datos de claves.
- Uso de timestamps como prueba.
  - Todas las máquinas mínimamente sincronizadas.
- Utiliza cifrado simétrico con el algoritmo DES.

T2.1.2 Servicio de Protección de Acceso

# Infraestructura de Clave Pública

# Infraestructura de clave pública. PKI

- Infraestructura de red formada por servidores y servicios que, en base a claves públicas, gestionan de forma segura TODAS las transacciones realizadas.
- Se basa en dos operaciones básicas:
  - Certificación (medio por el cual las claves se publican)
  - Validación, a través de revocación y autentificación
    - El modo de realizar estas operaciones define las características de cada PKI.
- Las claves públicas pueden ubicarse en servicios de directorios, en autoridades de certificación, redes de confianza, ...

# Infraestructura de clave pública

- Con el mecanismo de clave pública, cualquier persona que posea una copia de la clave pública del remitente puede verificar una firma digital creada con la clave privada del remitente.
- Pero, ¿cómo puede un destinatario de un mensaje obtener la clave pública de un remitente válido?
- ¿El servidor de autenticación esté siempre accesible y en línea?

T2.1.2 Servicio de Protección de Acceso

# Protocolos AAA

# Protocolos AAA

- Protocolos AAA:
  - RADIUS
    - ***R**emote **A**uthentication **D**ial-**I**n **U**ser **S**ervice)*
  - TACACS (RFC 1492)
    - **T**erminal **A**ccess **C**ontroller **A**ccess **C**ontrol **S**ystem
  - TACACS+ (RFC 8907)
    - Incompatible con TACACS a pesar del nombre. Estandarizado recientemente
  - DIAMETER (RFC 3588)
    - Más allá de las conexiones de terminales …

# RADIUS

- **RADIUS (RFC 2865)**
  - Es un protocolo que proporciona autenticación, autorización y contabilidad (AAA).
  - Se utiliza por proveedores de servicios de Internet (ISP) para controlar el acceso del cliente.
  - Utiliza el puerto 1812 UDP para establecer sus conexiones.

# Secure Electronic Transaction (SET)

- Se orienta a asegurar transacciones Client-to-Business y Business-to-Business.
- Sus características de diseño más importante son:
  - El comerciante asegura el cobro de cierto bien.
  - El comerciante no puede conocer el estado económico del comprador.
  - La agencia que da soporte financiero al cliente no conoce el bien comprado.
- Se basa en firmas digitales para la autenticación mutua de sujetos y criptografía de clave secreta para la confidencialidad.
- Sujetos implicados: Clientes, Comerciantes, Emisores de Tarjetas de Crédito, Autoridad de Certificación con un PKI, Pasarela de Pago y un Intermediario de recepción de mercancías.

T2.1.2 Servicio de Protección de Acceso

# Control de Acceso. Mecanismos

# Control de acceso

- **Mecanismo lógico que** en función de la identificación ya autenticada y los derechos concedidos permite acceder a datos o recursos.
  - La *autorización* es la concesión de derechos por parte de un propietario o de un controlador de un recurso a otros usuarios para acceder a éste.

# Control de Acceso

- Comprobar cada acceso.
  - Aunque suponga una sobrecarga
  - Pues los permisos pueden variar.
- Garantizar siempre el mínimo privilegio.
  - Proteger el recurso, ofreciendo el acceso más limitado posible
  - Pues protegemos de vulnerabilidades adicionales.
- Verificar un uso aceptable.
  - La actividad realizada sobre el objeto es adecuada

# Mecanismos de control de acceso

- Decide si un acceso solicitado se permite o no.
- Objeto: Entidad pasiva que contiene, recibe o trata información.
  - A la que se trata acceder
- Sujeto: Entidad activa que actúa sobre un objeto.
  - El que quiere acceder

# Mecanismos de control de acceso

- Las decisiones se realizan en base a
  - Identidad del sujeto que realiza la petición.
  - Identidad del objeto al que se quiere acceder.
  - Atributo de acceso de la petición (r, w, x).
  - Autorizaciones actuales:
    - Se definen por la relación: sujetos, objetos, atributos (autorización).
    - Pueden cambiar con el tiempo.
  - Se debe cumplir que:
    - No puede saltarse.
    - No puede ser alterado.

# Matriz de accesos (ACM)

- Mecanismo que se utiliza tanto en bases de datos como en sistemas operativos
- Relaciona sujetos con objetos y con autorizaciones de los sujetos para acceder a los objetos.
- Es una matriz.
  - En vertical, los sujetos.
  - En horizontal, los objetos.
    - Los elementos de la matriz son los conjuntos de operaciones de acceso entre un sujeto y su objeto correspondiente.

# MATRIZ DE ACCESOS ACM

r = autorización de lectura
w = autorización de escritura
x = autorización de ejecución

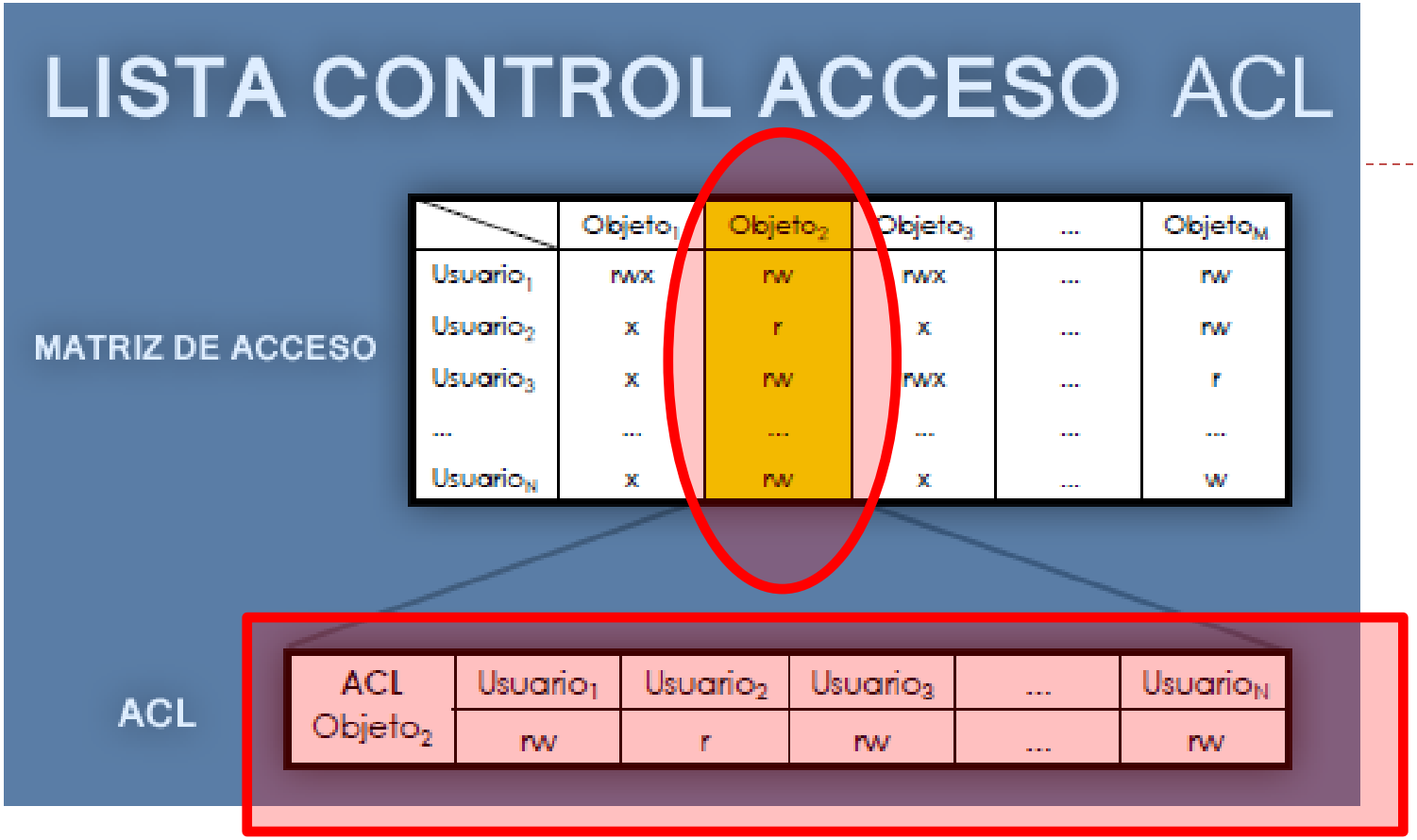
| | $Objeto_1$ | $Objeto_2$ | $Objeto_3$ | ... | $Objeto_M$ |
|---|---|---|---|---|---|
| $Usuario_1$ | rwx | rw | rwx | ... | rw |
| $Usuario_2$ | x | r | x | ... | rw |
| $Usuario_3$ | x | rw | rwx | ... | r |
| ... | ... | ... | ... | ... | ... |
| $Usuario_N$ | x | rw | x | ... | w |

Demasiado escasa de elementos, implementación ineficiente, gestión difícil, por tamaño.

Administración de estas estructuras de control de acceso consume mucho tiempo, es un proceso complicado y es susceptible a tener errores

# Lista de control de acceso (ACL)

- Permite asignar permisos concretos a usuarios (o grupos).
- Cada objeto tendrá una ACL
  - Conceder o impedir a cualquier sujeto el acceso al objeto concreto.
- La comprobación de TODOS los derechos de acceso de un sujeto concreto es poco eficiente.




- Es fácil determinar y revocar los permisos de un usuario sobre un objeto dado
- Es fácil revocar todos los permisos sobre un objeto dado.
- Es bastante costoso determinar y revocar todos los permisos de un usuario.

# Lista de control de acceso (ACL)

- Adecuado si
  - la gestión de control de acceso se realiza por objetos
  - las poblaciones de objetos son dinámicas.
- No adecuado si
  - la población de sujetos o grupos de sujetos cambia con mucha frecuencia.

# Habilitaciones/Capacidades

- Es un "testigo" utilizado para expresar los derechos de acceso de un sujeto a los objetos.
  - La capacidad para acceder a un objeto se demuestra cuando un sujeto posee una autorización o ticket.
  - El ticket contiene derechos de acceso permitidos como la lectura, la escritura o la ejecución.

# Habilitaciones/Capacidades

- Una lista **de capacidad** se corresponde con una fila de la matriz de control de acceso.
- Las listas de capacidad *se enfocan en los sujetos*.


- Resulta muy complicado determinar todos los sujetos que tienen acceso a un objeto concreto.

T2.1.2 Servicio de Protección de Acceso

# Modos de Control de Acceso

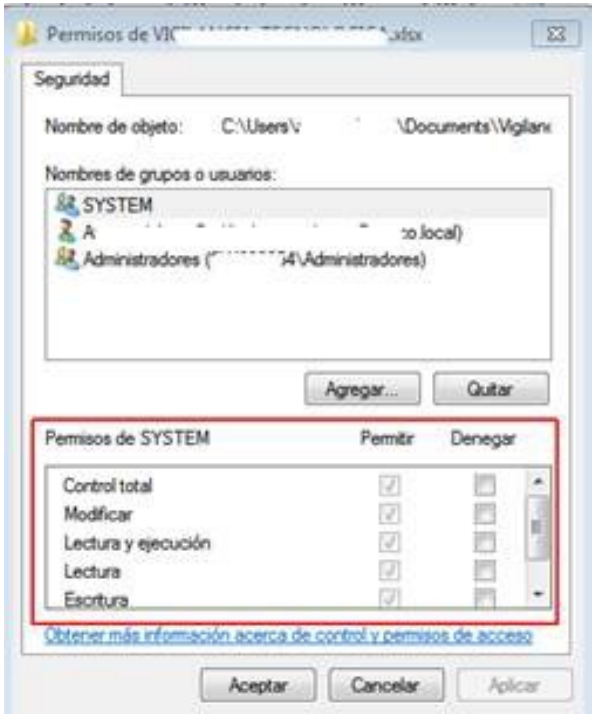
# Modos de control de acceso

- El dueño de la información decide sobre el acceso.
  - **DAC** o control de acceso discrecional basado en el *principio del propietario*.
- El papel que desempeñan los sujetos dentro de una empresa.
  - **RBAC** o control de acceso basado en roles.
- Las reglas incluidas en el sistema deciden sobre los accesos
  - **MAC** o control de acceso mandatario
  - Gobiernan el *principio del propietario*.

# Control de acceso discrecional (DAC)

- El dueño de la información decide sobre el acceso.
- Cada objeto tiene un sujeto propietario
  - sólo éste asigna permisos a otros sujetos.
- En la ACM, las operaciones de acceso se amplían con una más, p, la propiedad.
  - Se puede añadir la posibilidad de transferir la propiedad de un objeto a otro sujeto.
  - Puede existir delegación del derecho a asignar permisos.

# Control de acceso discrecional (DAC)




```
[usuario1@ ~]$ ls -l
total 0
-rwxrw-r--. 1 usuario1 grupo1 0 oct 21 09:45 fichero1
[usuario1@ ~]$
```

# Control de acceso discrecional (DAC)

- Problema del caballo de Troya:
  - Tres sujetos: $S_1$, $S_2$ y $S_3$.
  - $S_1$ es el propietario del objeto $O_1$, otorga permiso de lectura de $O_1$ a $S_2$, pero no se lo concede al tercer sujeto $S_3$.
  - El sujeto $S_2$ lee el objeto $O_1$, crea $O_2$ y copia el contenido de $O_1$ allí;
  - Otorga al sujeto $S_3$ permiso de lectura al objeto $O_2$.
  - El sujeto $S_3$ es capaz de leer el objeto $O_1$.

# Control de acceso basado en roles (RBAC)

- Se definen una serie de papeles (roles), que se asignarán según el perfil de cada usuario.
- Beneficios:
  - Sólo es necesario asignar usuarios y permisos a los roles.
  - Se puede utilizar herencia en la jerarquía de roles para reducir el número de asignaciones necesarias.
  - Simplifica la administración.

# Control de acceso basado en roles (RBAC)

- Ejemplos:
  - Oracle
  - Azure
  - Active Directory de Microsoft Windows

# Control de acceso mandatario (MAC)

- Conjunto de reglas de autorización
  - ¿una operación sobre un objeto realizada por un sujeto está permitida basándose en los atributos de ambos?
- Los objetos y los sujetos tienen unos atributos.
- Las reglas definidas se encargan de autorizar o denegar una acción.

# Control de acceso mandatario (MAC)

- Ejemplos:
  - SELinux
  - AppArmor
  - Sistemas multinivel

# Sistemas multinivel - MLS

- Un sistema de control de acceso es multinivel si:
  - Contiene objetos con diferentes niveles de sensibilidad (confidencialidad o integridad)
  - Admite sujetos con diferentes habilitaciones.
  - El acceso a los objetos se concede tras comprobar el nivel de sensibilidad del objeto con la habilitación del sujeto.
  - El objetivo de esos sistemas es el control estricto del flujo de información.

# Sistemas multinivel - MLS

- Sistemas multinivel
  - Bell-LaPadula:
    - Centrado en la confidencialidad, pensado para guardar secretos.
  - Biba:
    - Centrado en la la integridad.

# Modelo de Bell LaPadula (I)

- Se dividen las entidades en:
  - Objetos (documentos, BD, servicio web, …)
  - Sujetos (usuario, sistema remoto, …)
- Se establece el concepto de **estado seguro**:
  - Sólo un modo de acceso
  - En concordancia con la política de seguridad
- Se definen niveles de seguridad:
  - Priorizados, de más a menos protección o secreto
- Se asignan entidades a niveles
  - Cada objeto (p.e., documento) pertenece a un único nivel.
  - Cada persona también pertenece a un único nivel.

# Modelo de Bell LaPadula (II)

- Se definen 3 principios o **reglas para definir el acceso de objetos a niveles**:
  - 2 Obligatorios (MAC):
    - **La propiedad de seguridad simple**: un proceso que se ejecuta en el nivel de seguridad k sólo puede leer objetos de su nivel o de niveles inferiores (k'<k se lee hacia abajo).
    - **La propiedad estrella**: un proceso que se ejecuta en el nivel de seguridad k puede escribir sobre objetos de su nivel o de niveles superiores (k''>k se escribe hacia arriba).
  - Uno Discrecional (DAC):
    - Matriz de acceso: Especifica Control de Acceso Discrecional

- El **principio de tranquilidad** del modelo de Bell-La Padula.
  - Fuerte: la clasificación de un sujeto u objeto no cambia mientras está en operación el sistema
  - Débil: la clasificación no cambia de forma incompatioble con la política de seguridad (e.g.: principio de menor privilegio).

- **Modelo formal**: Si el sistema se mueve de estado seguro a estado seguro, se **puede demostrar que siempre cumplirá la política de seguridad.**

# Modelo de Bell LaPadula (III)

- Los usuarios pueden crear contenido solo en su nivel de seguridad o por encima.
  - En breve: **No leer hacia arriba, no Escribir hacia abajo**

- Investigadores de nivel secreto pueden crear archivos secretos o super secretos pero no archivos públicos.
  - Inversamente, los usuarios pueden ver solamente contenido de su propio nivel o inferior.

# Ejemplo de Bell LaPadula

Los usuarios tienen una etiqueta de seguridad con clasificación de acuerdo a niveles de confidencialidad como alto secreto, secreto, confidencial y público. Tenemos los siguientes elementos:

a) Clase de Acceso (sujetos y objetos)
   - X : {Sujetos, Objetos } → Nivel_de_Confidencialidad.

b) Relación "<=“:
   - posibilita un orden parcial y significa "domina a".

c) Derechos de Acceso:
   a) "A" puede leer "B" si y sólo si *X(B)* <= *X(A)* (Lectura descendente).
   b) "A" puede escribir sobre "B" si y sólo si *X(A)* <= *X(B)* (Escritura ascendente)
   c) "A" puede leer y escribir sobre "B" si y sólo si *X(A)* = *X(B)*.
   d) Acceso denegado en todos los demás casos.

Esto significa que la información no puede fluir hacia abajo, es decir, que el nivel de acceso del usuario nunca puede ser menor que la etiqueta del objeto.

# Modelo de Biba

- **Control de integridad en el flujo de la información**.
- **Objetivos**:
  - Impedir modificación de datos por partes no autorizadas.
  - Impedir modificación de datos no autorarizada por partes autorizadas.
  - Mantener consistencia interna y externa en los datos.

- Para determinar si un sujeto puede acceder a un objeto se comparan la habilitación del primero con la clasificación del segundo.

- Las propiedades de este modelo son:
  1. **Propiedad de seguridad simple**: Ningún proceso puede leer información de un nivel inferior.
  2. **Propiedad estrella**: Ningún proceso puede escribir información en un nivel superior.
  3. **Propierad de invocación**: Ningún proceso de nivel inferior puede solicitar acceso a nivel superior, sólo con sujetos de nivel igual o inferior.

- Cuanto mayor es el nivel de una etiqueta, mayor confianza en términos de integridad en el sujeto/objeto.