Control de accesos

Manuel Pons Martorell
Departament de Telecomunicacions
Escola Universitària Politècnica de Mataró

Agradecimientos

A tu, experta en controls, perquè em dones els moments més feliços encara que me'ls fas sofrir.

A l'Alex.

A en Leonard Janer i en Moisès Ortiz per la seva col·laboració en sistemes operatius.

A José Manuel Gómez de Kriptopolis y Gonzalo Alvárez Marañón de criptonomicon, puntales de la seguridad en español, por su desinteresada colaboración.

A Ignacio Baca Domingo de AENA por sus buenos consejos.

Índice

1. Introducción 3

1.1. Definiciones 3

1.2. Clasificación 3

2. Control de accesos por máquinas 3

2.1. Identificadores 3

2.2. Filtrado por dirección MAC 3

2.3. Filtrado por nombre o dirección de red y puerto. 3

2.4. Filtrado desde el servidor 3

2.5. Filtros con Routers o Firewalls 3

2.6. Ataques al control por IP o nombre 3

3. Control de accesos de usuario 3

3.1. Características generales 3

3.2. Control por contraseñas 3

3.3. Ataques a contraseñas 3

3.4. Defensas a ataques a contraseñas 3

3.5. Sistemas biométricos 3

3.6. Acceso con objetos físicos: Tokens 3

3.7. Acceso con certificados digitales 3

4. Autenticación Kerberos 3

4.1. Introducción 3

4.2. Características 3

4.3. Funcionamiento 3

4.4. Autenticación de usuario 3

4.5. Autenticación de servicios. 3

4.6. Instalación de Kerberos. 3

5. Autenticación Windows NT 3

5.1. Esquema general 3

5.2. Modelo de trabajo en grupo 3

5.3. Modelo de dominios 3

5.4. Relaciones de confianza entre dominios 3

5.5. Diferencias conceptuales con otros sistemas centralizados 3

Índice de figuras

Figura 1.1.1: Control unidireccional 3
Figura 1.2.1: Control en el servidor 3
Figura 1.2.2: Control en la red 3
Figura 2.1.1: Identificadores del paquete. 3
Figura 2.2.1: Bidireccionalidad en filtrado de MAC. 3
Figura 2.5.1: Apertura de sesión TCP. 3
Figura 2.5.2: Direccionalidad con filtros sencillos. 3
Figura 2.5.3: Direccionalidad con Firewall. 3
Figura 2.5.4: Conexión de RPC 3
Figura 2.6.1: Ataque con técnicas de tunneling 3
Figura 3.4.1: Algoritmo de una OTP 3
Figura 3.7.1: Algoritmos asimétricos 3
Figura 3.7.2: Control de accesos con certificados digitales 3
Figura 4.3.1: Proceso de autenticación Kerberos 3
Figura 5.2.1: Acceso compartido en modelo de trabajo en grupo 3
Figura 5.2.2: Acceso por usuarios en modelo de trabajo en grupo 3
Figura 5.3.1: Acceso por el modelo de dominios. 3
Figura 5.4.1: Acceso a un dominio de confianza. 3
Figura 5.5.1: Control de acceso Windows NT frente a otros de control remoto 3

1. Introducción

1.1. Definiciones

Realizar control de acceso a la información significa **seleccionar o filtrar** los usuarios que pueden acceder a recursos informáticos. Los sistemas que permiten el acceso a todos los usuarios y únicamente protegen de ataques de destrucción (denegación de servicios) no se pueden considerar control de accesos aunque si son parte de la seguridad y muchos autores los incluyen en este capítulo.

Entre los conceptos manejados por estas técnicas está el de **direccionalidad**, así un control de accesos permite unidireccionalidad si se puede hacer un control diferente para ir de A a B que para ir de B a A. Algunos sistemas de control no permiten la unidireccionalidad, así si se realiza un control de A a B automáticamente se instala el mismo control para ir de B a A (ver Figura 1.1.1).

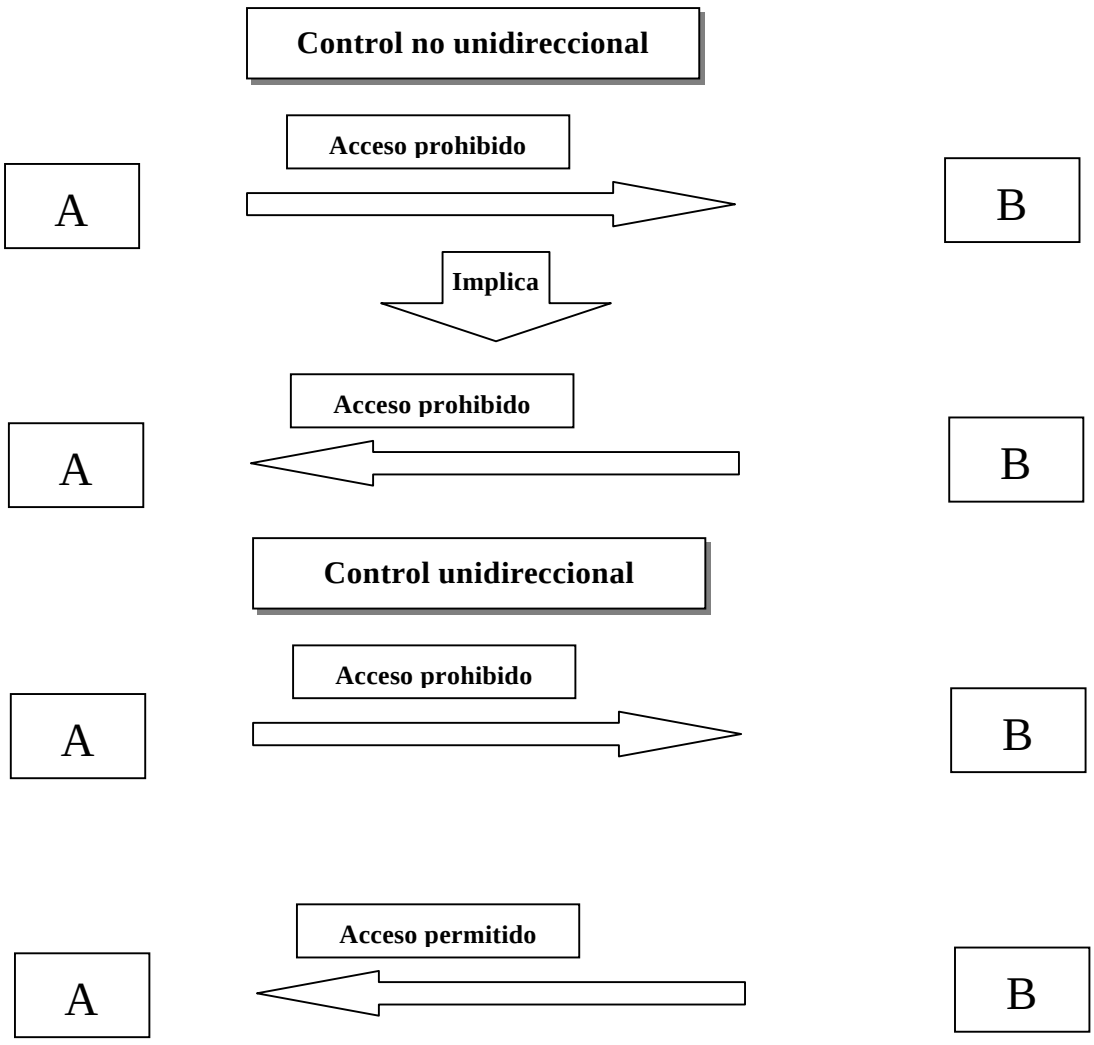

Figura 1.1.1: Control unidireccional

1.2. Clasificación

En las redes telemáticas se puede hacer la selección del acceso según **dos criterios**:

- La **máquina** (ordenador cliente).
- El **usuario** (persona o programa).

La selección por máquina sirve sólo para accesos remotos, o sea, no desde el ordenador donde está guardada la información. Normalmente son más seguros pero limitan la libertad de poder acceder desde cualquier ordenador, una prestación muy buscada desde la aparición de la red Internet.

Otro criterio de clasificación es dónde se pone el sistema de control. Se pueden agrupar dos familias:

- En el **servidor**.

El ordenador que almacena la información instala los **filtros en el sistema operativo o en el software**. Esto permite realizar la selección también para los usuarios que acceden físicamente al ordenador (Ver Figura 1.2.1).

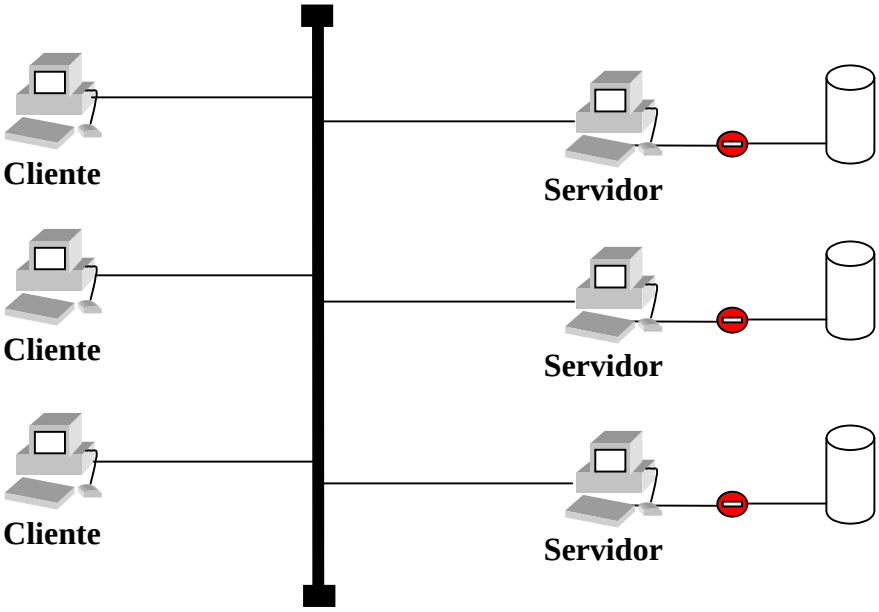

Figura 1.2.1: Control en el servidor

Un problema es gestionar el acceso de una lista de los mismos clientes en muchos servidores. Así el administrador se puede ver obligado a **actualizar todas las modificaciones en cada uno de los servidores**. Para **solucionar** esto hay sistemas que centralizan la gestión, en este trabajo se analizan dos: **Kerberos** y **Windows NT** (Capítulos *4 Autenticación Kerberos y 5 Autenticación Windows NT*).

- En la **red**

Se instalan **filtros en la red** que controlan los accesos desde máquinas remotas. No sirven para accesos físicos al servidor. Estos filtros se pueden implementar en **Switchs LAN**, **Routers** o **Firewalls**. Permiten controlar zonas y grupos filtrando únicamente la posibilidad de acceso (ver Figura 1.2.2).

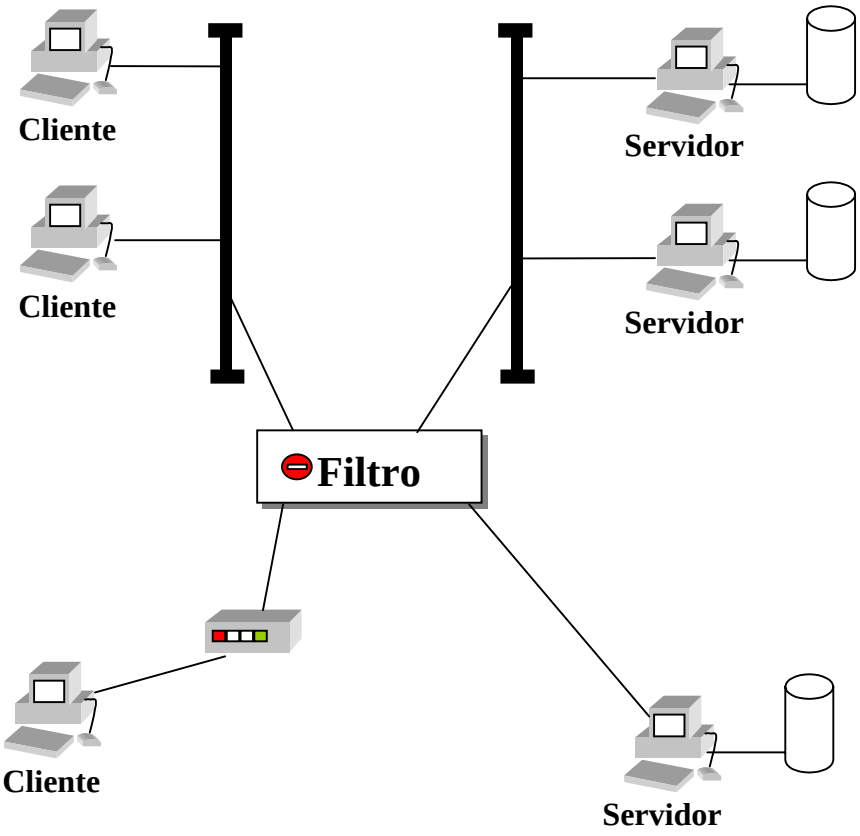

Figura 1.2.2: Control en la red

Otra forma de clasificar puede ser atendiendo a cuestiones más técnicas: protocolos, tipos de servicio, técnicas de Hacking, etc... Pero entonces las clasificaciones son más difusas y no ayudan a centrar conceptos sobre el control de accesos.

2. Control de accesos por máquinas

2.1. Identificadores

Este tipo de control selecciona a partir de la máquina utilizada para acceder. Por lo tanto, se deben poder identificar las máquinas y diferenciarlas y, si es posible, agruparlos en familias. Una máquina se puede **identificar** por:

- **Número de serie del procesador**.
- **Dirección MAC**.
- **Dirección IP o de otro protocolo de red**.
- **Nombre Internet**.

El **número de serie** del ordenador no se utiliza normalmente. En general los procesadores no tienen números de serie accesibles por el software excepto algunos de estaciones de trabajo.

Actualmente los nuevos Pentium III parece ser que llevarán un número de serie accesible, pero utilizarlo es poco ético. Así hay numerosos grupos de protesta contra esta nueva característica no evitable. En las estaciones de trabajo se utilizan los números de serie en el control de ventas o actualizaciones de software asignado a una máquina, así se evita la piratería informática. Este sistema también se puede realizar con la dirección MAC o IP, pero entonces se puede cambiar de máquina.

Los únicos datos que viajan en los paquetes son **las direcciones MAC y las direcciones IP** de la maquina origen (emisora) y destino (receptora) como indica la Figura 2.1.1. Así son los únicos identificadores que pueden utilizar legalmente los filtros.

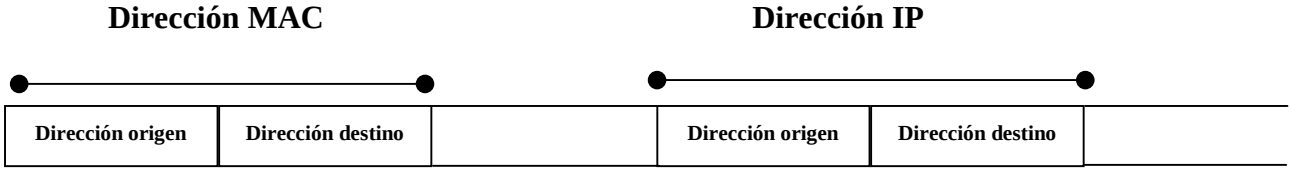

Figura 2.1.1: Identificadores del paquete.

Los **nombres de las máquinas** no están en los paquetes, ¿cómo se puede hacer para seleccionar un acceso con el nombre?. El filtro pide a un **servidor de nombres (DNS**) el nombre de la máquina a partir de su dirección origen IP, es el proceso de DNS inverso.

2.2. Filtrado por dirección MAC

Las direcciones MAC identifican las máquinas para **los protocolos de enlace de las redes LAN**. Así estas direcciones están en los paquetes de los protocolos de LAN: Ethernet, Token Ring, ATM, FDDI, etc...

Las **desventajas** de este método son:

- **Números difíciles de tratar**

La asignación de la dirección se hace por Hardware, o sea vienen programadas en las tarjetas de red, excepto raras excepciones. Todas las direcciones son únicas en el mundo pero sus valores no tienen más relación entre ellas que el proceso de fabricación, por lo tanto, las direcciones MAC de las tarjetas de los ordenadores de una empresa no tienen ninguna característica común.

- **Sólo se pueden utilizar en el entorno de la LAN**.

Sólo se utilizan dentro del entorno de una LAN, o sea, una red de Hubs y Switchs, así cambian cuando pasan por un Router y no existen en las máquinas aisladas conectadas telefónicamente a Internet. No se puede realizar control a nivel WAN.

- **No pueden discriminar entre servicios**.

No se puede discriminar por servicio porque en la cabecera del paquete de la capa MAC no hay información del servicio, sólo de la máquina destino de la misma LAN (puede ser un Router).

- **Sólo se pueden realizar desde filtros**.

Las aplicaciones no tienen acceso directo a la dirección MAC del paquete que llega, a menos que actúen por debajo del sistema operativo, que no es normal. Así esta forma de control siempre se realiza en la red mediante los equipos de interconexión, los Switchs y sistemas de VLAN.

- **Siempre son bidireccionales**.

No permiten controles unidireccionales, si se prohibe el acceso a la máquina A a B, B tampoco podrá acceder a A (Ver Figura 2.2.1).

Las **ventajas** son:

- **Método difícil de atacar**.

Es muy difícil falsificar la dirección origen del paquete. Además no se puede cambiar por software.

- **El filtrado es muy rápido**.

Como actúan a nivel dos, el filtrado se hace con elementos Hardware y, por lo tanto, es muy rápido.

Se utilizan poco y para casos especiales donde la separación entre las dos áreas quiere ser total y sin unidireccionalidad.

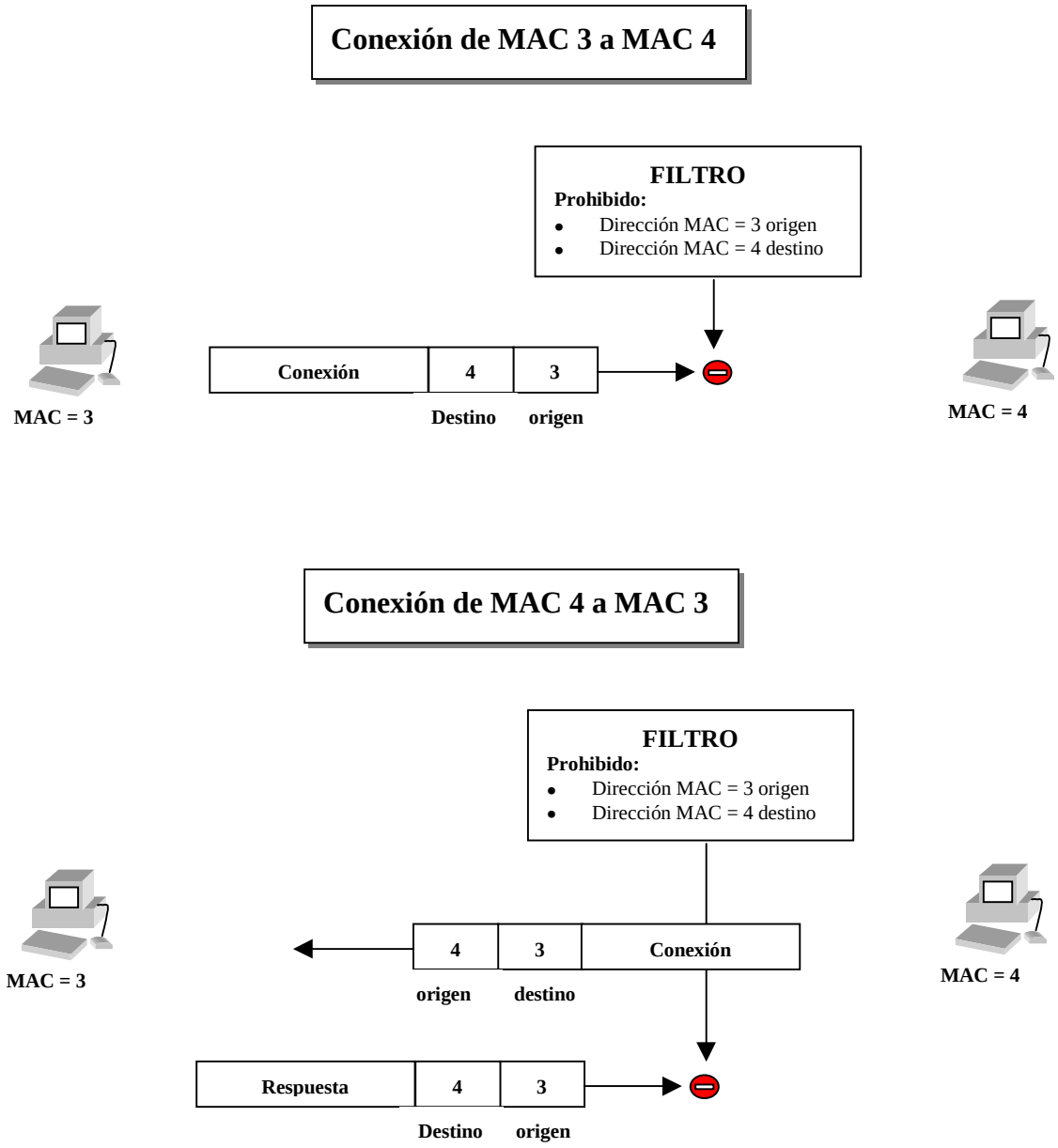

Figura 2.2.1: Bidireccionalidad en filtrado de MAC.

2.3. Filtrado por nombre o dirección de red y puerto.

Estos filtrados se realizan prohibiendo el acceso de una máquina identificada por una dirección IP o un nombre a un puerto concreto otra máquina también identificada por una dirección IP o un nombre. Así se permite **realizar un control de accesos a cada servicio**.

Tratar con **nombres de Internet** es más sencillo porque son más fáciles de recordar y tratar, pero tiene el inconveniente de que el nombre no viaja con el paquete. Así cuando el sistema de control filtra sólo conoce la dirección IP de la máquina que envió el paquete. Para saber el nombre **debe preguntar a un servidor de nombres (DNS)**

mediante el protocolo de DNS inverso. Esto debilita la seguridad porque introduce un elemento más para romper por el atacante, se puede modificar temporalmente la información del DNS o suplantar su mensaje de respuesta.

Las **direcciones IP y los nombres** son mucho más manejables que las direcciones MAC. **Se actualizan por softwa**re y siempre obedecen a una lógica. Los nombres son elegidos por la empresa y, por lo tanto, están relacionados. Las IPs se agrupan por clases o subclases ligadas a una empresa, una zona geográfica o cualquier grupo, así es fácil realizar filtros que afecten a todo un grupo de máquinas de la misma zona sin necesidad de introducir las direcciones una a una. Los grupos de direcciones IP afines se definen con las máscaras.

Así los filtros de IP se pueden definir prohibiciones o permisos con las siguientes posibilidades:

Máquina/s origen	**Máquina/s destino**	**Prohibición o permiso de**
IP x.x.x.x	IP x.x.x.x	Acceso completo de origen a destino.
IP x.x.x.x Máscara x.x.x.x	IP x.x.x.x	Acceso completo del grupo de máquinas origen a destino.
IP x.x.x.x Máscara x.x.x.x	IP x.x.x.x Máscara x.x.x.x	Acceso completo del grupo origen al grupo destino.
IP x.x.x.x	IP x.x.x.x Puerto y	Acceso de origen a un servicio de destino.
IP x.x.x.x Máscara x.x.x.x	IP x.x.x.x Puerto y	Acceso del grupo de máquinas origen a un servicio de destino
IP x.x.x.x Máscara x.x.x.x	IP x.x.x.x Máscara x.x.x.x Puerto y	Acceso de un grupo de máquinas origen a un servicio del grupo destino.

Tabla 2.3.1: Filtros de IP.

El control se puede hacer desde el servidor o desde un filtro intermedio (Router o Firewall). En los siguientes apartados se analizan las dos posibilidades.

2.4. Filtrado desde el servidor

Las **direcciones IP origen son accesibles desde el sistema operativo y las aplicaciones**, por lo tanto en el servidor se pueden montar filtros por IP o nombre.

En UNIX existen dos ficheros donde se pueden hacer un listado de las máquinas que tienen acceso o no a los servicios. Los ficheros son:

- *Host.allow*. Permite el acceso al servicio indicado a la IP o nombre indicado.
- *Host.deny*. Prohibe el acceso al servicio indicado a la IP o nombre indicado.

Si no se indica el acceso está permitido. La sintaxis es un fichero tipo texto donde cada línea es un filtro. Por ejemplo:

FTP: 145.22.22.22
Telnet: 122.33.33.33 255.255.255.192
ALL: 1195.1.1.0 255.255.255.0 (Para todos los servicios).

Muchas aplicaciones también permiten crear una lista propia de máquinas que tienen el acceso prohibido o necesitan password para acceder. Un ejemplo son los servicios r..., como el rlogin, rusers, rsh, donde se puede crear una lista de nombres con las máquinas que no necesitan password. Estas listas son muy peligrosas, para los atacantes es muy fácil hacer creer que su IP es la que corresponde a un nombre de la lista.

2.5. Filtros con Routers o Firewalls

Muchos Routers permiten crear filtros de IP y máscara origen a IP, máscara y puerto destino, cumplen las necesidades comentadas en los anteriores apartados.

Los **Firewalls** realizan funciones más avanzadas, permiten **filtrar mirando características de la capa de transporte**. Así el control se realiza de forma diferente para las distintas capas de transporte, con las siguientes posibilidades:

- En **TCP** únicamente se controla **el paquete de apertura de sesión**.
- En **UDP y ICMP**, que no abren sesiones, se realiza la **técnica de inspección de estados**.
- Se realizan **técnicas especiales** para los servicios **TCP con asignación dinámica de puertos**, como FTP o RPC.

El protocolo **TCP es orientado a conexión**, por lo tanto, antes de la transmisión de información debe realizar una apertura de sesión correcta y al finalizar un cierre de sesión. La apertura de sesión se realiza mediante 3 mensajes como indica la Figura 2.5.1. **Los Firewalls únicamente controlan los paquetes de conexión**, que se diferencian de los otros porque llevan el bit de ACK a '0'. Con este procedimiento se consiguen dos mejoras:

- **Más velocidad** porque el Firewall solamente examina un paquete de cada sesión.
- **Unidireccionalidad** en los casos de filtros para grupos de puertos.

Cuando el filtro se programa para prohibir el acceso a un puerto determinado normalmente no hay problemas de direccionalidad porque el puerto utilizado como cliente no es el mismo que el utilizado como servidor. Así si se programa prohibido el acceso de la máquina A al puerto 80 de la máquina B, la máquina B no tendrá problemas para acceder al 80 de la máquina A, ya que su puerto origen no será el 80 y recibirá sin problemas las respuestas a sus peticiones, ver Figura 2.5.2.

El problema aparece cuando se realizan prohibiciones como:

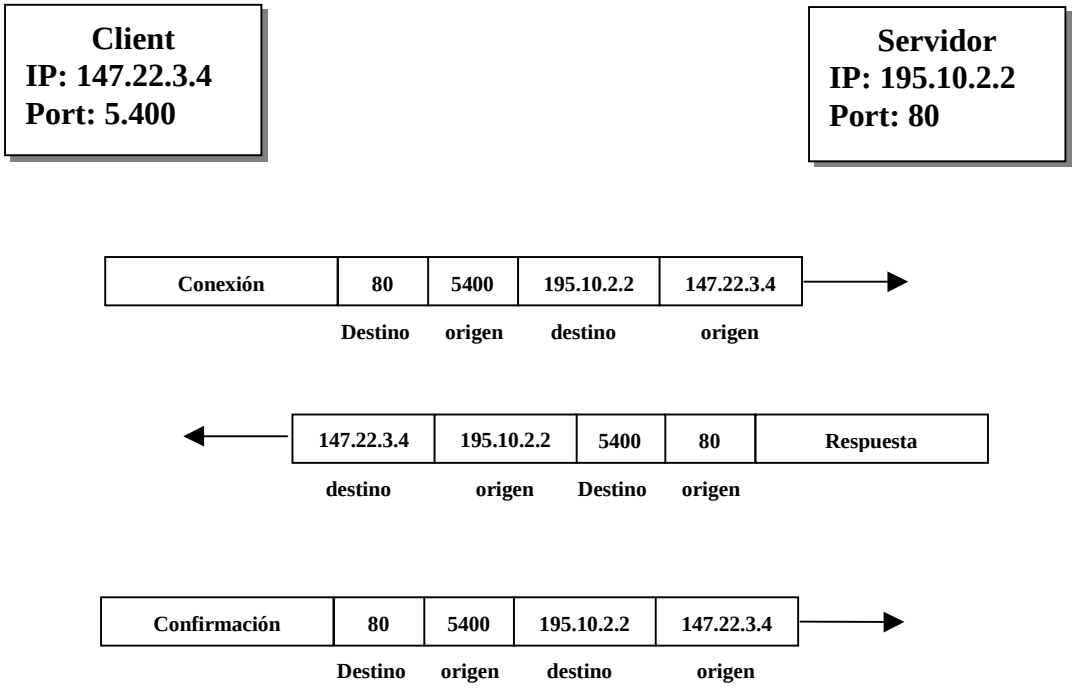

Figura 2.5.1: Apertura de sesión TCP.

- Prohibido el acceso a todos los puertos de la máquina A menos el 80 (una máquina que únicamente da servicio de Webs).

En este caso la máquina A no podrá conectarse a otras máquinas porque normalmente cuando hace de cliente utiliza los puertos de números mayores que 5.000. Así la solución es controlar únicamente el primer mensaje de inicio de conexión y permitir el paso de los mensajes de respuesta, ver Figura 2.5.3.

Los protocolos **UDP e ICMP no son orientados a conexión**, se envía información sin necesidad de haber abierto una sesión anteriormente. Por lo tanto no se pueden controlar los primeros paquetes, o se controlan todos o no se controla ninguno. Para evitar problemas de direccionalidad se utiliza la **técnica de inspección de estados**. Cuando se filtra con inspección de estados en principio se permite el paso, pero se guarda memoria de los paquetes que van pasando entre las dos máquinas para cada par de puertos. Así conociendo toda la historia de la comunicación y con la inteligencia de un Firewall puede detectar si se realiza algún ataque y cortar si es necesario.

En **algunos servicios**, como FTP o los que utilizan RPCs, el servidor **utiliza puertos distintos en cada conexión**. El cliente realiza siempre la primera conexión al mismo puerto pero durante la transmisión el servidor puede utilizar otros puertos para sesiones concretas, este puerto es comunicado al cliente durante el traspaso de información por el puerto inicial. Si el Firewall quiere hacer una programación como:

- Prohibido el acceso a todos los puertos excepto el 80 y 111.

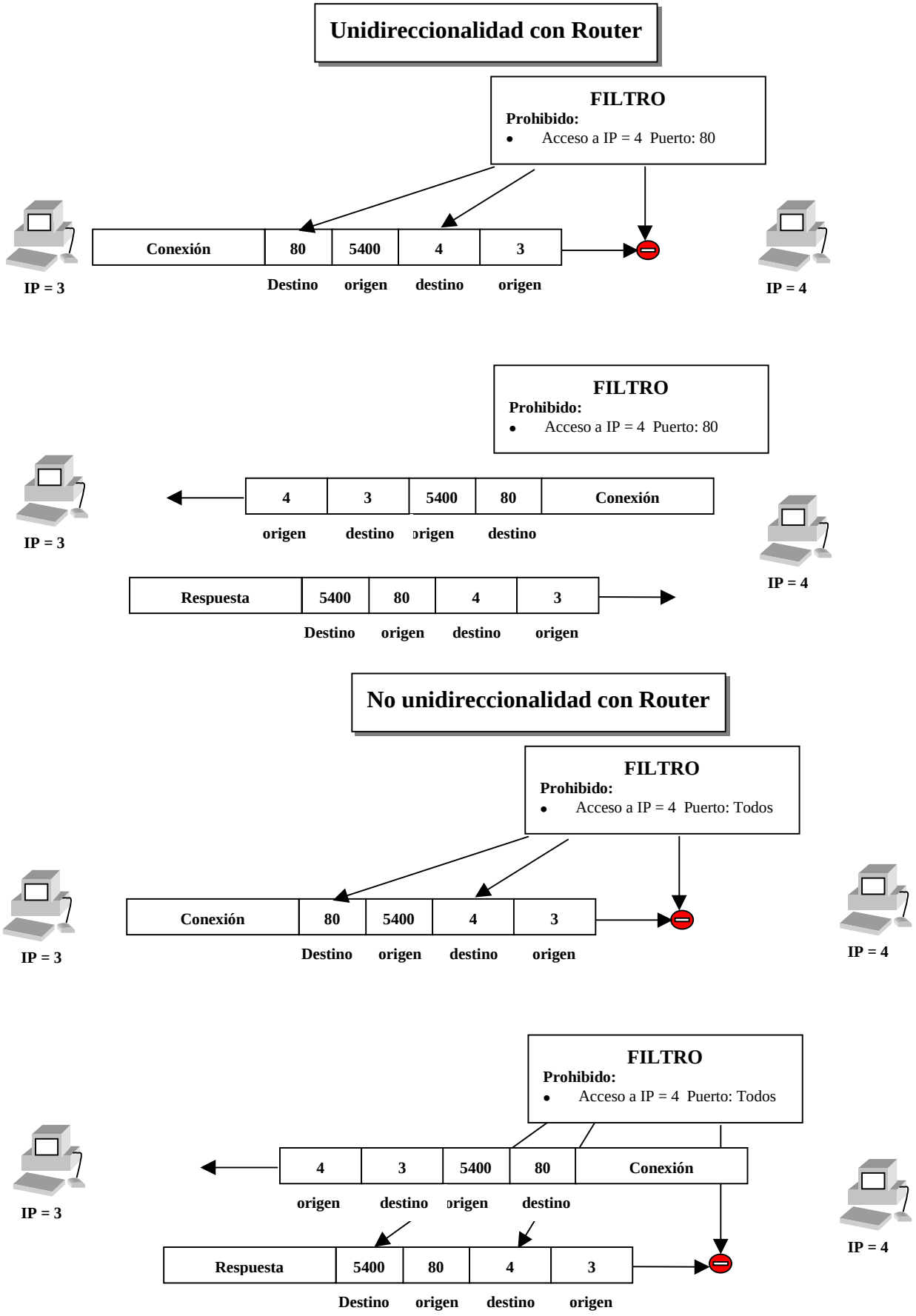

Figura 2.5.2: Direccionalidad con filtros sencillos.

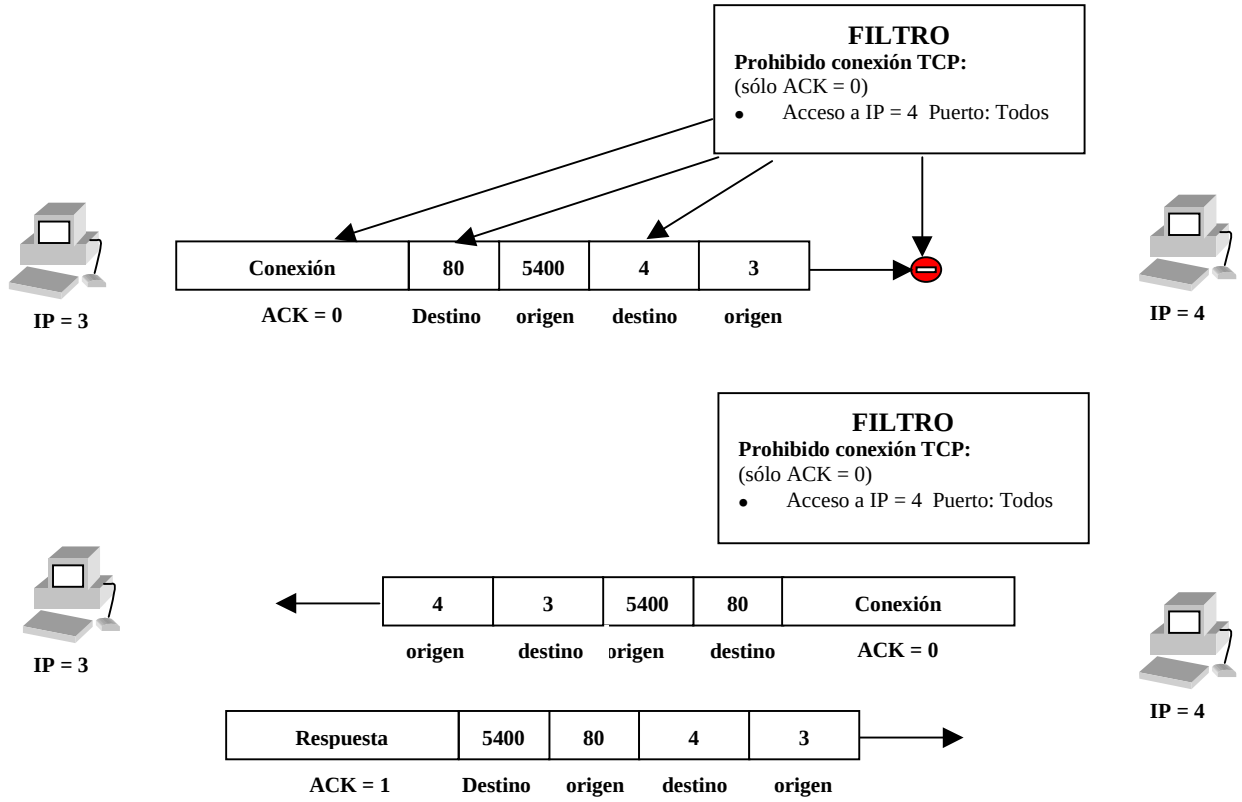

Figura 2.5.3: Direccionalidad con Firewall.

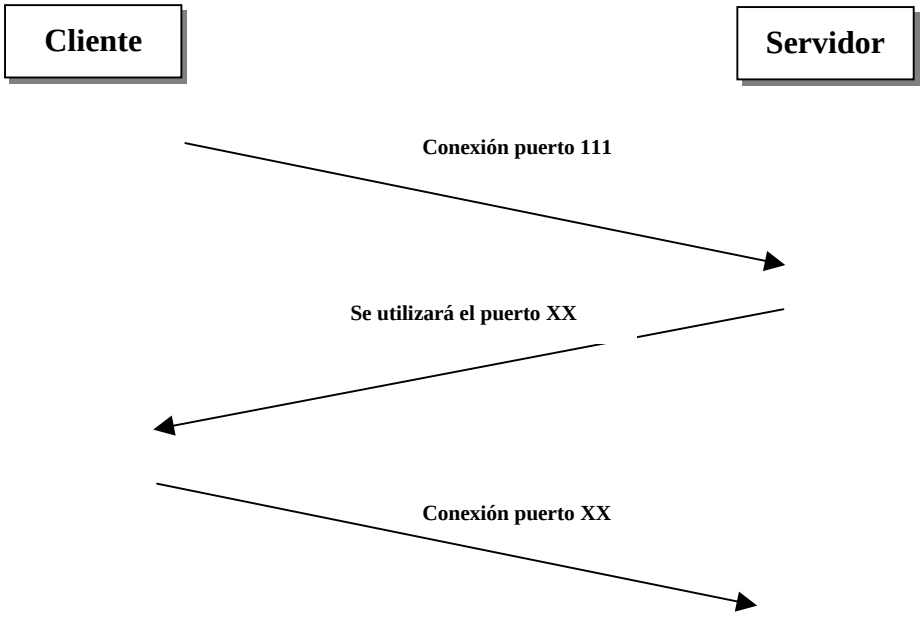

Figura 2.5.4: Conexión de RPC

El cliente y el servidor no podrán comunicarse por el nuevo puerto asignado dinámicamente ya que estará dentro de los prohibidos, no será ni el 80 ni el 111 (ver Figura 2.5.4.)

Para evitar esto los **Firewalls** examinan los paquetes de estos servicios y **detectan los puertos asignados dinámicamente** y así lo abren durante esa conexión.

2.6. Ataques al control por IP o nombre

Existen diversos ataques al control de accesos por IP o nombre. Algunos se saltan el control para acceder a la información restringida y otros están fuera del alcance de los filtros. Se pueden agrupar en:

- **Spoofing**. Consiste en cambiar la dirección origen por una que es aceptada por el filtro.
- **Hijacking**. Consiste en secuestrar una sesión, es decir, introducirse en la comunicación aprovechando una sesión que ha abierto un usuario con privilegios. Se deben enviar los mensajes con la IP del usuario que abrió la sesión y recibir las respuestas del servidor antes que el usuario legal.
- **Denegación de servicio (DoS) con paquetes UDP o ICMP.** Se aprovecha el control débil que se realiza sobre los paquetes no orientados a conexión para realizar ataques para destruir, no para obtener información.
- **Tunneling.** Se aprovecha un ordenador que está detrás del filtro o tiene permisos de acceso para utilizarlo de plataforma. El ordenador externo recibe una conexión del interno y a partir de ésta realiza los ataques. También se puede hacer utilizando una conexión permitida al ordenador interno y desde ésta pasar a un software capaz de atacar. Para ello se necesita la colaboración de algún usuario interno, poder instalar un caballo de Troya que abra un camino o utilizar un error (bug) de un programa inocente. (Ver Figura 2.6.1)
- **Ataques al DNS.** Modificar las memorias cachés del DNS falsificando las relaciones IP/nombre, así cuando el filtro pregunta a qué nombre pertenece una IP que pide permiso de entrada se consigue que el DNS conteste un nombre autorizado.

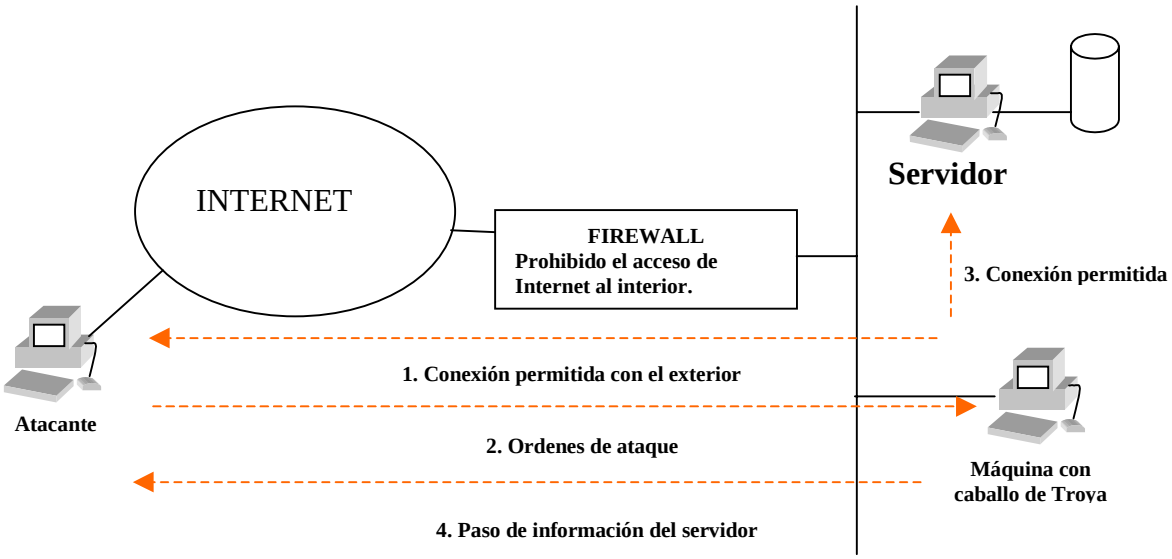

Figura 2.6.1: Ataque con técnicas de tunneling

3. Control de accesos de usuario

3.1. Características generales

Para realizar la selección de usuarios se debe hacer una identificación única del usuario o grupo de usuarios, debe ser independiente de la máquina utilizada y el sistema de telecomunicación. Hoy en día se conocen tres únicos métodos para identificar personas, son:

- Por las características **físicas**: **biométricos**.
- Por un **secreto** compartido: **contraseñas** (*Passwords*).
- Por la posesión de un **objeto** (software o hardware): **Tokens o certificados digitales**.

Los sistemas **más utilizados** actualmente son de **contraseña**, con diferentes variantes se aplican a casi todos los aspectos de la seguridad de la información. Los sistemas **biométricos** son mucho **más nuevos** pero se están desarrollando a gran velocidad, se espera que en pocos años se incorporen a muchos aspectos de la seguridad, aunque tienen condicionantes que retardan su desarrollo como: el precio de los equipos de captación, conceptos éticos, poca costumbre de utilización, etc...

Los sistemas de **posesión de un objeto** son los más antiguos en control de accesos físicos, la llave de las puertas o los sellos de los reyes son tan antiguos como el concepto de acceso o identificación de derechos. Pero **en el mundo digital se utilizan muy poco** para el acceso a sistemas de información, probablemente por el gasto extra que supone un identificador de objetos. Actualmente se están **desarrollando** mucho los accesos por sistemas criptográficos llamados **certificados digitales**.

Igualmente todos **los sistemas se pueden combinar para aumentar la seguridad**, especialmente el uso de contraseñas normalmente acompaña a los sistemas biométricos y los de objetos. No será extraño en el futuro tener que introducir una contraseña, una tarjeta inteligente y la huella dactilar para acceder a la información.

El control de accesos por usuarios también se puede clasificar por la **ubicación del filtro**, así puede estar en:

- **Servidor**. Permite control de acceso remoto y local.
- **Filtro de la red**. Sólo controla accesos remotos.

Por último, el control de accesos por usuarios se puede clasificar atendiendo a quien organiza este control. Así existen **tres tipos de organización**:

- **DAC (Discretionary Access Controls)**

El creador del fichero define los permisos de los objetos (ficheros, recursos, etc...). Desde la administración del sistema se pueden crear grupos de usuarios, usuarios genéricos y varios tipos de facilidades para que el creador del fichero pueda asignar los

permisos. Es el control más habitual en los sistemas operativos: Windows de Microsoft, UNIX, etc... Es muy vulnerable a los caballos de Troya.

- **MAC (Mandatory Access Controls)**

La administración del sistema operativo asigna los permisos a los objetos. El sistema operativo crea un número de etiquetas (secreta, confidencial, no calificada, dpt. comercial, etc...) con unos derechos de acceso asignados y cada objeto tiene su etiquetas. Los usuarios se agrupan en sujetos que tienen unos permisos definidos para cada etiqueta. Una protección buena contra caballos de Troya es hacer que cada nivel pueda escribir a los ficheros de su nivel o superior y leer en los de su nivel o inferiores.

- **RBAC (Role-Based Access Controls)**

Intenta tener las ventajas de los anteriores sistemas y evitar la rigidez del MAC y la inseguridad del DAC. El funcionamiento por roles se acerca más a la distribución de trabajos real de las empresas. **Los roles son funciones concretas** que realiza un usuario dentro de la empresa durante un tiempo determinado, así a los usuarios se les asigna unos roles y cada rol tiene unos permisos sobre los objetos.

3.2. Control por contraseñas

Las contraseñas son **un punto débil de los sistemas de seguridad**, pero para realizar control de acceso por usuario son el sistema **más sencillo, popular y probado**. Se puede hacer un símil con las protecciones físicas de los edificios, la puerta y su sistema de abertura (llaves, combinaciones, la cerradura,...) son imprescindibles pero también son el principal método utilizado para acceder sin permiso.

En los sistemas operativos y las aplicaciones con filtro las contraseñas se deben **guardar encriptadas en ficheros**. El problema es que estos ficheros no pueden tener permisos de usuarios restringidos ya que al entrar la contraseña el usuario puede ser cualquiera. Una forma de evitar este problema sería dar permiso de administrador al fichero y que el usuario por defecto cuando se introdujera la contraseña fuera el administrador, pero esto sería muy peligroso porque cualquiera tendría permiso de administrador por un momento.

Así este fichero sin permisos en principio es accesible por todos los usuarios, pero se utilizan técnicas para evitar este acceso. Un ejemplo: en Windows de Microsoft el fichero se está utilizando siempre por el sistema y los ficheros que utiliza el sistema no son accesibles para escritura, esta protección ya ha sido vencida por los programas de los atacantes.

Si los ficheros son accesibles, el atacante únicamente necesita desencriptar las contraseñas. Para hacer difícil esta tareas se utilizan sistemas de encriptación **irreversibles** y, además, el **descubrimiento de una contraseña no da pistas sobre las otras**. Dos ejemplos de encriptación son los siguientes:

- **UNIX**

El fichero guarda: el nombre del usuario, la contraseña encriptada, información necesaria para el shell utilizado por el usuario. La contraseña se encripta con el algoritmo crypt(3) o, en algunos UNIX concretos, crypt(16).

El crypt realiza la siguiente función:

$E_{contraseña}[E_{contraseña}[.....[E_{contraseña}[0].....]]$

La función $E_{contraseña}$ se realiza 25 veces.

$E_{contraseña}$ es realizar un algoritmo **DES modificado** por dos números denominados **SALT.** El SALT son dos números calculados aleatoriamente que se graban en el fichero con la contraseña, estos números hacen que la función no sea exactamente un DES. Los motivos de usar el SALT son:

- **No se puede utilizar un circuito electrónico** que implemente el DES para ir más rápido. Esta medida era efectiva cuando se inventó el sistema, actualmente no tiene sentido debido a la gran velocidad del software.
- **La misma contraseña** encriptada en dos máquinas tiene **un resultado diferente**. Así se dificulta la identificación automática de contraseñas descubiertas anteriormente.

La **clave del algoritmo es la contraseña** y siempre se encripta un 0 binario. El proceso de **utilizar la contraseña como clave** tiene dos **ventajas**:

- El proceso es **irreversible** porque el DES no permite encontrar una clave a partir del texto y el criptograma. Así la identificación se realiza volviendo a encriptar con la contraseña y comparando, el mismo proceso que usan los atacantes cuando prueban diferentes contraseñas.
- El conocimiento de una contraseña y su encriptación **no da información para descubrir las otras**.

Las contraseñas de UNIX tienen una **longitud máxima de 8 caracteres** que es la clave del DES. Algunos **sistemas especiales permiten frases como contraseña** (**passphrase**) y realizan una función Hash de la frase para convertirla en una palabra de 8 carácteres. Esto refuerza la seguridad frente a los ataques de *prueba y ensayo*.

- **Windows NT**

Los servidores de Windows NT permiten dos formas de codificar las contraseñas, la forma propia y la de LANManager, esta última sólo se utiliza para compatibilidad con las redes de este tipo. Aquí se trata la propia de Windows NT.

En Windows NT se ha buscado dar **más velocidad al proceso** a costa de utilizar **criptología débil**. Como los atacantes no intentan romper el algoritmo de encriptación sino que lo utilizan para probar contraseñas, este sistema no basa su seguridad en la fortaleza del algoritmo, cosa que es discutible. Se consigue una velocidad mucho más alta que en UNIX, esto proporciona comodidad al usuario pero también facilita el trabajo del atacante.

El sistema es:

Hash[Hash[contraseña]]

La función Hash utilizada fue en principio MD4, actualmente no se sabe si utiliza otra. También cumple **las propiedades** de ser:

- **Irreversible** porque las funciones Hash siempre son irreversibles.
- El conocimiento de una contraseña y su encriptación **no da información para descubrir las otras**.

Permite frases largas como contraseña (Passphrase) ya que las funciones Hash resumen textos de cualquier longitud variable.

3.3. Ataques a contraseñas

Las contraseñas son un punto **muy vulnerable** de la seguridad del sistema de información, si el atacante consigue esa secuencia de pocos caracteres que forma la contraseña tiene la puerta abierta a atacar cualquier recurso. Las formas de poder descubrir las contraseñas de los usuarios se pueden agrupar en:

- **Con acceso al fichero**.

Si se tiene acceso al fichero de contraseñas **adivinarlas es sólo cuestión de tiempo**. Para ello se utilizan programas denominados *Crackers* que prueban todas las posibilidades hasta encontrar una que al encriptarse coincide. Hay dos métodos de elegir las posibles palabras:

- **Diccionario**. Prueban todas las palabras que pueden aparecer en una enciclopedia, o sea, nombres comunes (de un diccionario), nombres de persona, de animal, geográficos, fechas, números, etc... Esto se puede hacer consecutivamente para varios idiomas y, además, ir haciendo pasadas intercalando números y signos de puntuación. Para que una contraseña sea fácilmente recordable debe ser inteligible para el usuario, por lo tanto, ser alguna palabra con significado. Pero este hecho reduce mucho el número de posibilidades, con 8 caracteres se pueden formar $128^8 = 7{,}2*10^{16}$ palabras mientras en las enciclopedias hay sólo unos centenares de miles de palabras.
- **Prueba y ensayo (*Task force*).** Se prueban todas las combinaciones de letras, números y signos posibles. Este método es mucho más lento que el anterior pero al final siempre da resultado (puede tardar meses). Normalmente se va aumentando el número de caracteres de forma progresiva, así se encuentran primero las contraseñas más cortas.

- **Caballos de Troya**.

Se **sustituyen programas útiles por aplicaciones preparadas por el atacante** que tienen el mismo nombre. Los ejecuta el propio usuario pensado que son un programa y realizan funciones de observación, modificación o destrucción de la información. Los caballos de Troya sirven para muchos tipos de ataques, uno concreto es la captura de

contraseñas. Se puede hacer sustituyendo uno de los programas que tratan las contraseñas en claro, capturando el teclado o capturando las transmisiones por la red si se envía en claro.

- **Espías de la red**.

Si se instala en una máquina un programa llamado **sniffer**, éste **captura toda la información que circula por la Ethernet o Token Ring de la máquina**. Estos programas descubren las contraseñas mientras circulan por la red. Si no están encriptadas (hay muchos sistemas que no encriptan las contraseñas para enviarlas), el atacante ya ha conseguido su medio de acceso. Pero si están encriptadas también los puede utilizar repitiendo el mensaje como respuesta a una petición de identificación. El atacante únicamente necesita poder instalar en el servidor o en una máquina de la misma LAN un programa de este tipo.

- **Ingeniería social**.

Uno de los sistemas más utilizados es el llamado por los atacantes ingeniería social, **no es técnico** sino que se basa en descubrir las contraseñas **directamente de los usuarios**. Los métodos pueden ser: observar el teclado cuando se introduce la contraseña, descubrirlo escrito en un papel, pedirlo por correo electrónico o teléfono haciéndose pasar por el administrador, etc... Aunque parezca imposible, las estadísticas dicen que es uno de los sistemas más utilizados.

- **Otros sistemas.**

Hay otros sistemas no tan generales para obtener la contraseña, como aprovechar errores (bugs) de los programas o sistemas operativos.

3.4. Defensas a ataques a contraseñas

Para defenderse de estos ataques se puede trabajar en **tres líneas**:

- **Políticas de personal**.
- **Herramientas de programas**.
- **Sistemas de contraseña de un uso**.

Las **políticas de personal** van orientadas a aconsejar u obligar al personal de la empresa a cumplir ciertas normas para proteger sus propias contraseñas. Tanto los ataques con acceso al fichero como los de ingeniería social se basan en aprovechar que los usuarios no tienen cuidado con la elección y el mantenimiento de sus contraseñas. Así una política puede fijar **normas** como:

- Tamaño mínimo.
- Intercalar entre las letras números y signos de puntuación.
- Prohibir passwords de diccionario.
- Cambiarlo cada cierto tiempo.
- Si un atacante entra utilizando el password de un usuario, sancionarlo.

- Etc...

Las **herramientas** pueden ser opciones del sistema operativo, programas complementarios al sistema o programas de inspección. Los objetivos son:

- Obligar por software a **cumplir las políticas de personal** comentadas en el anterior párrafo.
- Atacar con un Cracker u otro programa para **probar la resistencia del sistema** de contraseñas.
- **Cancelar cuentas** que han recibido intentos de acceso fallidos. Se recuperan después de un tiempo o a través del administrador.

Una manera de aumentar mucho la seguridad en los accesos remotos es utilizar unos **sistemas**, llamados **OTP (One-Time Password)**, **donde la contraseña de un usuario cambia cada vez que se usa, o sea, contraseñas de un uso**. El origen es el sistema S/Key propietario de la empresa Bellcore, pero actualmente el IETF ya ha estandarizado el método con el nombre de OTP. El servidor y el usuario deben estar sincronizados para saber en cada momento que contraseña se debe utilizar. Si algún atacante descubre una contraseña no le sirve porque para el siguiente acceso se necesita otra.

Los sistemas OTP necesitan **servidores preparados** para calcular cada vez la contraseña que toca y clientes con **un software o un equipo electrónico** capaz de realizar la misma función. Estos equipos electrónicos se llaman testigos (Tokens) y se pueden considerar de la familia de control de accesos por posesión de un objeto (ver*: 3.6Acceso con objetos físicos: Tokens*) combinado con contraseñas.

En OTP para calcular la contraseña se utilizan los siguientes **parámetros**:

- Una **frase secreta del usuario** (Passphrase).
- Una **palabra aleatoria** conocida por el servidor y el software o hardware del usuario.
- Una función **Hash.**
- El número de accesos que se han realizado desde el inicio, o sea, el **número de secuencia**.

Así se entra a una función Hash la passphrase y la palabra aleatoria, al resultado se le aplica varias veces la misma función Hash según marca el número de secuencia. El resultado se envía al servidor como contraseña, éste realiza el mismo proceso y se comparan los resultados (Ver Figura 3.4.1: Algoritmo de una OTP).

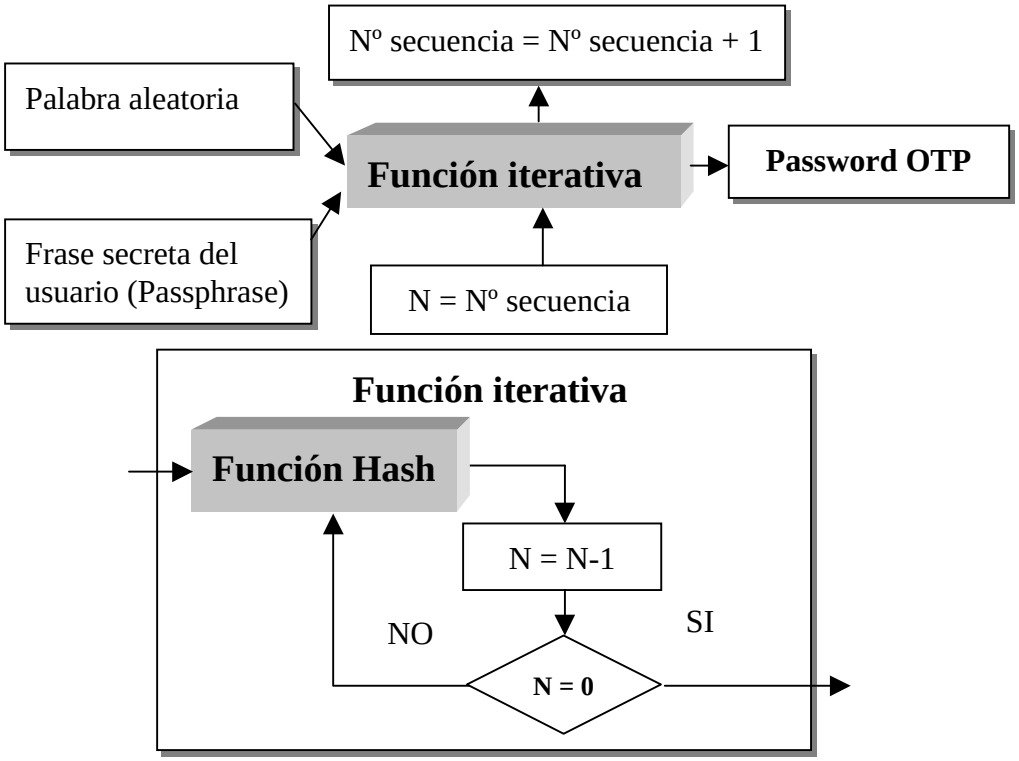

Figura 3.4.1: Algoritmo de una OTP

3.5. Sistemas biométricos

Estos sistemas utilizan una **característica física del usuario** (**autentificadores**). La característica debe ser única en las personas y no cambiar con las circunstancias (estado de ánimo, temperatura ambiente, iluminación, etc...) ni con el tiempo (insensible al envejecimiento). Estos sistemas son mucho más seguros que los de contraseña sobre todo si se combinan con otros, como **ventajas** tienen:

- **Intransferibles**. El atacante no los puede utilizar aunque los conozca. Esta característica es suficiente para considerar el sistema **mejor que los de contraseña o posesión de objetos**.
- **No necesitan gestión** del usuario, como cambiarlos a menudo, recordar frases largas, guardar objetos (Tokens), etc...
- Sirven tanto para accesos **físicos como lógicos**.
- Son **muy seguros** a cualquier ataque.

Actualmente aun tienen las **desventajas**:

- Necesitan **electrónica adicional** para realizar las lecturas de imágenes y, por lo tanto, son más caros.
- La tecnología **no está muy avanzada**.
- Tienen un cierto **rechazo del usuario** delante de la exposición física a un sensor.

- Hay algún **prejuicio moral** porque las características físicas de las personas son invariables y hacerlas públicas implica estar fichado para toda la vida.
- **No son exactos**.

La mayoría de estas **desventajas se corregirán con el tiempo**.

En una identificación biométrica se realizan las siguientes fases:

- **Captar** la imagen o sonido relativa al autentificador de la persona mediante un sensor.
- **Modificar** los datos brutos de la imagen o sonido mediante técnicas de tratamiento de señal para extraer los parámetros básicos y únicos del usuario (modelos/patrones), así como eliminar los datos dependientes de las condiciones externas de la medida.
- **Comparar** estos parámetros con los almacenados.

Como se puede deducir del proceso, la comparación de resultados **nunca es exacta**, por lo tanto se busca un grado de aproximación a partir del cual se considera que los parámetros medidos son de la misma persona que los almacenados. Así es posible tener errores, éstos están medidos estadísticamente para cada método biométrico con los siguientes índices:

- **FAR** (False Acceptence Rate). Mide en tanto por ciento la relación de identificaciones erróneas consideradas correctas.
- **FRR** (False Reject Rate). Mide en tanto por ciento la relación de rechazos al acceso que eran correctos.
- **SR** (Succes Rate). Da un índice global de la calidad del sistema, relacionando los índices anteriores, se utiliza la fórmula:

SR = 100 - (FAR + FRR)

En el proceso de **comparación** se pueden diferencian **dos métodos: identificación y verificación**. La **identificación** consiste en encontrar en una base de datos de parámetros biométricos si los medidos coinciden aproximadamente con algún usuario, es para un sistema de acceso donde no se introduce el nombre de usuario o para búsqueda de personas (por ejemplo en archivos policiales). La **verificación** compara directamente los parámetros medidos con los del usuario y según la aproximación matemática se considera el acceso permitido o denegado, es el sistema de acceso más habitual. Lógicamente .la verificación tiene índices de FAR y FRR mucho más elevados que la identificación.

Los sistemas biométricos actuales se basan en **medidas de**:

- **Emisión de calor**.

Se mide la emisión de calor del cuerpo o termograma, realizando un mapa de valores sobre la forma de la persona. Permite medidas sin contacto, o sea, a distancia.

- **Huella digital**.

Aprovecha las características diferentes entre todas las huellas digitales de los humanos. Necesita un escaner de dedos, un equipo bastante barato. Su FAR es de 0'001 % y su FRR de 0,001 (cortesía de Veriprint).

- **Mano**.

Es fácil de implementar y tiene un coste bajo. El problema es que varia mucho con el tiempo y las condiciones físicas de la persona. Los patrones se deben renovar de vez en cuanto.

- **Caras**.

Debe medir características únicas e invariables con el tiempo y las expresiones de las caras, como la distancia entre los ojos, de la boca a la nariz, etc...

- **Iris**.

El iris de los ojos presenta multitud de líneas concéntricas que son diferentes en todos los humanos. Un inconveniente es el rechazo social a colocar el ojo delante de un escaner. Es un sistema lento porque maneja muchos datos pero tiene mucha exactitud, el FAR es del 0,0006 % y el FRR del 0,007 %.

- **Retina**.

Este sistema tiene un FAR de 0 pero un FRR del 12 %, por lo tanto se puede utilizar para sistemas donde es muy importante evitar el acceso de atacantes pero no es muy molesto el rechazo de usuarios autorizados. También tiene el mismo rechazo social del sistema de iris.

- **Voz**.

Se graba la dicción de una frase, siempre la misma, por el usuario y en los accesos se compara la voz. Es muy sensible a factores externos como el ruido de fondo, el estado de ánimo o el envejecimiento pero tiene la ventaja de no necesitar contacto y utilizar sensores muy baratos y habituales en los ordenadores (micrófonos). Para acceso físico en lugares públicos tiene rechazo social. Es el único con posibilidad de transferirse ya que los atacantes pueden hacer una grabación externa sin ser vistos. Una ventaja es la posibilidad de verificación telefónica.

3.6. Acceso con objetos físicos: Tokens

Los Tokens son objetos utilizados para el control de accesos de usuario. Pueden ser:

- **Memorias**. Guardan una palabra clave, contraseña. La ventaja es poder utilizar contraseñas aleatorias sin necesidad de recordarlas.
- **Inteligentes**. Son equipos electrónicos que realizan un algoritmo donde se crean contraseñas de un uso (OTP) o se genera un protocolo entre el servidor y el token (certificados).

Pueden estar **contenidos en**:

- **Tarjetas magnéticas**. Sólo permiten memoria, se necesita un lector magnético.
- **Tarjetas chip**. Tienen un procesador interno que permite inteligencia. Se necesitan lectores especiales.
- **Memorias EPROM o Flash**. Se introducen en llaveros o otros objetos pequeños y permiten almacenar contraseñas sin inteligencia.
- **Calculadoras**. Son pequeños ordenadores que permiten inteligencia. Se comunican con el usuario mediante teclados, displays y/o conexiones serie al ordenador.

Estos sistemas **complementan otro sistema de acceso: contraseñas, biométricos o certificados digitales**. Así **su función es reforzar los otros**, por lo tanto, aumentan mucho la seguridad porque añaden el factor de posesión de un objeto.

El problema puede ser el robo o la pérdida del Token, para solucionar esto **se deben combinar con la entrada de una contraseña o una medida biométrica**. Probablemente en el futuro casi todos los sistemas necesitarán un Token, una contraseña y una medida biométrica.

3.7. Acceso con certificados digitales

Este sistema **utiliza criptología** para dar un objeto lógico, no físico, a los usuarios con permisos. Está **protegido contra robo, pérdidas y repetición de mensajes** porque el proceso de acceso incluye un protocolo de validación.

Un usuario autorizado **debe tener**:

- Una **clave privada** de algún algoritmo asimétrico.
- Un **certificado digital** con la clave pública pareja de la privada y firmado digitalmente por el servidor.

Los algoritmos asimétricos funcionan con dos claves, con una se encripta y con la otra se puede desencriptar, no se puede encriptar y desencriptar con la misma (Ver Figura 3.7.1). Así una clave es privada y únicamente la tiene el usuario, su descubrimiento rompe todo el sistema de seguridad, la otra se transmite antes de la conexión mediante un certificado digital.

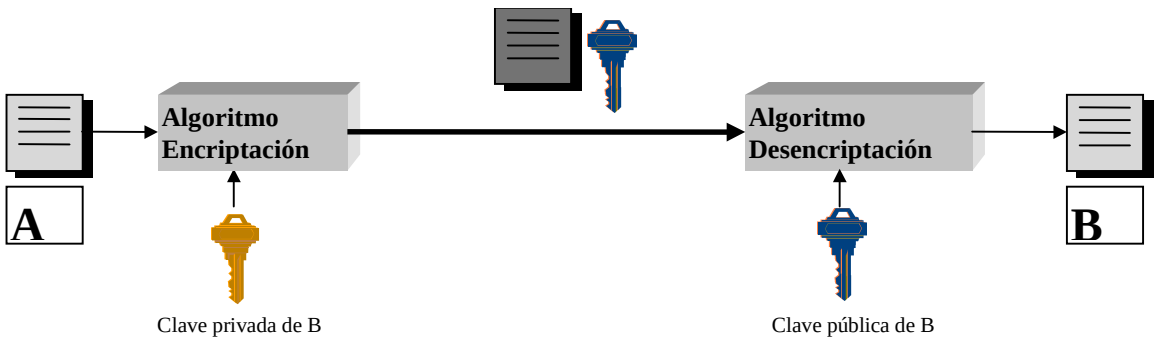

Figura 3.7.1: Algoritmos asimétricos

El certificado digital es un objeto lógico (código) que **contiene**:

- **Nombre y datos del usuario**.
- La **clave pública** del usuario.
- **Datos e informaciones generales**.
- **La firma digital de una tercera persona**.

Esta tercera persona asegura que la clave pública es de quien dice ser. Así la seguridad se basa en la corrección de la firma digital de la tercera persona.

Una **firma digital** se realiza haciendo el resumen del texto y encriptandolo con la clave privada de firmante. Así al desencriptarlo con la pública y comparando con el resumen otra vez calculado **puede comprobarse** que:

- El **texto no ha sido modificado** porque los resúmenes coinciden.
- La **firma es de la persona que tiene la clave privada** pareja de la pública utilizada para desencriptar.

El sistema de acceso con certificados digitales se basa en las siguientes **fases** (ver Figura 3.7.2):

- El usuario autorizado **ha recibido un certificado digital** con su nombre y su clave pública **firmado por el servidor** donde quiere acceder. También ha recibido de manera secreta **la clave privada**.
- Para acceder **envía su certificado**.
- **El servidor comprueba la firma** del certificado y guarda la clave pública.
- El servidor envía un **número aleatorio**.
- El usuario **encripta el número aleatorio con su clave privada** y envía el resultado.
- El servidor **desencripta y comprueba que la clave privada** es pareja de la pública que ha llegado con el certificado.

El proceso puede complicarse pero siempre se debe basar en los mismos **principios**:

- La **posesión del certificado digital correctamente firmado implica** que este usuario **tiene la clave privada pareja de la pública indicada** y la ha recibido del servidor.
- La posibilidad de **encriptar con la clave privada indica** que la persona que ha enviado el certificado **es quien dice ser**. Se evita los ataques de personas que han robado por la red el certificado.

Un problema es: **¿cómo dar de baja usuarios?**. Para esto se utilizan:

- Todos los certificados tienen **fecha de caducidad**.
- **Listas de revocación de certificados (CRL)**. Si se quiere dar de baja un certificado y no está caducado se añade a la CRL hasta que caduque.

Los certificados son como el carnet de identidad de las personas, por lo tanto implican **mucha gestión del servidor**. Así si el servidor quiere activar un grupo de usuarios con los mismos permisos no puede utilizar un único certificado sino que debe crear uno para cada usuario. Además también se deben gestionar las bajas y la entrega de claves privadas de una manera segura. Además **para un usuario con varios accesos** también puede representar **una complicación tener que gestionar diversas claves privadas y certificados**. La solución a estos problemas es un sistema nuevo llamado **certificados de atributos**.

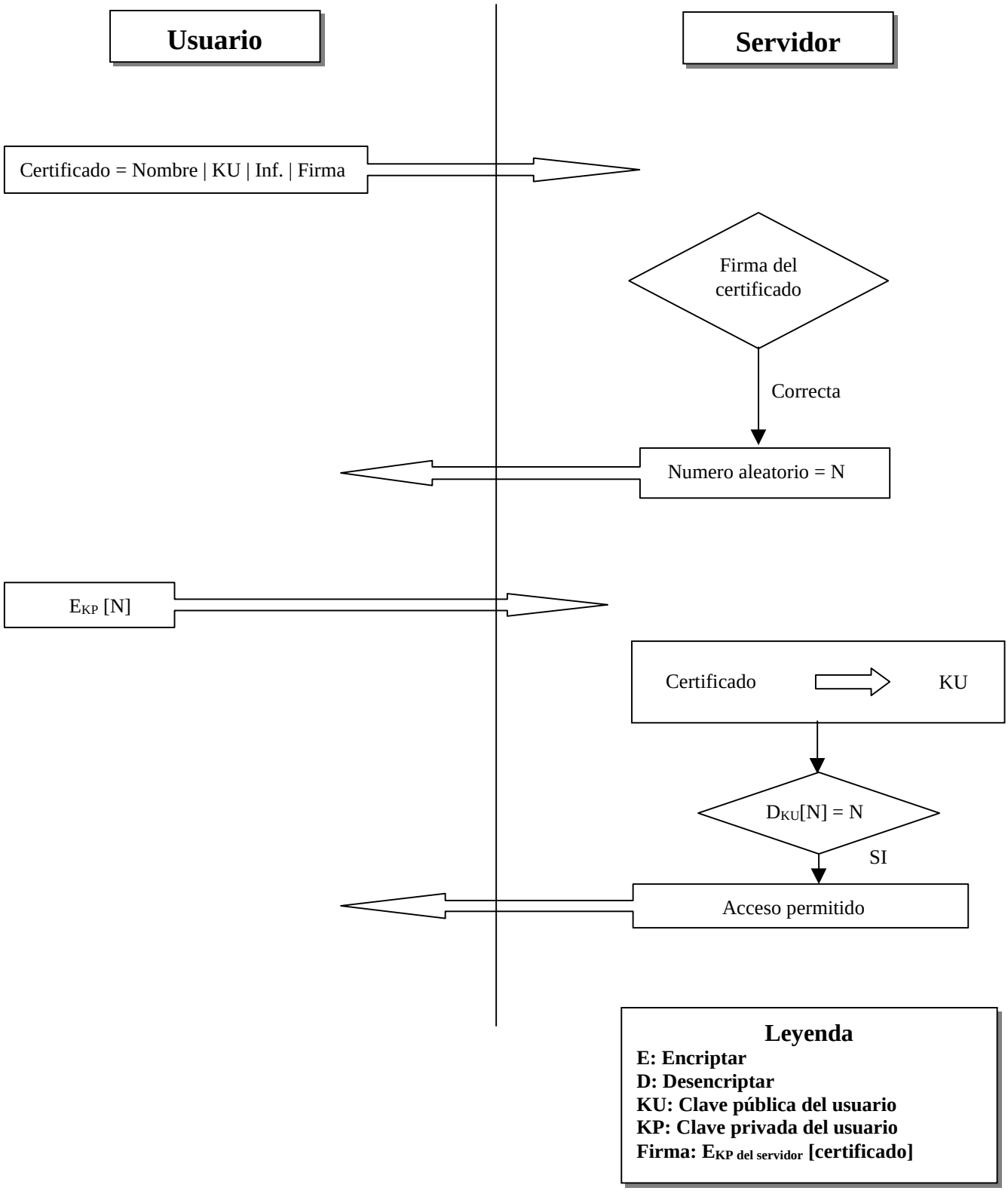

Figura 3.7.2: Control de accesos con certificados digitales

Los certificados de atributos añaden una filosofía nueva universal para la seguridad y los controles de acceso. **El certificado individual** de una persona física o jurídica **debe ser como el carnet de identidad**, se debe asignar por una entidad que ofrece confianza a todo el mundo (como la policía que emite los carnets de identidad), este certificado no sirve para acceder pero si para identificar al usuario delante de cualquiera. El formato del certificado individual deberá ser estándar para todo el mundo y las entidades que los emiten y firman reconocidas por todo el mundo. **Para aplicaciones concretas**, como el control de accesos, **se utilizan certificados de atributos** que tienen las siguientes características:

- Explican **atributos concretos de la persona física o jurídica**. Por ejemplo: pertenecer a una empresa, tener una nacionalidad, no estar fichado, ser solvente, estar de alta en el acceso a una Web, formar parte de un grupo con permisos de acceso, etc... Estos atributos son los que interesan para acceder a recursos.
- Tienen una **duración muy corta** y se han de estar renovando continuamente. Así se evita la gestión de las CRL.
- Tienen **formato libre** y pueden ser expedidos por cualquiera.
- Siempre **se entregan con el certificado personal** que avala la persona propietaria de los atributos. Es como presentar a la entrada de un club un carnet de socio (sin fotografía) y el carnet de identidad para asegurar la identidad persona.

Así a las personas con acceso se les daría un certificado de atributos después de presentar el personal. Los usuarios gestionarían un certificado personal intransferible y diversos certificados de atributos para cada aplicación concreta. Esto es un **proyecto de futuro** y todavía está en fase embrionaria.

Por último queda el problema de **¿cómo transportar el certificado?**. Si siempre se accede desde la misma máquina se puede grabar en el disco, pero en control de accesos por usuario siempre se intenta dejar al usuario libertad de máquina. La solución es **utilizar Tokens (en concreto tarjetas chip) donde se almacenan los certificados y se implementa el protocolo**.

4. Autenticación Kerberos

4.1. Introducción

Kerberos es un sistema de control de accesos y autenticación completo inventado por el M.I.T. Las primeras versiones se realizaron para el sistema operativo UNIX pero actualmente se están creando nuevas versiones para otros sistemas operativos.

Sus objetivos son:

- **Exigir autenticación a los usuarios para la utilización del sistema y en particular para cada servicio ofrecido.**
- **Exigir autenticación a los servicios** (software de los servidores).

Este sistema identifica usuario y servicios como objetos, por lo tanto, es independiente de las máquinas y su ubicación física**. Es muy eficiente para conexiones remotas a servicios de uso restringido y permite centralizar la gestión de accesos.**

Existen dos versiones:

- Versión 4. Más utilizada.
- Versión 5. Corrige problemas de seguridad encontrados en la anterior, su estándar es el RFC1510.

4.2. Características

- Utiliza únicamente **clave simétrica**.
- **Los passwords nunca viajan por la red**.
- Se utiliza **control de accesos** individualizado para **cada servicio**, pero sólo **se introduce el password una vez por sesión**.
- Se puede separar la red en **diferentes dominios** físicos de seguridad.
- Basa el control de accesos en un sistema (hardware y software), llamado **Servidor de Autenticación AS**, diferente de los servidores de información.
- En la versión 4 utiliza el algoritmo DES, en la 5 permite **cualquier algoritmo y cualquier longitud de clave**.

4.3. Funcionamiento

Intervienen los siguientes elementos:

- **Usuario.**
- **Servidor de servicios.**

- **Servidor de autenticación (SA).**
- **Servidor de concesión de tickets (TGS).**

Aunque estos dos últimos pueden estar físicamente en la misma máquina.

Para que los passwords no viajen por la red se utilizan **tickets** para validar el acceso a los servicios. Estos tickets deben estar en posesión del usuario y enviarse a los servidores para conseguir el acceso. Un ticket es información encriptada con un passwords del sistema que permite el acceso al usuario que lo posee. Siempre tienen una fecha de caducidad para que no puedan ser aprovechados por los espías de la red. Tampoco se guardan passwords ni tickets en las máquinas de los usuarios para evitar a los Hackers que tienen accesos a estas máquinas.

El proceso de autenticación se divide en dos fases y 6 mensajes. La Figura 4.3.1 muestra estas fases.

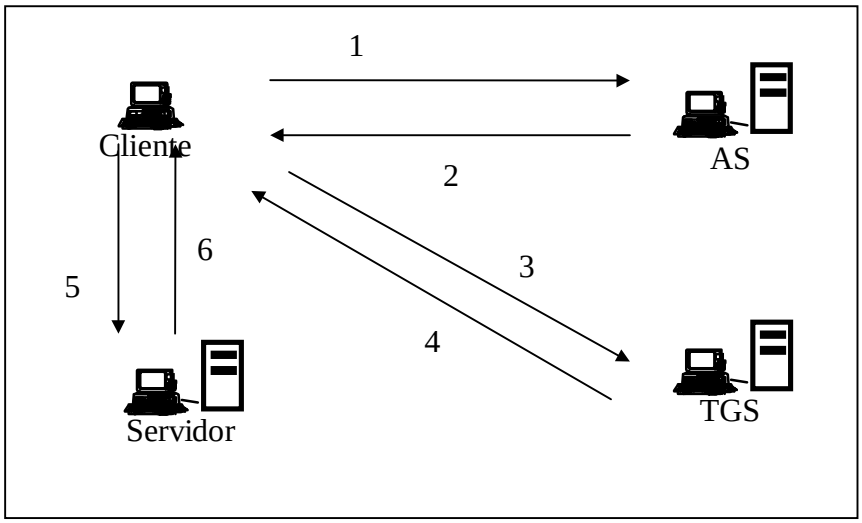

Figura 4.3.1: Proceso de autenticación Kerberos

Las fases son las siguientes:

1. Autenticación de usuario. Mensajes 1 y 2.

2. Autenticación de servicio. Mensajes 3, 4, 5 y 6.

Al conectarse se debe realizar la autenticación del usuario al sistema Kerberos y después se pueden hacer tantas de servicios como se necesiten conectar, pero sin necesidad de repetir la de usuario.

4.4. Autenticación de usuario

Se autentica el usuario al sistema, **es la única fase del proceso donde se introduce el password**. Esta autenticación sirve para posteriormente acceder al TGS, que concede tickets para los servicios.

El resultado final es la posesión del **$TICKET_{TGS}$**. Con este ticket se puede pedir autorización en el TGS a tantos servicios como se necesite.

La seguridad se basa en siguientes claves simétricas:

- El password del usuario genera (con un proceso matemático) una clave para encriptar el mensaje 1. El AS posee la misma clave y con ella comprueba la autenticidad del usuario.
- El $TICKET_{TGS}$ está encriptado con una clave conocida solamente por el TGS y el AS. Por lo tanto, el usuario no puede generar ni modificar un ticket de este tipo.
- Una clave de sesión generada aleatoriamente para las transmisiones entre TGS y usuario. Se le envía al usuario encriptada con la del password y al TGS dentro del $TICKET_{TGS}$. Ninguna de las partes la puede modificar.

El resultado final es el Ticket para el usuario, que no se puede modificar y tiene una fecha de caducidad, y posteriormente el TGS lo reconocerá como auténtico. También se recibe la clave de sesión a utilizar con el TGS. Si alguien captura el Ticket en la línea no lo puede utilizar con el TGS ya que no conoce su clave de sesión, está viene para el usuario encriptada con el password.

Al acabar esta fase se destruye el password de usuario. El $TICKET_{TGS}$ se puede utilizar para pedir autorización a varios servicios mientras no caduque, sin necesidad de volver a acceder al AS ni introducir el password.

4.5. Autenticación de servicios.

El usuario pide al TGS el **$TICKET_{SERVICIO\ X}$** para autenticarse delante del servicio X, también para comprobar la identidad de éste. Este proceso se realiza tantas veces como servicios distintos quiera utilizar el usuario, pero nunca vuelve a introducir el password.

La seguridad se basa en siguientes claves simétricas:

- La clave de sesión mencionada en el capítulo 3.2.
- El $TICKET_{SERVICIO\ X}$ está encriptado con una clave conocida por el servicio X y el TGS.
- Una clave de sesión para utilizar en las comunicaciones entre el servicio y el usuario. La conoce el usuario porque llega encriptada con la clave de sesión actual y el servicio porque está en el Ticket. Ninguna de las partes la puede modificar.

El resultado es la obtención del $TICKET_{SERVICO\ X}$ y una clave para la sesión con el servicio. Si alguien captura el Ticket en la línea no lo puede utilizar ya que desconoce la clave de sesión.

El servicio se identifica utilizando la clave de sesión. Si es falso no podrá desencriptar el Ticket y, por lo tanto, no tendrá la clave de sesión.

4.6. Instalación de Kerberos.

El software se puede conseguir gratis por Internet. Estas versiones no disponen de servicio técnico, actualizaciones, instalación, ni formación, por lo tanto, sólo es aconsejable para pequeñas empresas. **Existen muchas versiones comerciales** de Kerberos con soporte técnico y formación a precios muy razonables.

A parte de comprar el software de cliente, el AS y el TGS **se deben adaptar las aplicaciones cliente/servidor (los servicios) al entorno Kerberos. Esto puede suponer un aumento considerable de los costes de la instalación** de este sistema y es muy importante tenerlo en cuenta. Existen herramientas de programación para Kerberos, como GSS-API, útiles para adaptar el software y mantenerlo actualizado a los cambios en sistemas de autenticación.

Sobre el papel, este sistema **es el mejor entorno actual para seguridad de accesos y autenticación**. Pero antes de instalarlo en una empresa se debe tener en cuenta los siguientes factores:

1. **Debe formar parte de un plan de seguridad**. Instalar Kerberos no supone resolver los problemas de seguridad. Un sistema Kerberos en una red sin seguridad es como ***un muro de papel con una puerta de acero***.
2. **Como mínimo necesita una máquina adicional para el AS y el TGS** que esté comunicada con todos los servicios **y otra para Backup**, porque la caída de la primera significaría la denegación de todos los servicios de la red. Estas máquinas **deben ser potentes** ya que todos los accesos pasan por ellas.
3. **La red debe ser rápida** porque el sistema Kerberos genera muchos mensajes adicionales y se puede colapsar.
4. Supone **gastos adicionales de personal** para el mantenimiento del sistema.
5. Como se comenta en capítulo 4.1, se **debe adaptar a Kerberos todo el software** que presta servicios con seguridad.
6. **Asegurar la compatibilidad con los nuevos sistemas de autenticación** que surgirán en los próximos años. Si el sistema Kerberos es incompatible con otros servicios pronto se quedará aislado de la red Internet.
7. **Necesita sincronización de todos los relojes de la red**.

Por lo tanto, la instalación de Kerberos debe salir de un estudio muy cuidadoso del estado de la red y las necesidades de la empresa.

5. Autenticación Windows NT

5.1. Esquema general

Windows NT es un sistema operativo que utiliza autenticación de acceso a los objetos siguiendo una organización tipo **DAC (Discretionary Access Controls)**. Así los propietarios de los objetos definen los permisos y derechos de los usuarios y grupos de usuarios.

Para iniciar el sistema operativo se necesita pasar un control de usuario protegido por contraseñas y después para acceder a máquinas remotas también se examina la identidad del usuario. La organización de los accesos a máquinas de la red tiene un sistema de gestión único en Windows NT y este trabajo analiza este tipo de sistema.

El sistema de acceso a máquinas remotas se puede hacer según **dos métodos diferentes**, es así porque se intentan cumplir dos **objetivos**:

- La **compatibilidad** con sistemas anteriores de Microsoft, como: Windows 95, Windows 3.11 o DOS y sencillez para entornos con pocas máquinas.
- Crear un método de **gestión centralizada** donde las contraseñas no se deben actualizar en todas las máquinas de un mismo grupo de trabajo.

Así el acceso a recursos de red se puede montar siguiendo uno de estos **modelos**:

- **Trabajo en grupo**. Este sistema implementa una red de igual a igual (*peer-to-peer*) que permite una gestión ágil para entornos con pocas máquinas, además de no necesitar la intervención de un Windows NT Server.
- **Dominios**. Un servidor centraliza todos los accesos a los servidores y clientes de su grupo, llamado dominio.

5.2. Modelo de trabajo en grupo

Este modelo es mucho más difícil de gestionar y se debe utilizar en los siguientes casos:

- **Redes pequeñas** con pocas máquinas y pocos usuarios, donde la creación de dominios es más complicada que la gestión de las bases de datos de usuarios.
- Redes donde **máquinas Windows 95, Windows 3.11 o DOS** deben compartir o acceder a recursos remotos.
- Redes Windows NT donde **todos los sistemas operativos son WorkStation** y no hay Servers. Aunque se aconseja instalar Servers si la red es un poco grande.

Este modelo permite **dos métodos** de compartir recursos:

- Método **compartido**.
- Método **de usuarios**.

El **método compartido** está pensado para máquinas con Windows 95 aunque se puede utilizar en Windows NT Workstation. El usuario local define los objetos que quiere compartir y les puede asignar una contraseña pero no un nombre de usuario. **La contraseña de un objeto compartido es la misma para todos los usuarios**. Cuando se accede a un objeto protegido el servidor siempre pide una contraseña con independencia del usuario. Este sistema es muy poco seguro porque implica el reparto de contraseñas a todos los usuarios que pueden acceder. (Ver Figura 5.2.1)

Realizar la gestión del método compartido se **complica** mucho si hay varios recursos, sobre todo porque:

- Puede haber **tantas contraseñas como recursos**, por lo tanto, el usuario debe guardar la gestión de muchas contraseñas. Además se deben transmitir personalmente a cada usuario y esto comporta un gran peligro.
- **Cualquier usuario puede dejar recursos de su máquina personal al acceso de cualquiera**. Así la seguridad de las máquinas está en manos del usuario y no del gestor de red, por lo tanto puede haber múltiples agujeros y plataformas en la red sin que el gestor lo sepa.

Es aconsejable sólo utilizar este sistema en casos de necesidad y prohibirlo para los usuarios de grandes redes.

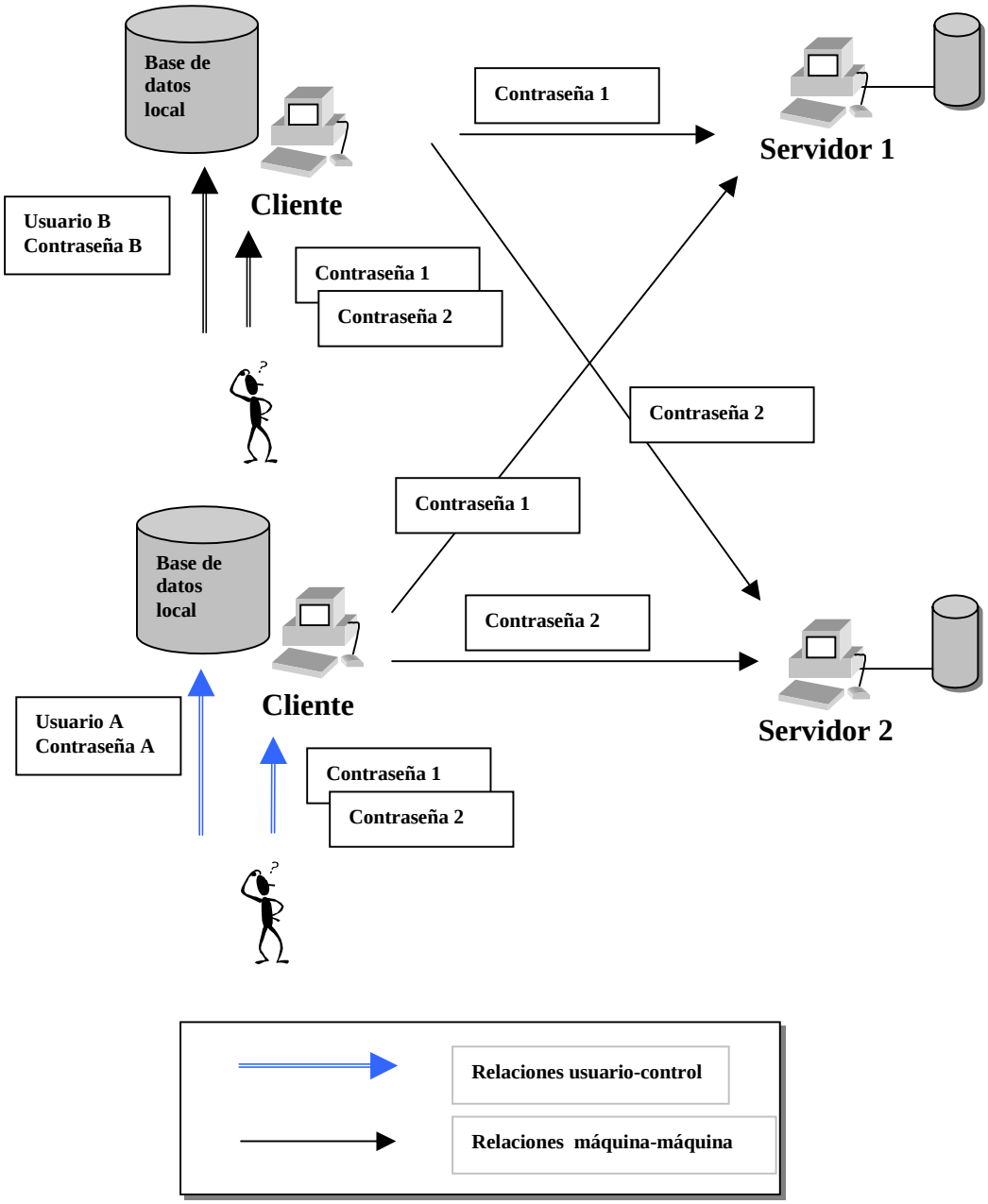

Figura 5.2.1: Acceso compartido en modelo de trabajo en grupo

El **método de usuarios** necesita una máquina Windows NT o un servidor NetWare. Los **accesos se realizan por nombre de usuario y contraseña**. La base de datos de usuarios se guarda en una máquina NT o NetWare, el nombre de esta máquina se debe indicar en todas las que quieren compartir recursos utilizando esa base de datos. Para compartir recursos **se asigna cada recurso a los usuarios de la base de datos**.

Cuando se accede a un recurso la máquina cliente envía los datos que el usuario introdujo para abrir la sesión en su máquina local, con estos datos se comprueba si puede acceder o no. Así **no se debe dar la contraseña cada vez que se accede** a un recurso (Ver Figura 5.2.2).

Este método **evita el tráfico de contraseñas** del método compartido pero aunque mejora la gestión no se puede considerar un sistema centralizado. Presenta los siguientes **problemas**:

- **Cualquier usuario puede dejar recursos de su máquina** a disposición de quien quiera. Esto permite tener máquinas no controladas que pueden ser agujeros o plataformas para otros ataques.
- Los usuarios **se deben dar de alta en su máquina y en la base datos** para los recursos. Si hay muchas bases de datos los cambios de contraseña se deben actualizar en todas.
- Las **máquinas DOS o Windows 95 no tienen seguridad de acceso físico**, así cualquiera puede acceder con cualquier nombre de usuario y copiar los ficheros internos del disco duro.
- **Sólo se puede utilizar desde máquinas Windwos NT donde la base de datos local tiene el mismo usuario que la remota**. Por lo tanto o sólo utilizan una máquina o se deben dar de alta en todas y así **se pierde la ventaja de sistema centralizado**.

Igualmente si se decide utilizar este método es aconsejable que el gestor de red controle las máquinas que pueden compartir recursos y prohiba a las otras dejar recursos para acceso remoto.

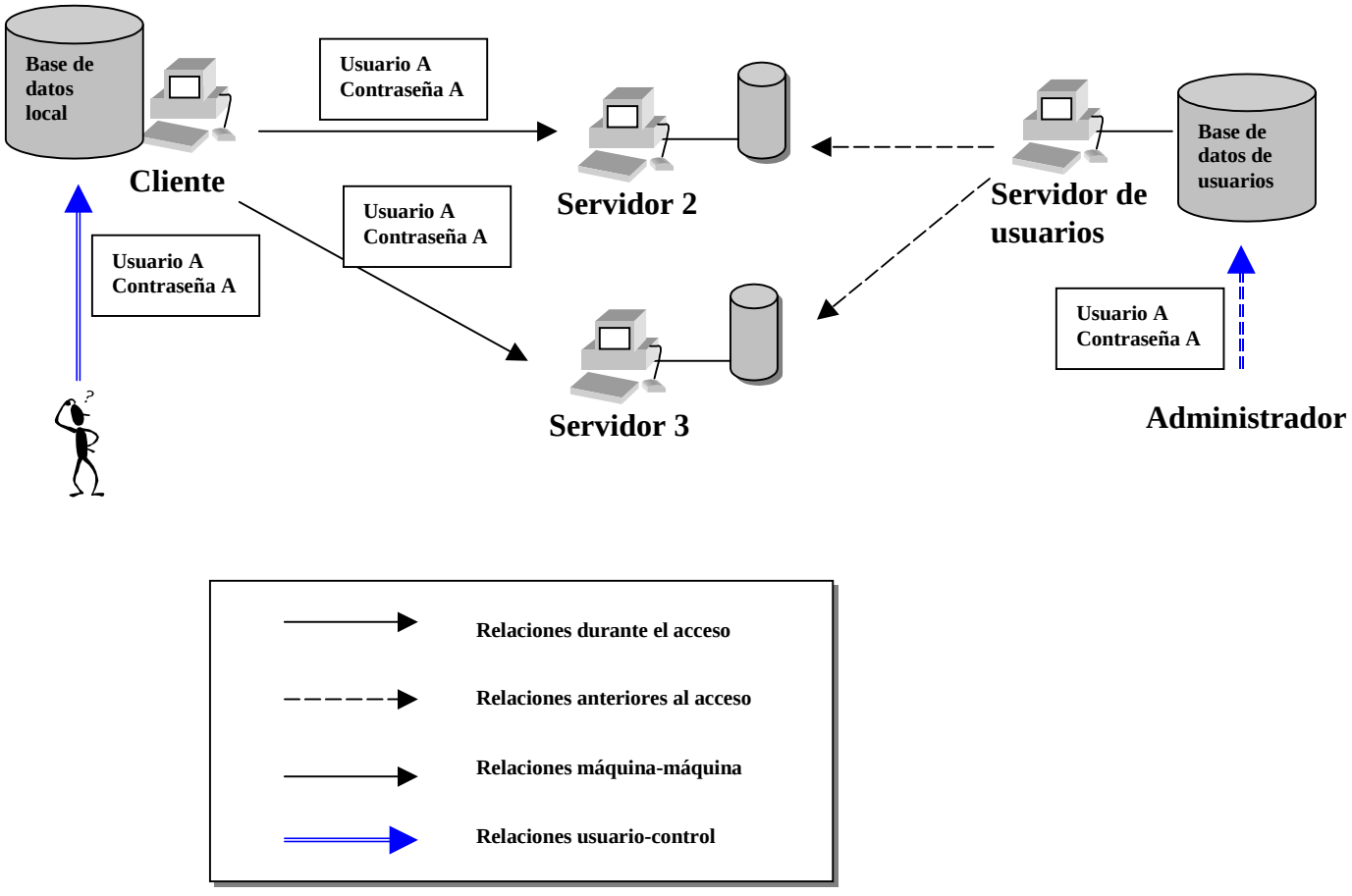

Figura 5.2.2: Acceso por usuarios en modelo de trabajo en grupo

5.3. Modelo de dominios

Es el **modelo de gestión centralizada ofrecido por Windows NT**. Como mínimo debe haber un Windows NT Server para cada dominio.

La base de datos de usuarios se guarda en el controlador de dominios (PDC) y se duplica en los controladores secundarios (BDCs). Todos los controladores deben ser Windows NT Server. La duplicación se realiza para mantener el acceso si el PDC cae y para no concentrar todos los accesos en la misma máquina (reparto de carga), frecuentemente los BDCs actualizan la base de datos, que puede ser únicamente los cambios o toda.

Los **objetivos** de los dominios son:

- **Centralizar el control de accesos a las máquinas clientes**. Así un usuario de un dominio se puede conectar a cualquiera de las máquinas de ese dominio, no hace falta que se dé de alta en la base de datos de todas.
- **Gestionar el acceso a los recursos de manera ordenada**. Desde la base de datos del dominio se puede acceder a los recursos de tu propio dominio sin necesidad de volver a entrar la contraseña.

Cuando un usuario inicia una sesión en su máquina local puede introducir el nombre del dominio de esta máquina o entrar de forma local. Si entra por el domino, la máquina envía al controlador de dominio el nombre de usuario y la contraseña y el **PDC realiza el control de accesos**. Así el nivel acceso a las máquinas locales y los servidores no depende de la base de datos local, **es independiente de la máquina de acceso y, por lo tanto, un sistema centralizado, siempre que la máquina de inicio de sesión esté dentro del mismo dominio**. (Ver Figura 5.3.1)

Dentro de una sesión de dominio, para acceder a un servidor del mismo dominio, la máquina local envía el nombre de usuario y la contraseña que el servidor comprueba en su base de datos. Si son servidores del dominio no PDC ni BDC pueden tener una base de datos de usuarios propia o utilizar la del dominio, pero nunca las dos. **La contraseña sólo se introduce al iniciar la sesión**.

Igualmente **se puede acceder en un servidor de dominio habiendo entrado a la máquina local sin especificar dominio**, pero el usuario y la contraseña **deben estar dados de alta en el servidor**.

En resumen el sistema de dominios **mejora** respecto a los de trabajo en grupo:

- Se puede acceder al servidor **desde cualquier máquina que pertenezca al dominio**, no hay dependencia de la máquina de acceso.
- La **base de datos** local de acceso de las máquinas se puede **centralizar en el controlador de dominio**.
- El **inicio de sesión dentro de un dominio** asegura que se **podrá acceder** a los servidores del mismo dominio.

Los usuarios de un dominio se pueden conectar con servidores de otros dominios mediante las **relaciones de confianza.**

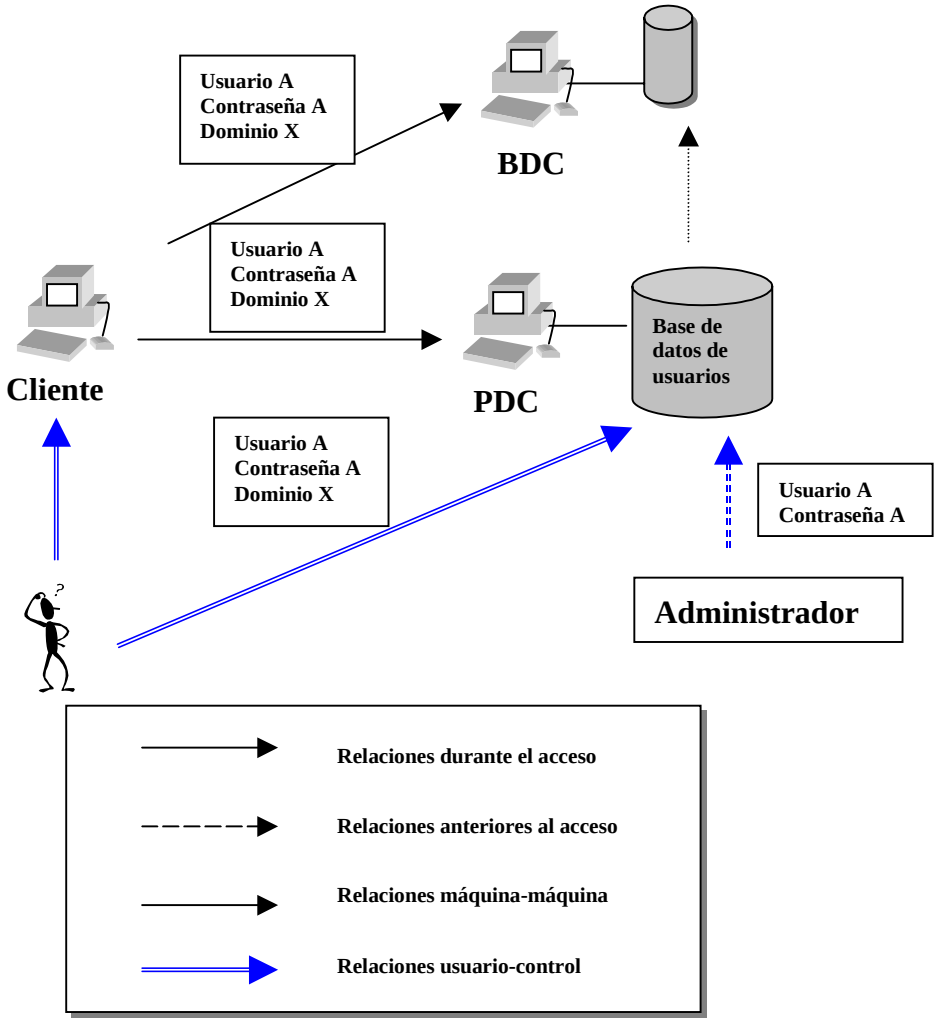

Figura 5.3.1: Acceso por el modelo de dominios.

5.4. Relaciones de confianza entre dominios

Las **relaciones de confianza** permiten a los usuarios de **un dominio acceder a recursos de otros dominios**. Se pueden establecer relaciones unidireccionales (uno confía en el otro) o bidireccionales (confianza mutua). Otra solución sería dar de alta a los usuarios en todos los dominios que necesitan pero esto rompería el modelo centralizado ya que las contraseñas se deberían actualizar en varios controladores. Así **con relaciones de confianza un usuario sólo debe estar en una base de datos y puede acceder a varios dominios**.

Se puede iniciar una sesión en un dominio de confianza. Si el controlador propio de la máquina ve que el dominio es otro de confianza delega el control de acceso al controlador del dominio que ha pedido el usuario.

También se puede acceder a recursos de dominios de confianza. Cuando se **accede a un servidor de otro dominio de confianza, éste envía el nombre de usuario y la contraseña al controlador del dominio del usuario**. Si el controlador admite el acceso, el servidor del dominio de confianza también. (Ver Figura 5.4.1)

Las relaciones de confianza son muy peligrosas porque los servidores no controlan directamente quién accede sino que delegan este control. Por este motivo se deben diseñar bien y gestionar de una manera centralizada. Microsoft recomienda utilizar una estructura jerárquica con un dominio maestro en la cabeza que tiene relaciones de confianza unidireccionales con todos los dominios de trabajo.

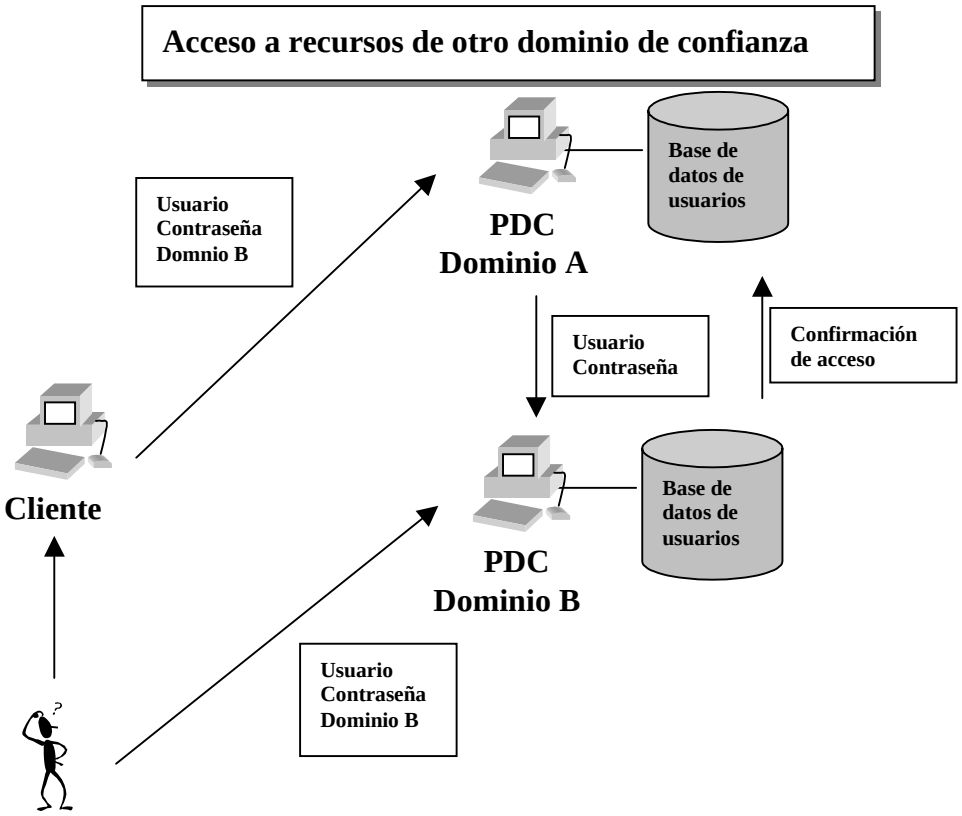

Figura 5.4.1: Acceso a un dominio de confianza.

5.5. Diferencias conceptuales con otros sistemas centralizados

El sistema de control de accesos Windows NT **no es un sistema completamente centralizado respecto a otros como los de directorios, como el NDS de Novell, o Kerberos**.

El mantenimiento del sistema de trabajo en grupo hace que la gestión sea complicada y permita la convivencia de diferentes controles de acceso en la misma red. Este problema se resolverá en las próximas versiones. La comparación se debe hacer entre el sistema de dominios y los otros.

Los **objetivos** del sistema de **dominios** son centralizar el control de **accesos remoto y local** mientras que los **otros sistemas** sólo controlan el **remoto**. Así en Kerberos o

directorios se debe realizar un control de acceso local (o acceso libre) y después otro remoto.

Así las **ventajas** de cada sistema son:

- **Windows NT sólo utiliza un control de accesos para local y remoto**.
- Los sistemas de **directorio y Kerberos** permiten el **acceso remoto desde cualquier máquina**, no hace falta que sea del dominio.

Para **solucionar este problema** Windows NT ha creado las **relaciones de confianza** que permiten a máquinas de un dominio acceder a los otros. Pero son relaciones de bloque, o sea, o todos los usuarios o ninguno, **no permiten casos personalizados**.

Por el mismo motivo, los **dominios de Windows** deben crear **una base de datos** de usuarios **en cada grupo** (servidor y sus clientes). En cambio **los otros sistemas** pueden **centralizar la base de datos** de usuarios en **una máquina** y, si fuera necesario para descargar tránsito, pueden dejar réplicas totales o parciales en otras máquinas. En Windows NT esto se realizaría con **un único dominio** y muchos BDCs pero entonces los usuarios **podrían acceder localmente a cualquier máquina cliente**. (Ver Figura 5.5.1)

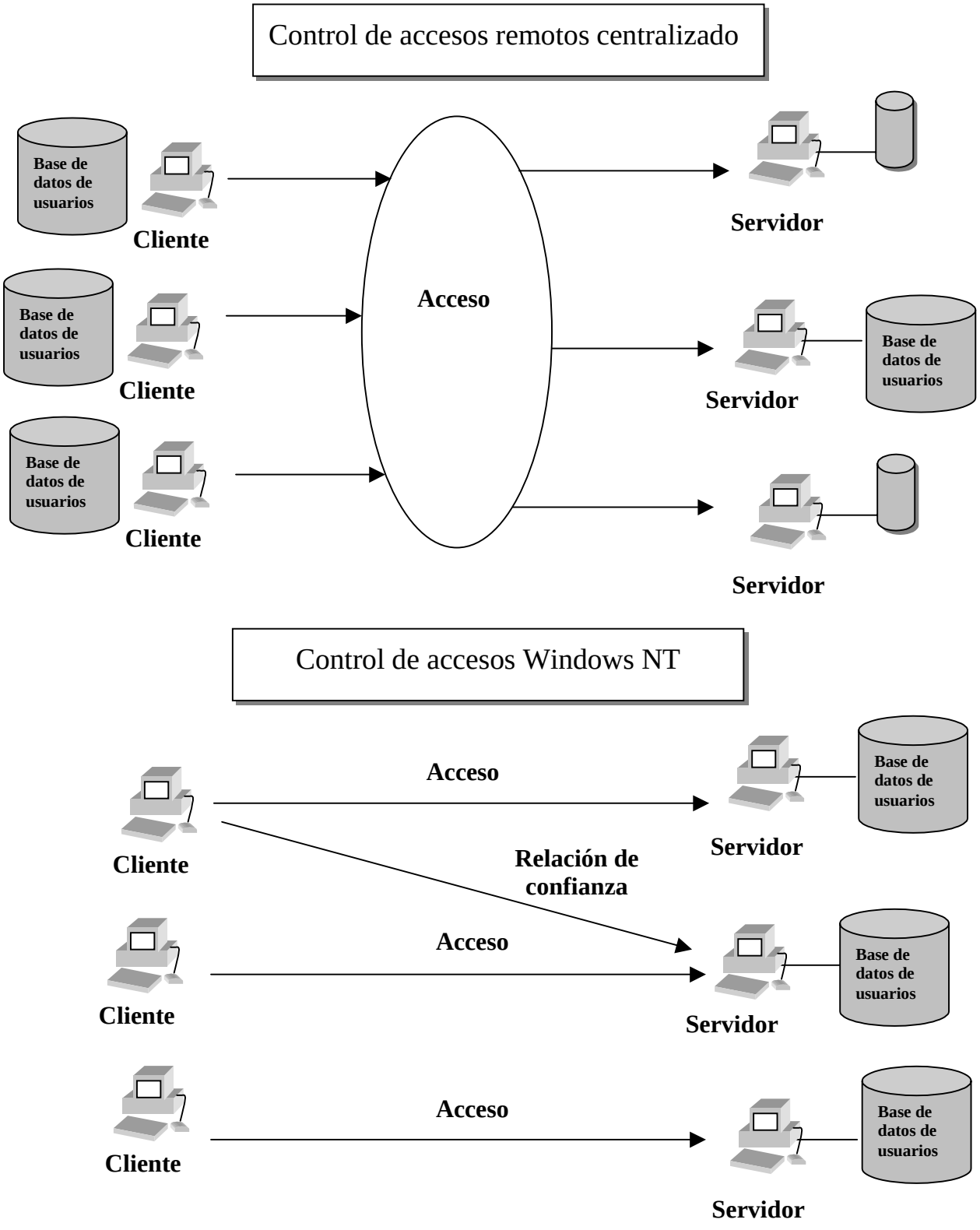

Figura 5.5.1: Control de acceso Windows NT frente a otros de control remoto