

## Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

# seguridad

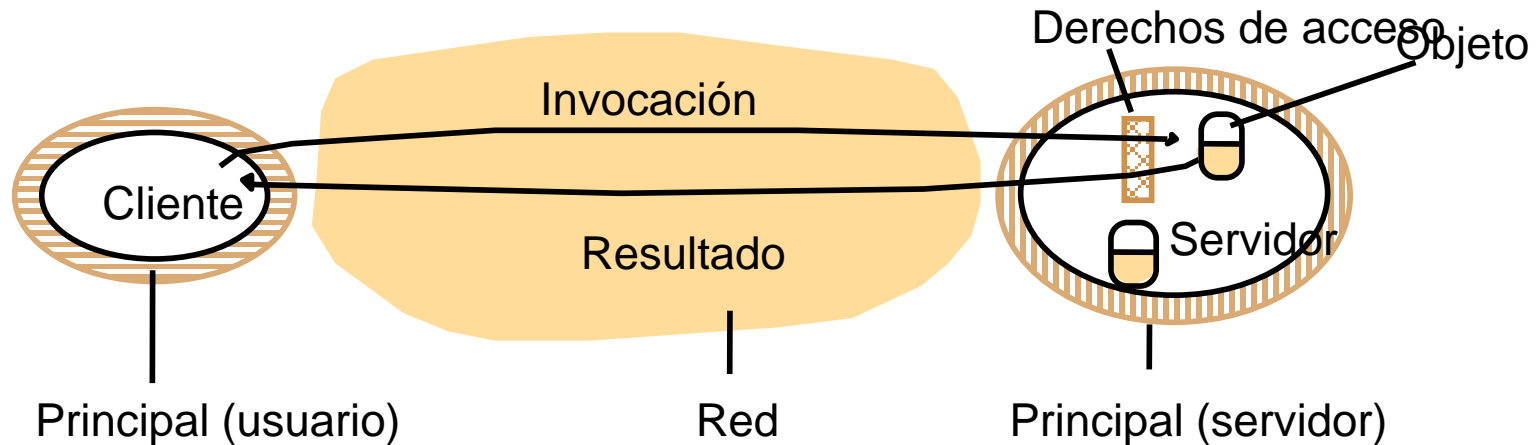
### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
seguridad  
coordinación  
transacciones

- @ Modelo de seguridad
  - Tipos de amenaza
- @ Técnicas básicas
  - Técnicas criptográficas
    - Secreto
    - Autenticación
    - Certificación y credenciales
    - Control de accesos
  - Auditoría de perfiles
- @ Algoritmos de encriptación simétricos y asimétricos
- @ Firmas digitales
- @ Aproximaciones al diseño de sistemas seguros
- @ Casos de estudio

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones



#### @ Objeto (o recurso)

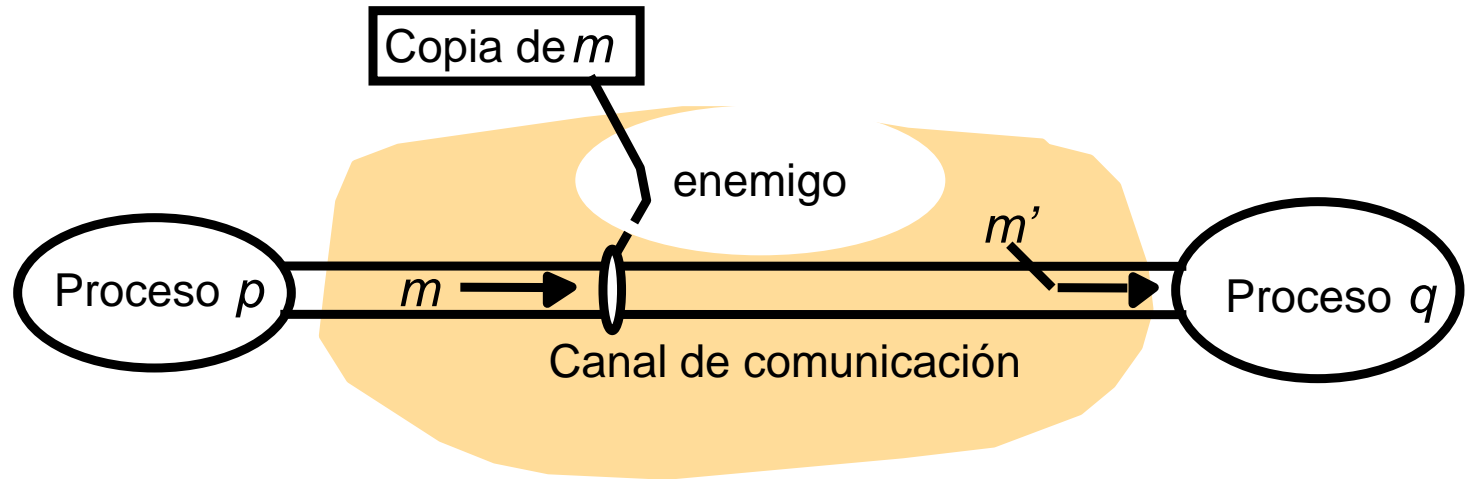
- Buzón de correo, sistema de archivo, parte de una web comercial

#### @ Principal

- Usuario o proceso que tiene derechos para realizar acciones
- La identidad del principal es importante

### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones



### @ Ataques

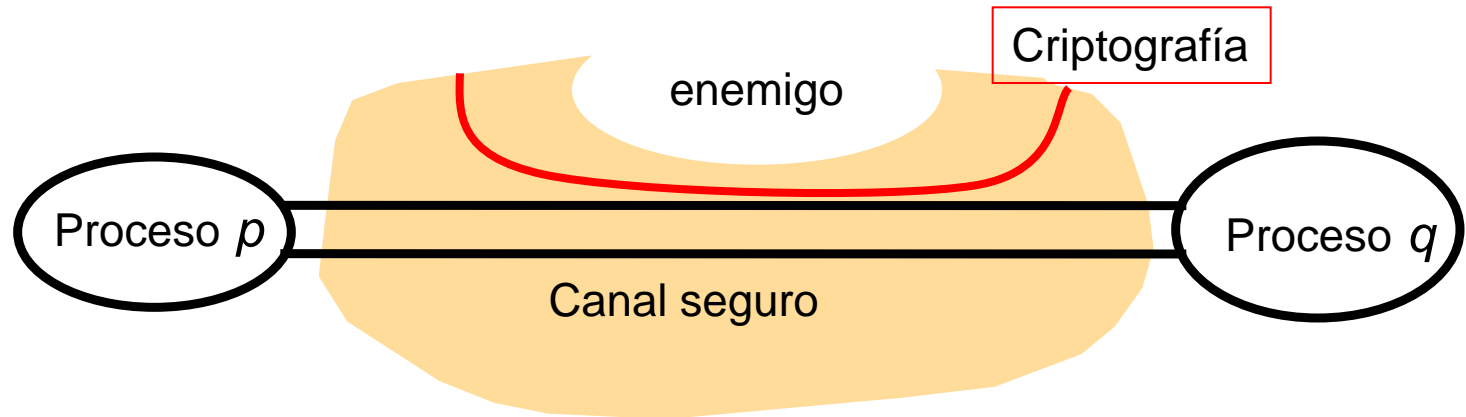
- En aplicaciones que manejan transacciones comerciales u otra información cuyo secreto o integridad es crucial

### @ Amenazas

- A procesos, a los canales de comunicación, denegación de servicio

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones



#### Propiedades

- *Cada proceso está seguro de la identidad del otro*
- *Los datos son privados y protegidos contra la manipulación*
- *Protección contra repeticiones y reordenación de datos*

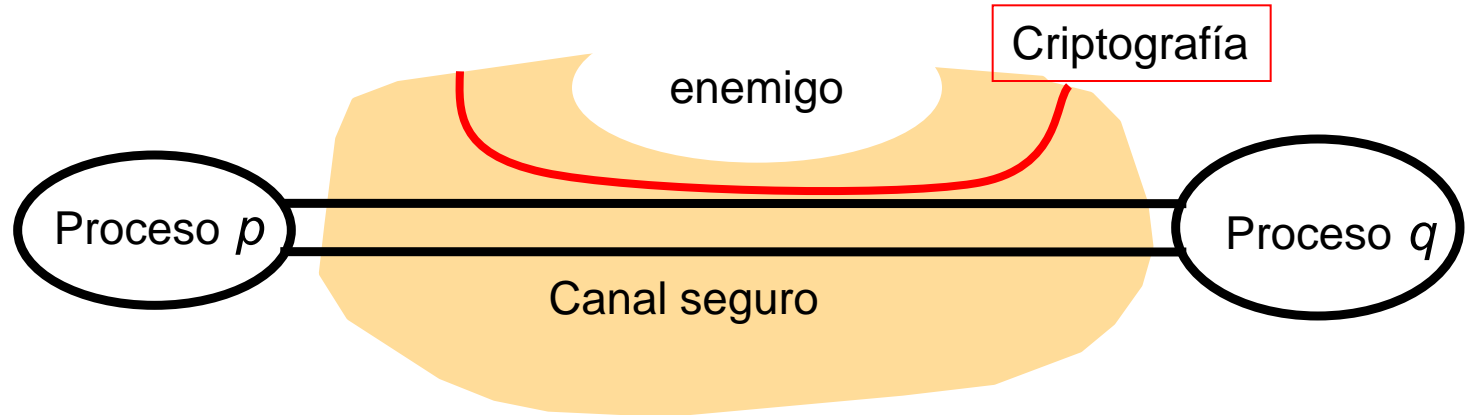


#### Utiliza criptografía

- *El secreto se preserva mediante ocultamiento criptográfico*
- *La autenticación basada en la prueba de posesión de secretos*

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones



#### Ocultamiento criptográfico basado en:

*Confusión y difusión*

*o el otro*

- Los datos son privados y protegidos contra la manipulación
- Protección contra repeticiones y reordenación de datos

#### Posesión de secretos:

Claves convencionales compartidas

Pares de claves públicas/privadas

*amamiento criptográfico*

*de posesión de secretos*

#### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

- @ Escuchar a escondidas
  - Obteniendo información privada o secreta
- @ Enmascarse
  - Asumiendo la identidad de otro usuario/principal
- @ Manipular mensajes
  - Alterando el contenido de mensajes en tránsito
- @ Reenviar
  - Almacenando mensajes seguros y enviándolos más tarde
- @ Negación de servicio
  - Inundando un canal u otro recurso, negando acceso para los otros

#### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

#### @ Ataques de negación de servicio

- El uso excesivo de recursos hasta el grado de impedir su uso a usuarios legítimos
  - *por ejemplo, el ataque a Amazon y Yahoo en febrero del 2000*

#### @ Los caballos de Troya y otros virus

- Los virus sólo pueden entrar en computadoras cuando el código de programa es importado.
- Pero los usuarios a menudo requieren programas nuevos:
  - *La instalación nueva de software*
  - *Código móvil importado dinámicamente (p. e., los applets Java)*
  - *La ejecución accidental de programas transmitidos subrepticamente*

Defensas: autenticación de código (mediante firmas),  
validación de código (comprobación de tipo), seguridad  
JVM... **ANÁLISIS, DISEÑO Y PRUDENCIA** ▶



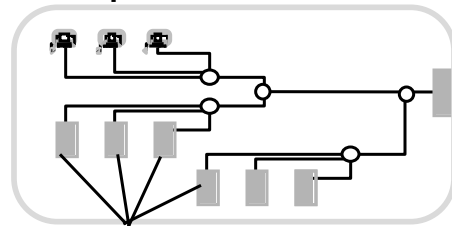
## seguridad

ejemplo: todo empezó con un *ping*...

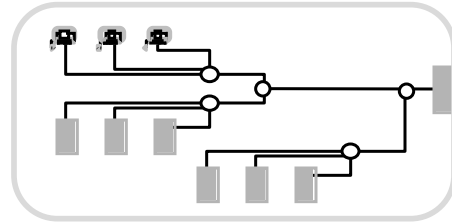
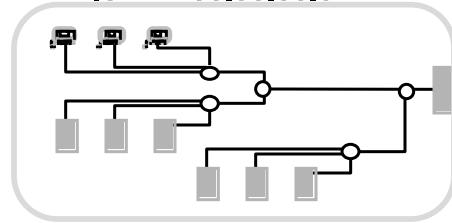
### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

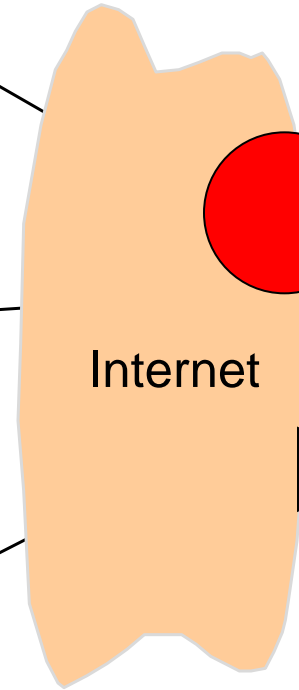
#### Campus intranets



IP = n.n.n.i



#### Firewall



amazon.com  
IP = x.x.x.x

yahoo.com  
IP = y.y.y.y

*Desde un servidor malicioso se hacen ping a muchas máquinas:*

¡falso origen!

PING | source = x.x.x.x | destination = n.n.n.i

*...resultando:*

PONG | source = n.n.n.i | destination = x.x.x.x

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

Alice	Primer participante
Bob	Segundo participante
Carol	Otro participante en los protocolos a tres o cuatro bandas
Dave	Participante en los protocolos a cuatro bandas
Eve	Fisgón
Mallory	Atacante malevolente
Sara	Un servidor

$K_A$	Clave secreta de Alice
$K_B$	Clave secreta de Bob
$K_{AB}$	Clave secreta compartida por Alice y Bob
$K_{Apriv}$	Clave privada de Alice (sólo conocida por Alice)
$K_{Apub}$	Clave pública de Alice (publicada por Alice para la lectura de cualquiera)
$\{M\}_K$	Mensaje $M$ encriptado con la clave $K$
$[M]_K$	Mensaje $M$ firmado con la clave $K$

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

Alice y Bob comparten una clave secreta  $K_{AB}$

1. Alice usa  $K_{AB}$  y acuerda una función de encriptación  $E(K_{AB}, M)$  para codificar y enviar una serie de mensajes  $\{M_i\}_{K_{AB}}$
2. Bob lee los mensajes encriptados usando la correspondiente función  $D(K_{AB}, M)$ .

Alice y Bob pueden funcionar con  $K_{AB}$  mientras estén seguros que  $K_{AB}$  no es conocida

#### Problemas:

- *Distribución de clave:* ¿Cómo envía Alice una clave compartida a Bob de forma segura?
- *Caducidad de la comunicación:* ¿Cómo sabe Bob que el mensaje no es una copia capturada por Mallory y reenviada más tarde?

#### Contenido

introducción

fundamentos

tecnologías

nombres

tiempo

seguridad

coordinación

transacciones

Bob es un servidor de ficheros; Sara es un servidor de autenticación. Sara comparte  $K_A$  con Alice y  $K_B$  con Bob

1. Alice envía un mensaje no encriptado a Sara identificándose y solicitando un *ticket* para acceder a Bob. ➡
2. Sara responde a Alice con  $\{\{\text{Ticket}\}_{K_B}, K_{AB}\}_{K_A}$ . Consistente en un mensaje codificado según  $K_A$  con un ticket (para comunicar con Bob para cada fichero) encriptado según  $K_B$  y una nueva clave  $K_{AB}$ .
3. Alice usa  $K_A$  para desenscriptar la respuesta.
4. Alice envía a Bob el ticket, su identidad y una respuesta  $R$  para acceder al fichero:  $\{\text{Ticket}\}_{K_B}, \text{Alice}, R$ .
5. El ticket es realmente  $\{K_{AB}, \text{Alice}\}_{K_B}$ . Bob usa  $K_B$  para desenscriptarlo, chequea la identidad y usa  $K_{AB}$  para encriptar las respuestas a Alice.

## seguridad

### escenario 2: autenticación con servidor

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo

#### seguridad

- coordina
- transacci

Un ticket es un mensaje encriptado conteniendo la identidad del principal solicitante y una clave compartida para la sesión

- @ Esto es una simplificación del protocolo Needham and Schroeder (y Kerberos)
- @ Edad y repetición – resuelto en N-S y Kerberos completo
- @ No válido para comercio electrónico...

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

Bob genera un par de claves pública/privada  $\langle K_{Bpub}, K_{Bpriv} \rangle$

1. Alice obtiene un certificado firmado por una autoridad de confianza que posee la clave pública de Bob,  $K_{Bpub}$
2. Alice crea una clave compartida  $K_{AB}$ , la encripta según  $K_{Bpub}$  un algoritmo de clave pública y envía el resultado a Bob
3. Bob usa  $K_{Bpriv}$  para desencriptar  $K_{AB}$ .

(si desean asegurar que el mensaje no ha sido manipulado, Alice puede incluir algún dato aceptado por ambos y Bob chequearlo )

Problemas:

- Ⓜ Mallory puede interceptar la solicitud de certificado de clave pública y enviarle su propia clave pública, pudiendo desencriptar el resto de mensajes. La firma digital lo impide

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

Alice quiere publicar un documento  $M$  de forma que cualquiera pueda verificar su procedencia

1. Alice calcula un resumen de longitud fija del documento  $\text{Resumen}(M)$  ➡
2. Alice encripta el resumen con su clave privada, lo adjunta a  $M$  y hace el resultado  $(M, \{\text{Resumen}(M)\}_{K_{\text{Apriv}}})$  público
3. Bob obtiene el documento firmado, extrae  $M$  y computa  $\text{Resumen}(M)$
3. Bob usa la clave pública de Alice para desencriptar  $\{\text{Resumen}(M)\}_{K_{\text{Apriv}}}$  y lo compara con el resumen calculado por él. Si coincide, entonces la firma es válida.

@ La función de resumen debe ser segura frente al "ataque del cumpleaños"

#### Contenido

introducción

fundamentos

tecnologías

nombres

tiempo

seguridad

coordinación

transacciones

Función de resumen seguro  $h=H(M)$ :

1. Dado  $M$ , debe ser fácil calcular  $h$
2. Dado  $h$ , debe ser muy difícil calcular  $M$
3. Dado  $M$ , debe ser difícil encontrar otro  $M'$ , tal que  $H(M)=H(M')$

También llamada función de dispersión de un solo sentido

Ataque sustentado sobre la "*paradoja del cumpleaños*":

La probabilidad de encontrar un par idéntico en un conjunto es mucho mayor que la de encontrar la pareja para un individuo dado. Con paciencia...



#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

1. Alice prepara dos versiones  $M$  y  $M'$  de un contrato para Bob.  $M$  favorable y  $M'$  desfavorable
2. Alice fabrica varias versiones de  $M$  y  $M'$  sutilmente diferentes (espacios al final de línea,...). Ella compara los valores de dispersión de todos los  $M$  con todos los  $M'$  buscando un par igual
3. Alice envía el contrato favorable  $M$  a Bob, éste lo firma digitalmente usando su clave privada
4. Cuando lo devuelve, Alice sustituye  $M$  por  $M'$ , pero manteniendo la firma de Bob sobre  $M$

#### Contenido

introducción

fundamentos

tecnologías

nombres

tiempo

seguridad

coordinación

transacciones

1. Alice prepara dos versiones M y M' de un contrato para Bob. M favorable y M' desfavorable
2. Alice fabrica varias versiones de M y M' sutilmente diferentes

Por ejemplo, que para generar colisiones en una función aleatoria perfecta (en funciones hash) de n bits, con una probabilidad del 50% aproximadamente, se requieren solo  $2^{n/2}$  intentos.

3. Alice envía el contrato favorable M a Bob, éste lo firma digitalmente usando su clave privada
4. Cuando lo devuelve, Alice sustituye M por M', pero manteniendo la firma de Bob sobre M

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
seguridad  
coordinación  
transacciones

## Certificado de cuenta de Alice

- |                                    |   |
|------------------------------------|---|
| 1. <i>Tipo de certificado:</i>     | Número de cuenta  |
| 2. <i>Nombre:</i>                  | Alice   |
| 3. <i>Cuenta:</i>                  | 6262626   |
| 4. <i>Autoridad certificadora:</i> | Banco de Bob  |
| 5. <i>Firma:</i>                   | $\{\text{Resumen}(\text{campo 2} + \text{campo 3})\}_{K_{Bpriv}}$ |

## Certificado de clave pública del Banco de Bob

- |                                    |   |
|------------------------------------|---|
| 1. <i>Tipo de certificado:</i>     | Clave pública   |
| 2. <i>Nombre:</i>                  | Banco de Bob  |
| 3. <i>Cuenta:</i>                  | $K_{Bpub}$  |
| 4. <i>Autoridad certificadora:</i> | Fred, la Federación de Banqueros                                |
| 5. <i>Firma:</i>                   | $\{\text{Resumen}(\text{campo2} + \text{campo3})\}_{K_{Fpriv}}$ |

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

**Certificado:** sentencia firmada por un principal que sirve de credencial y/o autenticación.

Un certificado necesita:

- Un formato estándar acordado
- Acuerdo sobre la forma en que se construyen las cadenas de certificados
- Fechas de expiración, de forma que pueda ser revocado

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

seguridad

coordinación  
transacciones

Mensaje: M, clave: K, funciones criptográficas E, D

@ Simétricos (clave secreta)

$$E(K, M) = \{M\}_K$$

$$D(K, E(K, M)) = M$$

La misma clave para E y D

M debe ser difícil de computar si se desconoce K

La forma usual de ataque es la fuerza bruta. Resistente haciendo K suficientemente grande ~ 128 bits

@ Asimétricos (clave pública)

Claves de encriptación y desencriptación separadas:  $K_e$ ,  $K_d$

$$D(K_d, E(K_e, M)) = M$$

se basa en el uso de funciones de puerta falsa. E tiene un alto coste computacional. Las claves son muy grandes > 512 bits

@ Protocolos híbridos – usados en SSL (actualmente llamado TLS)

Usa criptografía asimétrica para transmitir la clave simétrica que es usada para encriptar la sesión

**Contenido**

introducción

fundamentos

tecnologías

nombres

tiempo

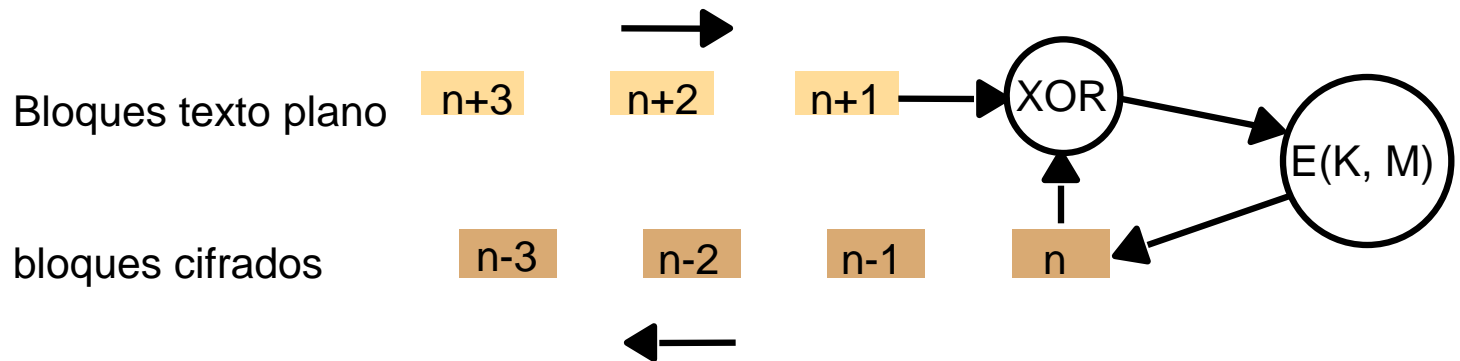
seguridad

coordinación

transacciones

La mayoría de cifradores trabajan sobre bloques de 64 bits

**Debilidad** de un cifrador de bloque simple: los patrones repetidos pueden ser detectados

**CIFRADOR DE CADENA (CBC)**

1. El bloque encriptado en el paso anterior es combinado con el siguiente mediante XOR
2. Existe debilidad en el primer bloque cifrado. Se usa **vector de inicialización**
3. La conexión debe ser fiable, no se pueden perder bloques

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo

#### seguridad

- coordinación
- transacciones

Todos estos algoritmos realizan operaciones de confusión y de difusión sobre bloques de datos binarios

- Ⓢ **TEA:** un simple pero efectivo algoritmo desarrollado en U. Cambridge (1994) [lo explicaremos a continuación]. Clave de *128-bit, 700 kbytes/s*
- Ⓢ **DES:** US Data Encryption Standard (1977). No demasiado fuerte en su formato original. Clave de *56-bit, 350 kbytes/s*
- Ⓢ **Triple-DES:** aplica DES tres veces con dos claves distintas.  $E_{DES}(K_1, D_{DES}(K_2, E_{DES}(K_1, M)))$ . Clave *112-bit, 120 KB/s*
- Ⓢ **IDEA:** International Data Encryption Algorithm (1990). Parecido al TEA. *128-bit key, 700 kbytes/sec*
- Ⓢ **AES:** US Advanced Encryption Standard (1997). Clave de *128/256-bit*

*Las mediciones se refieren a un Pentium II a 330 MHZ*

# seguridad

## algoritmo de encriptación TEA

### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
seguridad  
coordinación  
transacciones

clave 4 x 32 bits

```
void encrypt(unsigned long k[], unsigned long text[]) {  
    unsigned long y = text[0], z = text[1];  
    unsigned long delta = 0x9e3779b9, sum = 0; int n;  
    for (n = 0; n < 32; n++) {  
        sum += delta;  
        y += ((z << 4) + k[0]) ^ (z + sum) ^ ((z >> 5) + k[1]);  
        z += ((y << 4) + k[2]) ^ (y + sum) ^ ((y >> 5) + k[3]);  
    }  
    text[0] = y; text[1] = z;  
}
```

texto plano  
y resultado  
2 x 32

XOR

desplazamiento

© Triple de veloz que el DES



#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

- @ Todos ellos dependen del uso de funciones de puerta falsa:
  - funciones de un solo sentido con una salida secreta: p.e. producto de dos números grandes (primos); fácil de multiplicar, imposible de factorizar (obtener multiplicandos)
- @ **RSA**: El primer algoritmo práctico (Rivest, Shamir y Adelman 1978) y el más frecuentemente usado. Tamaño de la clave puede variar, 512-2048 bits. Velocidad 1-7 kbytes/s
- @ **Curvas elípticas**: Método reciente, claves más cortas y más veloz (Menezes 1993 – elliptic curve public key crypto)
- @ Los algoritmos asimétricos son ~1000 veces más lentos y no son prácticos para encriptaciones masivas; sin embargo, sus propiedades los hacen idóneos para distribución de claves y para autenticación

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
seguridad  
coordinación  
transacciones

Para encontrar el par de claves  $e, d$ :

1. Elegir dos primos muy grandes,  $P$  y  $Q$  (mayor de  $10^{100}$ ), y calcular:

$$N = P \times Q$$

$$Z = (P-1) \times (Q-1)$$

2. Para  $d$  elegir un número primo respecto a  $Z$  (es decir,  $d$  no tiene factores comunes con  $Z$ ).

Ilustramos los cálculos con valores pequeños de  $P$  y  $Q$ :

$$P = 13, Q = 17 \rightarrow N = 221, Z = 192$$

$$d = 5$$

3. Para encontrar  $e$  se resuelve la ecuación:

$$e \times d = 1 \bmod Z$$

$e \times d$  es el elemento más pequeño divisible por  $d$  en la serie  $Z+1, 2Z+1, 3Z+1, \dots$

$$e \times d = 1 \bmod 192 = 1, 193, 385, \dots$$

385 es divisible por  $d$

$$e = 385/5 = 77$$

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

seguridad

coordinación  
transacciones

Para encriptar según RSA, el texto se divide en bloques de  $k$  bits donde  $2^k < N$  (el valor numérico de un bloque es siempre menor que  $N$ ;  $k$  entre 512 y 1024)

$k = 7$ , entonces  $2^7 = 128 (< N = 221)$

La **función de encriptación** de un bloque de texto  $M$  es:

$$E'(e, N, M) = M^e \bmod N$$

para  $M$ , el texto cifrado es  $M^{77} \bmod 221$

La **función de descriptación** del bloque cifrado  $c$  es:

$$D'(d, N, c) = c^d \bmod N$$

Rivest, Shamir and Adelman probaron que  $E'$  y  $D'$  son inversas mutuas:

$$E'(D'(x)) = D'(E'(x)) = x) \quad 0 \leq P \leq N$$

#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

- Ⓢ **MD5:** Desarrollado por Rivest (1992). *Calcula un resumen de 128 bits. Velocidad: 1740 kbytes/s*
  - Cuatro vueltas con una de cuatro **funciones no lineales** sobre cada 32 bits de un bloque de 512 bits de texto

- Ⓢ **SHA:** (1995) basado en MD4 de Rivest, pero más seguro, produce un resumen de 160-bit. *Velocidad: 750 kbytes/s*

**Cualquier algoritmo simétrico se puede usar en CBC (cifrador de cadena):**

El último bloque es el resumen  $H(M)$



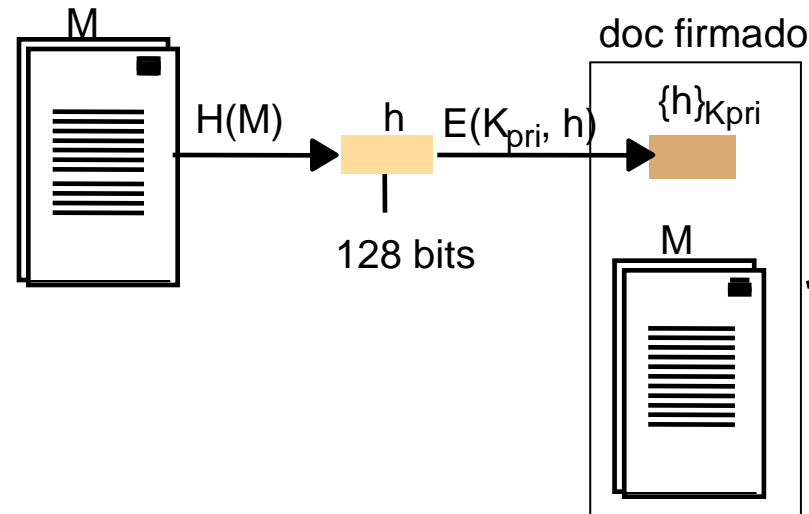
# seguridad

## firma digital con claves públicas

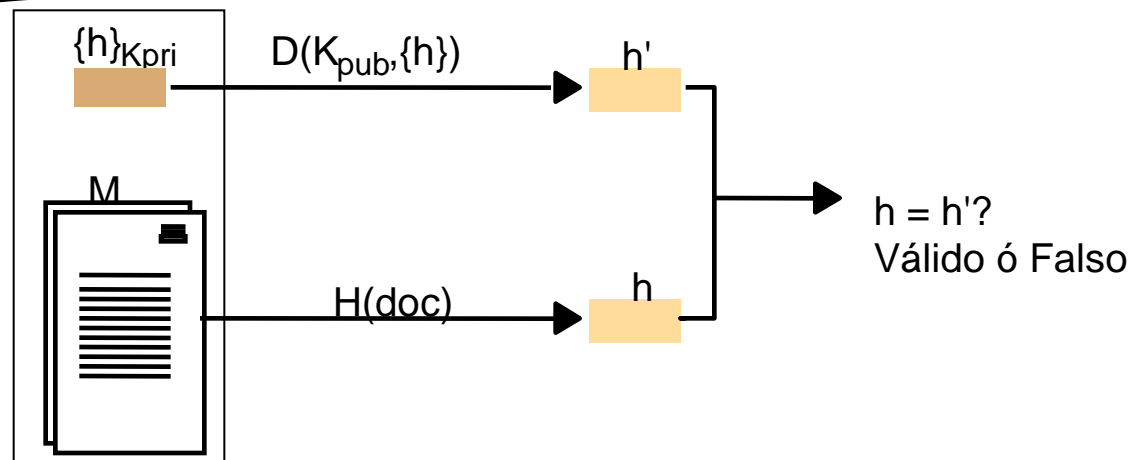
### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

firma



verificación



#### Contenido

- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

@ En los primeros sistemas distribuidos (1974-84) era difícil proteger los servidores:

- P.e. contra ataques enmascarados sobre un servidor de ficheros
- No había mecanismos de autenticación del origen de la petición
- La criptografía de clave pública no estaba disponible
  - computadoras demasiado lentas para cálculos importantes
  - RSA no disponible hasta 1978

@ Needham y Schroeder desarrollaron un protocolo de autenticación y distribución de claves para uso en red local:

- Supuso un primer ejemplo del cuidado en el diseño de protocolos de seguridad
- Introdujeron varias ideas de diseño: p.e. *ocasiones* ▶

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

**seguridad**

coordinación  
transacciones

<i>Encabezado</i>	<i>Mensaje</i>	<i>Notas</i>
1. A->S:	$A, B, N_A$	A solicita una clave a S para comunicarse con B
2. S->A:	$\{N_A, B, K_{AB},$ $\{K_{AB}, A\}_{K_B}\}_{K_A}$	S devuelve un mensaje encriptado en la clave secreta de A, con una clave nueva $K_{AB}$ y un “ticket” encriptado en la clave secreta de B. La ocasión $N_A$ demuestra que el mensaje fue enviado en respuesta al anterior. A confía en que S envió el mensaje porque sólo S conoce la clave secreta de A
3. A->B:	$\{K_{AB}, A\}_{K_B}$	A envía el “ticket” a B
4. B->A:	$\{N_B\}_{K_{AB}}$	B descripta el “ticket” y utiliza la nueva clave $K_{AB}$ para encriptar otra ocasión $N_B$
5. A->B:	$\{N_B - 1\}_{K_{AB}}$	A demuestra a B que fue el emisor del mensaje anterior devolviendo una transformación acordada sobre $N_B$ .

Ticket

**Contenido**

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

## seguridad

coordinación  
transacciones

<i>Encabezado Mensaje</i>		<i>Notas</i>
1. A->S:	$A, B, N_A$	A solicita una clave a S para comunicarse con B
$N_A$ es una <b>ocasión</b> : entero que se añaden a los mensajes para demostrar la frescura de la transacción. Son generados por el proceso emisor cuando se necesita (p.e. incrementando contador o leyendo el tic del reloj)		
3. A->B:	$\{K_{AB}, A\}_{K_B}$	A A envía el "ticket" a B
4. B->A:	$\{N_B\}_{K_{AB}}$	B descripta el "ticket" y utiliza la nueva clave $K_{AB}$ para encriptar otra ocasión $N_B$
5. A->B:	$\{N_B - 1\}_{K_{AB}}$	A demuestra a B que fue el emisor del mensaje anterior devolviendo una transformación acordada sobre $N_B$ .



#### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

**seguridad**

coordinación  
transacciones

- @ Comunicación segura con servidores en una red local
  - Desarrollado en el MIT en 80s para ofrecer seguridad en la red del campus > 5000 usuarios
  - basado en Needham - Schroeder
- @ Estandarizado e incluido en muchos SO
  - Internet RFC 1510, OSF DCE
  - BSD UNIX, Linux, Windows 2000, NT, XP, etc.
  - Disponible en la web del MIT
- @ El servidor Kerberos crea una clave secreta compartida para cada servidor solicitado y la envía encriptada al computador del usuario
- @ El password del usuario es el secreto compartido inicial en Kerberos

# Arquitectura del sistema Kerberos

## Contenido

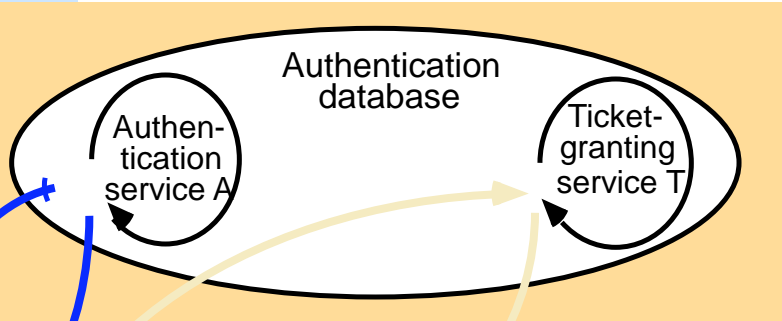
Introducción

Centro de distribución de claves de Kerberos

TGS: Ticket-granting service

### Step A

1. Request for TGS ticket
2. TGS ticket



### Step B

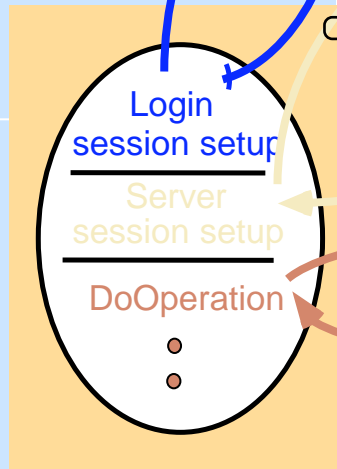
3. Request for server ticket
4. Server ticket

### Step C

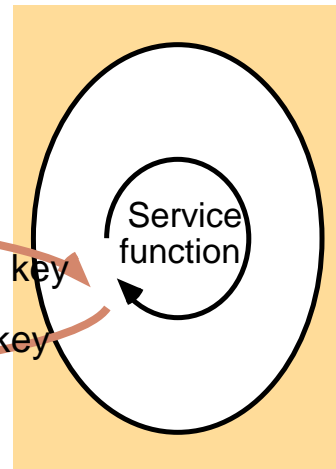
5. Service request

Request encrypted with session key

Reply encrypted with session key



Client



Server

## Protocolo Needham - Schroeder

1.  $A \rightarrow S: A, B, N_A$
2.  $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_B}\}_{K_A}$
3.  $A \rightarrow B: \{K_{AB}, A\}_{K_B}$
4.  $B \rightarrow A: \{N_B\}_{K_{AB}}$
5.  $A \rightarrow B: \{N_B - 1\}_{K_{AB}}$

**Paso A** una vez por inicio de sesión

**Paso B** una vez por sesión cliente-servidor

**Paso C** una vez por transacción del servidor

#### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo

seguridad

coordinación  
transacciones

1. Educación de los usuarios: ninguna herramienta puede proteger de los errores de un usuario. Deben sensibilizarse sobre lo que puede ocurrir. El miedo o la prohibición no son soluciones.

Los programas de sensibilización deben tratar:

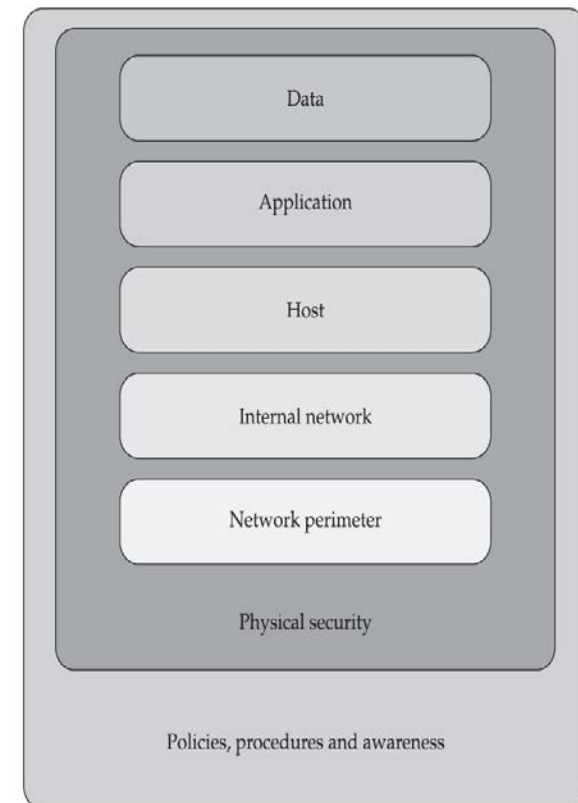
- Política de seguridad corporativa
- Establecimiento de contraseñas y renovación
- Comportamiento ante los virus y prevención
- Uso y abuso del email: puerta de entrada a infección
- Acceso a Internet: un privilegio, no un derecho
- Robo de dispositivos portátiles: proteger los datos
- Ingeniería social
- Importancia del control de acceso a las instalaciones
- Régimen de regulación y jerarquía

### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

## 2. Defensa elástica: en vez de parar el ataque, se busca ralentizarlo al máximo

- Firewalls
- Filtros de paquetes
- Redes virtuales (VPN)
- Acceso temporizado
- Biometría
- Soft/hard no conectado exterior
- Auditoría y logs
- etc.



#### Contenido

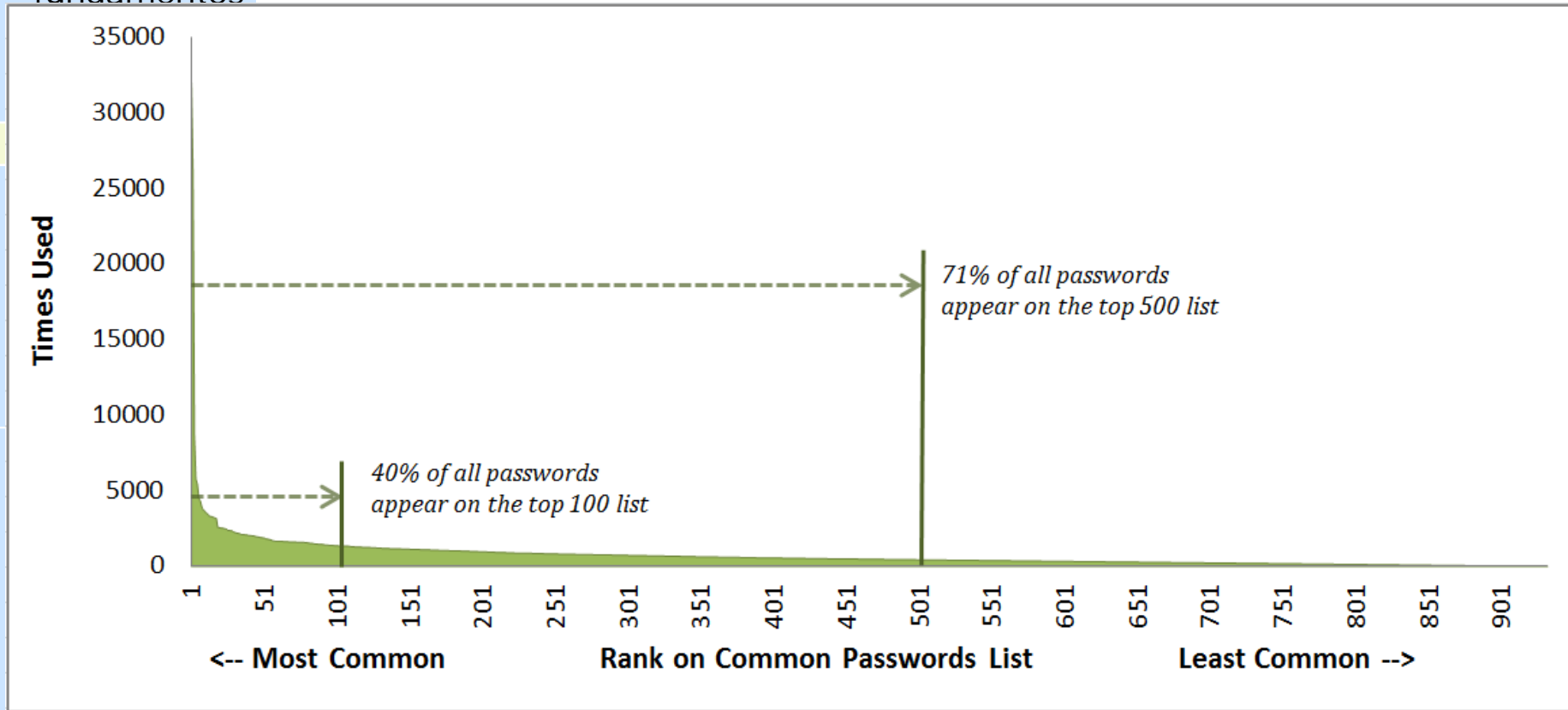
- introducción
- fundamentos
- tecnologías
- nombres
- tiempo
- seguridad**
- coordinación
- transacciones

3. *Robustecimiento del sistema*: eliminar utilidades y programas no esenciales. Parar cualquier servicio innecesario. Evitar arranques SO desde elementos externos. TCP/IP único protocolo instalado. Evitar compartición de ficheros e impresoras. Eliminar cuentas "guest". Renombrar a "root". Enjaulado de servicios (chroot).
4. *Actualizaciones automáticas*: parches y mejoras
5. *Virtualización*: se mejoran consumos y mantenimiento, recuperación ante desastres y procedimientos de seguridad. Mejor que el "enjaulado de servicios".
6. *Uso de herramientas de control (unix)*: rkhunter (compara hash de archivos con originales), chkrootkit (shell script),...
7. *Registros externalizados*, guardando regularmente copias de seguridad de los mismos.

#### Contenido

introducción  
fundamentos

<http://xato.net/files/10k%20most%20common.zip>



<http://xato.net/files/10k%20most%20common.zip>



### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

Existen diversas distribuciones linux que recogen las herramientas más actualizadas para realizar "auditoría" de redes wifi y cableadas

- WifiSlax (<http://www.wifislax.net/>)
- BackTrack (<http://www.kali.org/>)
- Wifiway (<http://www.wifiway.org>)

### Herramientas (ataque WEP)

- Kismet (inspección)
- Airodump (sniffer y almacén)
- Aireplay (reinyección)
- Aircrack (extracción de clave → estadísticamente)
- Macchanger → en caso de filtrado MAC



## Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

## ¿Por qué proteger la red wifi?

- El tráfico de red puede ser capturado y examinado
  - URLs, websites, ...
  - Passwords, suplantación de identidad
- Los recursos de red están expuestos a usuarios desconocidos directamente por la vulnerabilidad del canal de transmisión
  - Ficheros y directorios
  - Instalación de programas de mal comportamiento
- Uso de la conexión para asuntos ilegales o para delinquir

### Contenido

introducción  
fundamentos  
tecnologías  
nombres  
tiempo  
**seguridad**  
coordinación  
transacciones

### Contramedidas → ataques wifi

1. Cambiar las opciones por defecto de routers y webs de configuración (no usando información personal en el SSID)
2. Actualizar firmware y hardware → objetivo WPA o WPA2
3. Apagar el AP cuando no se usa (o con algún *timer*)
4. Filtrado de MAC y número de clientes simultáneos
5. Bajar al mínimo útil la potencia de transmisión de AP
6. Encriptación WPA o WPA2 (con claves largas, no en diccionario y cambio periódico)
7. Encriptar los volúmenes – particiones y ficheros del sistema
8. Incorporar siempre antivirus, firewalls de dos direcciones (ver COMODO → Windows; iceFloor → Mac) y software anti-intrusión