Lección 12
Seguridad y criptografía

Seguridad

- Los sistemas distribuidos son más inseguros que los centralizados por que exponen más la información.
- Un sistema distribuido tiene más puntos atacables.
- Contrapartida: un sistema centralizado sólo hay que vencerlo una vez.
- En un sistema distribuido puede haber muchas “puertas”, pero todas se tienen que abrir (caso ideal).

Amenazas

- **Filtraciones**.- Adquisición de información por elementos no autorizados.
- **Falsificaciones**.- Alteración no autorizada de información.
- **Robo de recursos**.- Uso de recursos del sistema sin autorización.
- **Vandalismo**.- Interferencia en el correcto funcionamiento del sistema sin ganancia para el atacante.

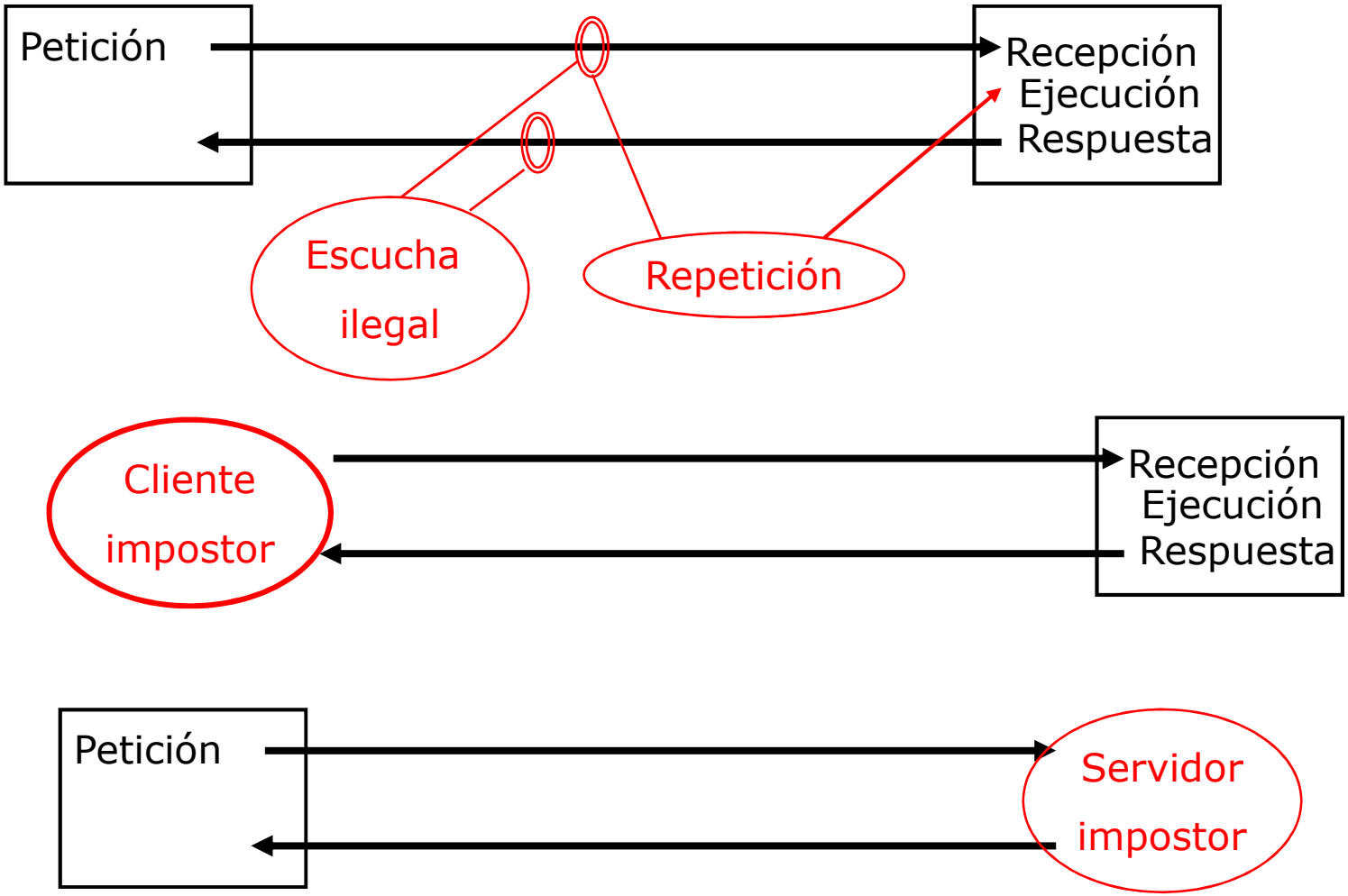
Terminología:

- **Principal**: Elemento (persona o programa) autorizado para acceder a la información o los recursos del sistema. Identificado por *nombre* y *clave.*

Ataques

- **Escuchas ilegales**.- Se puede realizar escuchando directamente la red o accediendo a almacenes mal protegidos.
- **Suplantación**.- Se envía y recibe información usando la identidad de un principal sin su autorización. Robo de identidad (nombre y clave) o bien usando elementos de identificación después de caducar.
- **Falsificación de mensajes**.- Interceptación de mensajes y alteración del contenido antes de reenviarlo.
- **Repeticiones**.- Almacenamiento de mensajes para enviarlos en fechas posteriores.

Escenarios


Petición
Recepción
Ejecución
Respuesta
Escucha
ilegal
Repetición
Cliente
impostor
Recepción
Ejecución
Respuesta
Petición
Servidor
impostor

Técnicas de seguridad

1. **Criptografía**.- Usos:
   1. Mantener la información privada a salvo de escuchas ilegales. Se supone que un mensaje encriptado con una clave sólo podrá ser desencriptado por el que conozca la clave inversa.
   2. Autentificación entre principales: si el receptor de un mensaje lo descodifica usando una determinada clave y encuentra un valor determinado, puede suponer que el emisor es el esperado.
   3. Firma digital.- Garantiza que una determinada información proviene de un determinada fuente.
2. **Autentificación**.- En sistemas centralizados: nombre + clave. En sistemas distribuidos: claves de encriptación. La autentificación la hace un ***servicio de autentificación***.
3. **Listas de control de acceso** (ACL).- Garantizan que unos determinados recursos sólo sean utilizados por los usuarios autorizados a hacerlo.

Criptografía

- Objetivo: Convertir *texto claro* en *texto cifrado* aplicando un regla. El receptor convierte el texto cifrado en texto claro aplicando la regla inversa.
- La encriptación y desencriptación se divide en dos partes:
  - La *función.*
  - La *clave.*
- La función define un algoritmo de encriptación dependiente de la clave.
- Notación:
  - ***M***: Texto claro.
  - ***K***: Clave.
  - $\{M\}_K$: Texto cifrado que se obtiene al aplicar a *M* la función de encriptación con la clave *K*

Criptografía. Tipos.

- La efectividad depende de la dificultad de descubrir *M* a partir de $\{M\}_K$ o de descubrir *K* a partir de *M* y $\{M\}_K$.
- Tipos de criptografía:
  - Basada en clave secreta.
  - Basada en claves públicas.

Criptografía de clave secreta

- Modelo más antiguo.
- Como la clave es secreta, la función de encriptación y su inversa no necesitan serlo.
- Emisor y receptor *DEBEN* compartir la misma clave. Problema de adquisición de la clave.
- Secuencia de comunicación:

  $f(K,M) = \{M\}_K \quad \rightarrow \quad f^{-1}(K,\{M\}_K) = M$
- Estándar inicial de clave secreta: DES (*Data Encryption Standard) [IBM 1997].*
  - *Convierte 64 bits de texto claro en 64 bits de texto cifrado usando una clave de 56 bits.*
- Debido a su inseguridad ha sido sustituido por el AES (*Advanced Encryption Standard*) en 2001. Utiliza claves de 128, 192 y 256 bits.

Ejemplos de aplicación

- La criptografía de clave secreta (o simétrica) se usa en muchos ámbitos, debido a su rapidez de ejecución y posibilidad de implementación en hardware
  - Cifrado de documentos o archivos para uso privado
  - Protecciones DRM
  - Cifrado de comunicaciones Wi-Fi (RC4)
  - Sistemas completos de autenticación y seguridad: Kerberos
  - Cifrado de canales seguros (SSH, SSL). En este caso se usa de forma mixta con la criptografía de clave pública.

Criptografía de clave pública

- Elimina el problema de tener que compartir una clave secreta entre los dos extremos de la comunicación.
- Se utiliza en sistemas de firma digital (FNMT, por ejemplo).
- Bases del sistema:
  - Existen dos funciones ***E*** y ***D***, conocidas, que se usan para encriptar y desencriptar los mensajes.
  - Existen dos claves, ***Ks*** y ***Kp*** para cada una de las funciones. ***Ks*** se mantiene secreta y ***Kp*** se publica.
  - La relación entre las claves ***f(Ks)=Kp*** es una función que, aunque se conoce, es muy costoso (computacionalmente) de obtener.
  - La relación entre ***D*** y ***E*** es de simetría: ***E(D(x))=D(E(x))=x***.

Criptografía de clave pública. Casos de Uso

Principales: Ana y Blas.

Ana genera Ks_A y Kp_A

publica Kp_A

Blas genera Ks_B y Kp_B

publica Kp_B

Envío de Mensaje

Computa $E(Kp_B, M) = \{M\}_{KpB}$ —Envío→ $D(Ks_B, \{M\}_{KpB}) = M$

¡Sólo lo lee Blas!

Envío de Mensaje Identificando al Emisor

$E(Ks_A, M) = \{M\}_{KsA}$

$E(Kp_B, \{M\}_{KsA}) = \{\{M\}_{KsA}\}_{KpB}$

—Envío→ $D(Ks_B, \{\{M\}_{KsA}\}_{KpB}) = \{M\}_{KsA}$

¡Sólo lo lee Blas!

$D(Kp_A, \{M\}_{KsA}) = M$

¡Lo envía Ana!

Criptografía de clave pública. Casos de Uso

Principales: Ana y Blas.

Ana genera Ks_A y Kp_A

publica Kp_A

Blas genera Ks_B y Kp_B

publica Kp_B

Firma Digital

M'=Mensaje (M) + Firma (F)

Computa ***Hash(M')= H***

$E(Ks_A,H)= \{H\}_{KsA}$

Publica (***M'+ $\{H\}_{KsA}$***)

Hash(M')= H1

$D(Kp_A, \{H\}_{KsA}) = H2$

Si H1=H2 mensaje de Ana y sin alterar.

También se podría encriptar $M'+\{H\}_{KsA}$

Clave pública. Algoritmo RSA

El RSA (Rivest, Shamir y Adelman) es el más utilizado. Pasos:

1. Se escogen P y Q números primos muy grandes (al menos 256 bits > 2^{100}).
   P=7 Q=13

2. Calcular *N=P*Q* y *Z=(P-1)*(Q-1)*
   N=91 Z=72

3. Se escoge Kp tal que sea un número primo relativo de Z. (Sin factores comunes con Z).
   Kp=5 ó 7 ó 11 ó 13 ...

4. Ks es el primer número que nos resuelve la ecuación:
   *Kp*Ks = 1 (mod Z)*.
   Kp*Ks=1 ó 73 ó 145,... Si Kp=5, Ks=?;

5. La funciones de cifrado y descifrado son:
   E(Ks,N,M) = M^{Ks} mod N y D(Kp,N,C)= C^{Kp} mod N
   E= M^{29} mod 91 D= C^{5} mod 91

6. *El texto se divide en bloques de k bits / 2^k < N*
   2^k < 91 ➔ k=6

Algoritmo RSA. Ejemplo

Tenemos:

P=7 **N=91** **Kp=5**

Q=13 **Z=72** **Ks=29** **k= 6 bits**

Mensaje:

ASCII

Esto es una prueba → 45 73 74 6F 20 65 73 20 75 …

Binario: 0100 0101 0111 0011 0111 0100 0110 1111 0010 0110 0101 ….

Agrupando de k en k bits

010001 010111 001101 110100 011011 110010 011001 01 ….

17 23 13 52 …. (están en decimal)

Cifrado:

$C = 17^{29} \bmod 91 = 75$

Veremos cómo obtenerlos

Decodificación:

$M = 75^{5} \bmod 91 = 17$

Cómo calcular el resultado

Tenemos que encontrar el resultado de $17^{29} \pmod{91}$

- Método "***bruto***": Calcular 17^{29} (sale 48196857210675091509141825223071697) y despues aplicarle mod 91.

- Método basado en la propiedad del módulo:

  $(a \times b) \bmod M = ((a \bmod M) \times (b \bmod m)) \bmod M$

  17^{29} es 17 * 17 * 17 ..... (29 veces). En cada multiplicación podemos aplicar el modulo 91 y asi nunca salen números demasiado grandes.

17^2	$17 \times 17 = 289 \equiv 16 \pmod{91}$
17^3	$16 \times 17 = 272 \equiv 90 \pmod{91}$
17^4	$90 \times 17 = 1530 \equiv 74 \pmod{91}$
...	*aun así tenemos 28 multiplicaciones y módulos*
17^{28}	$90 \times 17 = 1530 \equiv 74 \pmod{91}$
17^{29}	$74 \times 17 = 1274 \equiv 75 \pmod{91}$

Cómo calcular el resultado

- Algoritmo de **exponenciación binaria** (o "elevar al cuadrado y multiplicar"), para elevar un número a una potencia.
- Se quiere calcular x^N
  - Poner el exponente, *N*, en binario
  - Inicializar *aux:*=1
  - Para cada bit *b* de *N*, comenzando por la izquierda
    - Elevar al cuadrado *aux*
    - Si el bit *b* es 1, hacer además aux:=aux * *x*
    - Si el bit *b* es 0, no hacer nada extra
  - Al final, *aux* contiene el resultado buscado (x^N)

Cómo calcular el resultado

- ¿Por qué funciona el algoritmo?
- Ejemplo: 17^{29}

29=11101

Iteración	*aux* inicial	aux^2	bit *b*	nuevo *aux*
1	1	1	1	1*17
2	17	$(17)^2$	1	17^2*17
3	17^3	$(17^3)^2=17^6$	1	17^6*17
4	17^7	$(17^7)^2=17^{14}$	0	17^{14}
5	17^{14}	$(17^{14})^2=17^{28}$	1	17^{28}*17

17^{29}

- Este algoritmo llega a la solución con pocas operaciones, pero los datos que maneja se hacen enseguida enormes.

Algoritmo usado en la práctica

- Se usa la exponenciación binaria antes explicada, pero en cada asignación de *aux* se aplica el módulo.

Iter.	*aux* inicial	aux^2	mod 91	bit *b*	nuevo *aux*	mod 91
1	1	1	1	1	1*17	17
2	17	289	16	1	16*17 = 272	90
3	90	81000	1	1	1*17	17
4	17	298	16	0	16	16
5	16	256	74	1	74*17 = 1258	75

$17^{29} \pmod{91}$

Comparativa clave secreta vs clave pública

- **Seguridad**: Con los algoritmos y el manejo de claves adecuado, ambas técnicas son suficientemente seguras.
- **Comodidad**: La encriptación de clave pública es muy cómoda ya que no necesita compartir un secreto.
- **Rendimiento**: El cifrado de clave secreta es más eficiente que el de clave pública (estimado en 2 ó 3 órdenes de magnitud).

La encriptación de clave pública no cifra el mensaje.

Se utiliza para intercambiar la clave secreta con la que se cifra el mensaje.