Seguridad en redes de computadores

Programación y administración de redes - Semana 13

Grado en *Ingeniería Informática*

Departamento de Informática. Universidad de Jaén

Objetivos

General

Identificar los aspectos esenciales a considerar para conseguir una comunicación segura en una red de ordenadores y en Internet en general

Específicos

- Identificar las propiedades básicas de una **comunicación segura**
- Conocer los tipos de **potenciales ataques** que puede sufrir una red
- Analizar el uso de **técnicas criptográficas** en este contexto y diferenciar varias **técnicas de autenticación** y sus problemas
- Conocer el **funcionamiento de TLS** como respuesta a las demandas de comunicación segura
- Saber cómo es posible **garantizar la integridad** de los mensajes intercambiados por los ordenadores y **controlar el acceso** con cortafuegos

Comunicación segura

Definición

Llamamos comunicación segura a aquella que garantiza que nadie podrá acceder al contenido de los mensajes intercambiados, ya sea para examinarlo, alterarlo o impedir la propia comunicación

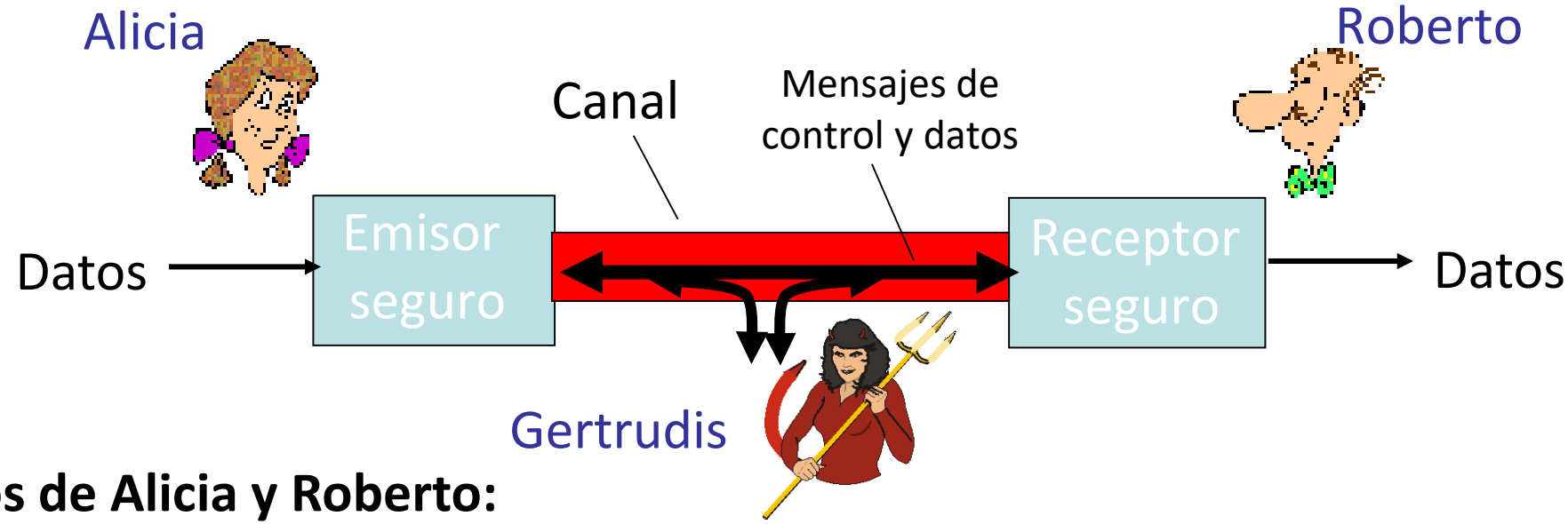
Características

- **Confidencialidad**: únicamente el emisor y el receptor deseado deben "entender" el contenido del mensaje.
  - Para ello el emisor cifra el mensaje y el receptor lo descifra
- **Autenticación**: emisor y receptor quieren confirmar la identidad de cada uno
- **Integridad del mensaje**: emisor y receptor quieren estar seguros de que el contenido de sus comunicaciones no es alterado (durante la transmisión o después) sin detección
- **Disponibilidad y acceso**: los servicios deben ser accesibles y deben estar disponibles para los usuarios. Se debe estar preparados ante ataques de denegación del servicio

Comunicación segura

Contexto de trabajo

Tenemos un emisor y un receptor que se comunican a través de un canal que, en general, no es seguro



- **Ejemplos de Alicia y Roberto:**
  - Navegador y servidor web para transacciones electrónicas: compras por Internet
  - Servidores DNS y peticiones de los clientes
  - Router que intercambian actualizaciones de tablas de encaminamiento

Comunicación segura

Al emplear canales de comunicación accesibles públicamente surgen múltiples tipos de ataques por parte de l@s chic@s mal@s

Problemas

- **Escuchar a escondidas**: interceptar mensajes para acceder a su contenido
- **Insertar** activamente mensajes en la conexión para engañar a las partes
- **Suplantación**: puede falsear la dirección fuente en el paquete (o cualquier campo en el paquete)
- **Secuestro**: "apoderarse" de la conexión entrante eliminando al receptor o al emisor e insertándose en su lugar
- **Denegación del servicio**: impedir que el servicio sea utilizado por otros, por ejemplo sobrecargando los recursos

Soluciones

- **Criptografía**: cifrado del contenido de los mensajes
- **Autenticación**: identificación segura de las partes que actúan en la comunicación
- **Integridad**: verificación de que el contenido de los mensajes no ha sido alterado
- **Control de acceso**: impedir el acceso a actores no deseados
- **Otras contramedidas**: técnicas alternativas para el aseguramiento de la comunicación entre las partes y la disponibilidad de servicios

Cifrado de los datos
Criptografía

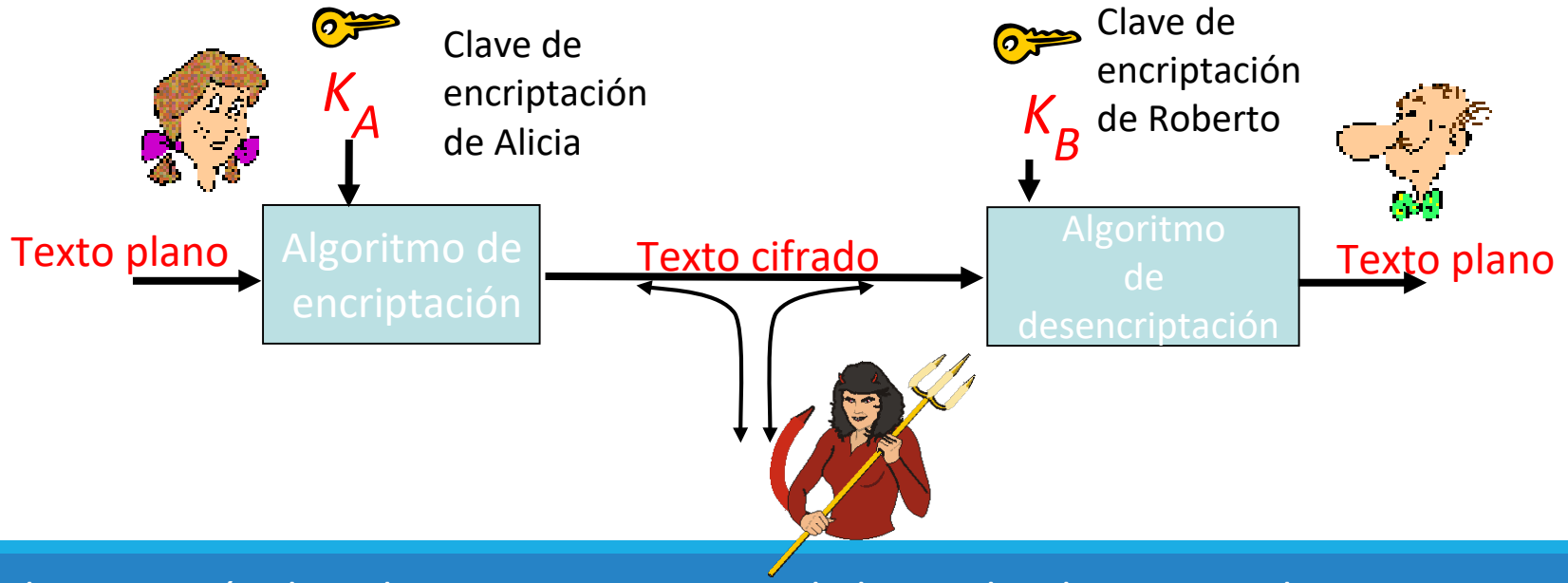
Cifrado de los datos - Introducción

Definición

El uso de técnicas criptográficas permite ocultar un mensaje de forma que los intrusos no puedan acceder a su contenido

Tipos

- Criptografía de clave simétrica: las claves de cifrado/descifrado del emisor y receptor son *idénticas*. La clave solo es conocida por ellos. Ejemplos de algorimos de cifrado simétrico: DES, AES, …
- Criptografía de clave pública: encriptación con clave *pública*, desencriptación con clave *privada* (secreta)



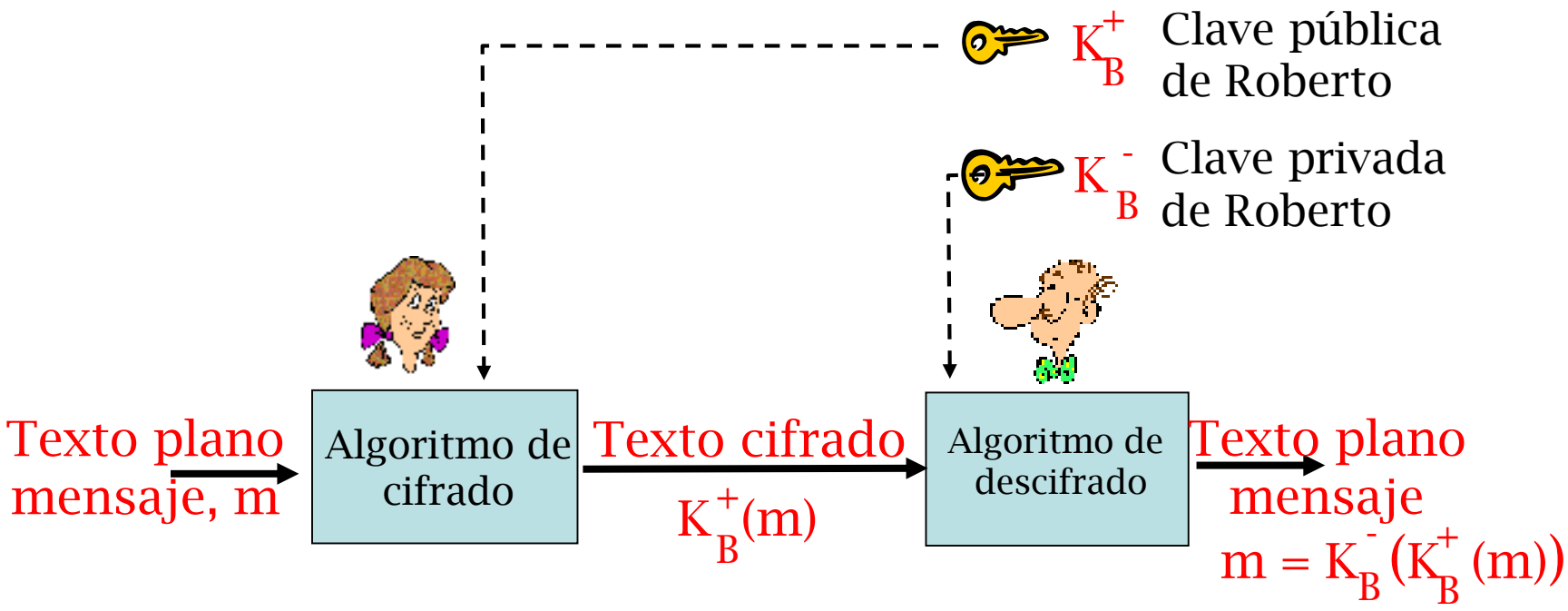
Estos conceptos los conocéis de la asignatura **Seguridad en tecnologías de la información** del primer cuatrimestre

Cifrado de los datos - Criptografía de clave pública

- Enfoque distinto al cifrado simétrico, ya que ahora emisor y receptor **no comparten clave** secreta. Ejemplos: RSA, DSA, …
- Clave de cifrado *pública* conocida por *todos*
- Clave de descifrado *privada*, conocida solo por el receptor

Debe ser imposible obtener la clave privada a partir de la clave pública

- Propiedad RSA $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$



Identificación de los interlocutores

Autenticación

Protocolo de autenticación - Introducción

Definición

Denominamos autenticación al proceso por el cual se demuestra que un ***interlocutor es quien realmente dice*** *ser*

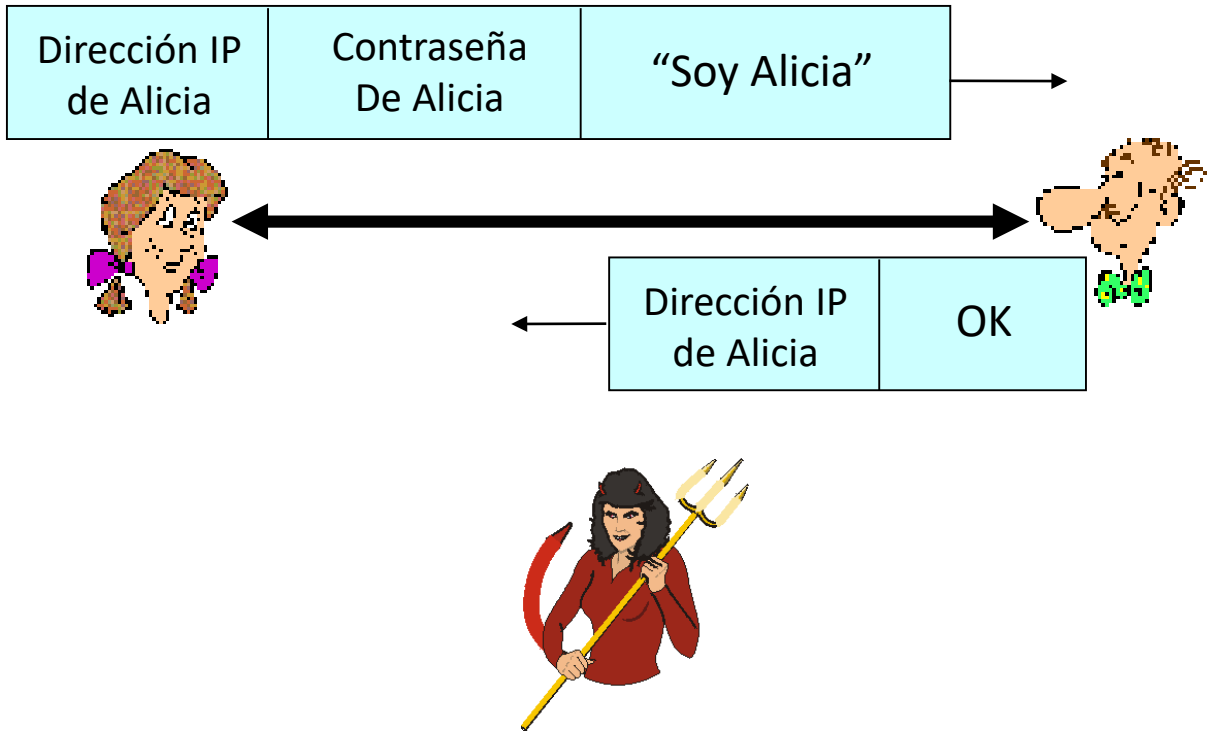
Características

- Para esto se utiliza lo que se llama un **protocolo de autenticación** donde se intercambiarán unos mensajes entre los interlocutores
- Este proceso debería de **realizarse antes de intercambiar** cualquier información que se quiera proteger
- Partiendo de un protocolo de autenticación (**pa**) básico, identificaremos sus debilidades y lo iremos mejorando paso a paso

Protocolo de autenticación - pa3.0

Funcionamiento

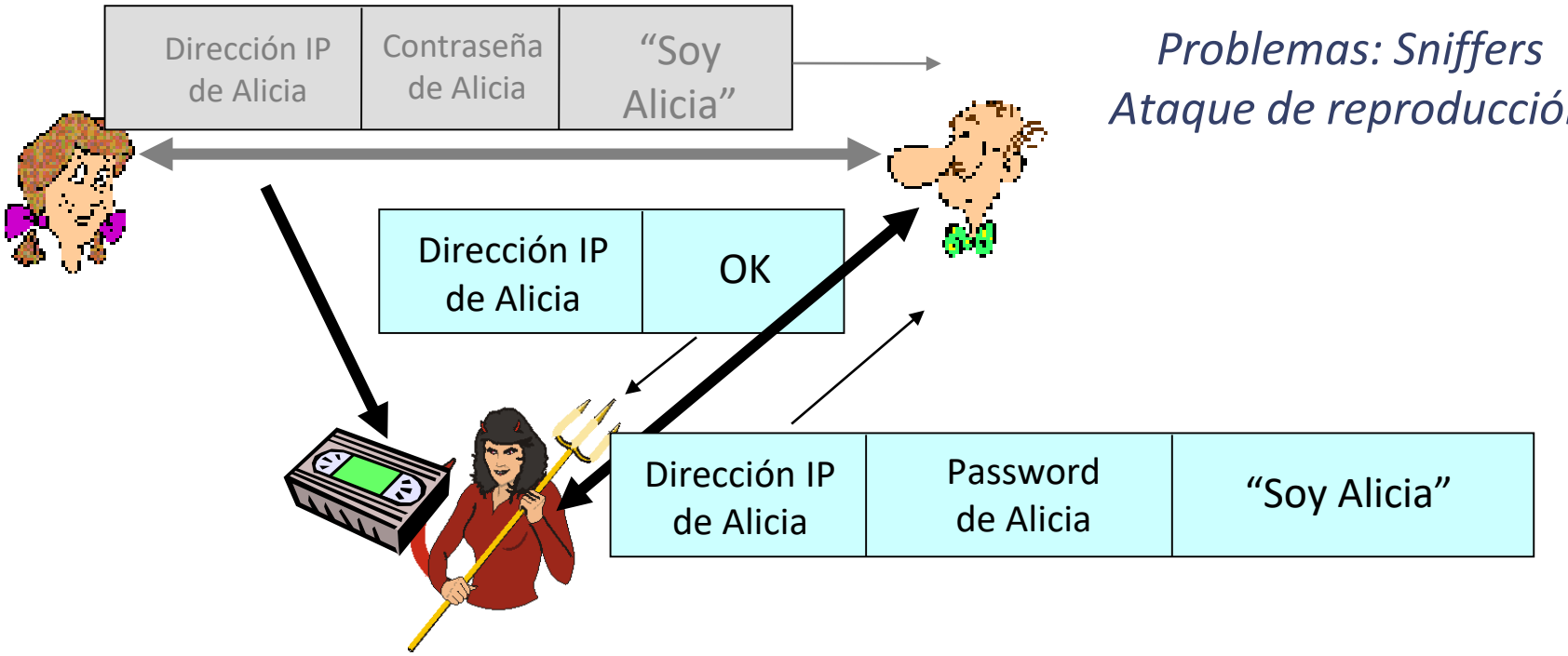
El interlocutor dice quién es y envía una contraseña



Protocolo de autenticación - pa3.0

Escenario de fallo: ataque de reproducción

Gertrudis graba el paquete de Alicia, usando un *sniffer*, y más tarde se lo reproduce a Roberto

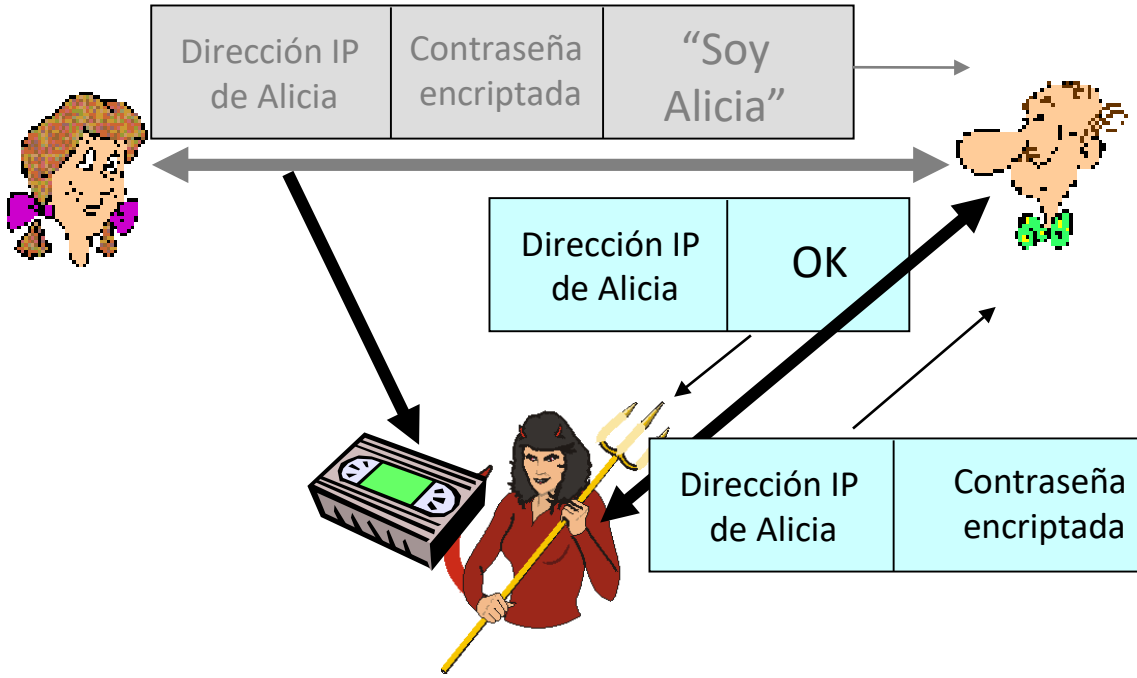


Problemas: Sniffers
Ataque de reproducción

Protocolo de autenticación - pa3.1

Mejora: cifrado de la contraseña

Persiste el mismo problema, la contraseña puede grabarse y volver a enviarse aunque esté cifrada

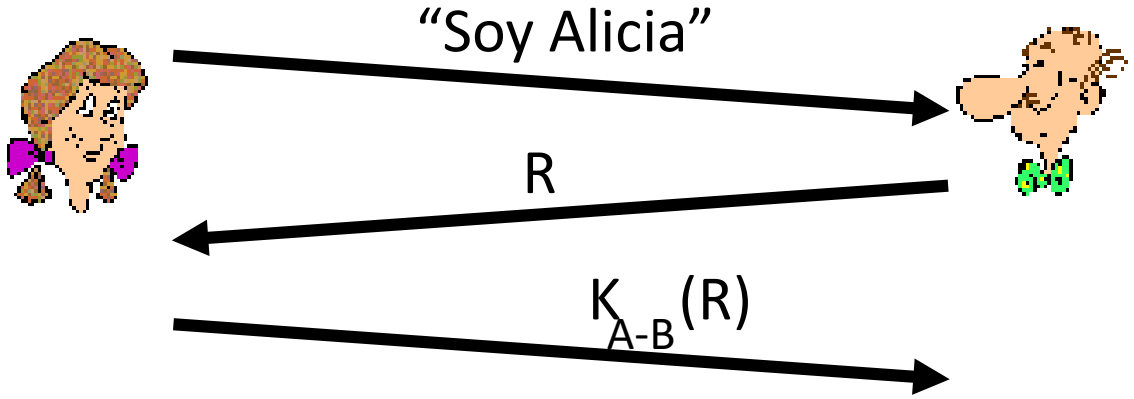


¡Grabar y reproducir sigue funcionando!
Ataques de diccionario …

Protocolo de autenticación - pa4.0

Objetivo: evitar el ataque de reproducción

- Se usa un número (R en el diagrama inferior) al que llamamos *núnico* ya que **se usa una sola vez** durante la vida del protocolo
- Para probar "en directo" que Alicia es quien dice ser, Roberto le envía un **núnico R**. Alicia debe devolver R cifrado con una **clave secreta compartida** entre ambos interlocutores



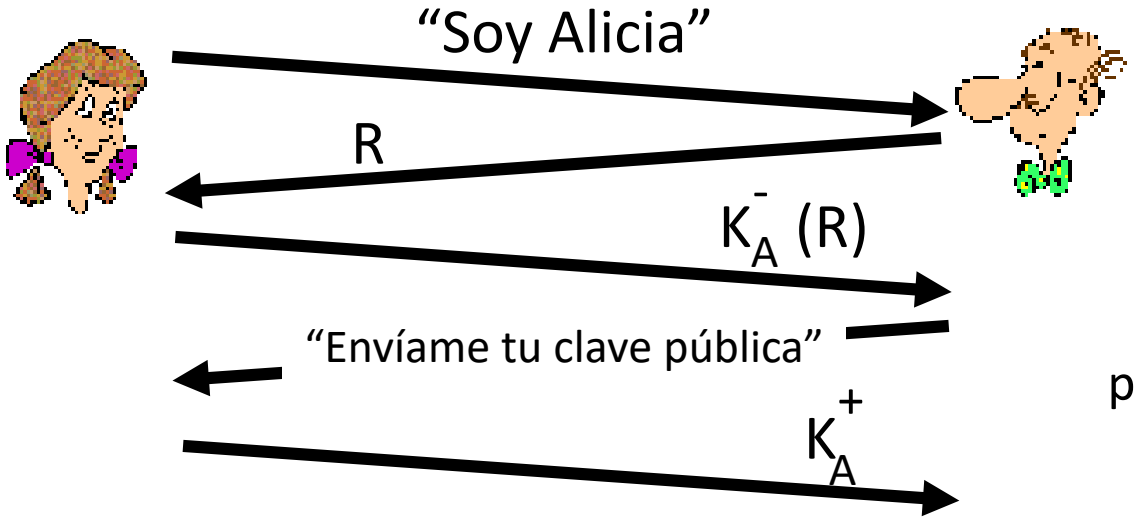
Alicia está en directo, y solo ella conoce la clave para encriptar el *núnico*, así que este protocolo ya sí es seguro

- **Problema**: es necesario llevar la clave compartida de una máquina a otra, al menos inicialmente antes de poder establecer la comunicación, lo cual implica cierta inseguridad

Protocolo de autenticación - pa5.0

Objetivo: eliminar la necesidad de compartir una clave

- En esta versión pa5.0 se sigue usando el núnico, pero en este caso cifrándolo con una **clave pública** de forma que únicamente el receptor, que tiene la clave privada, pueda descifrarlo



Roberto calcula

$K_A^+(K_A^-(R)) = R$

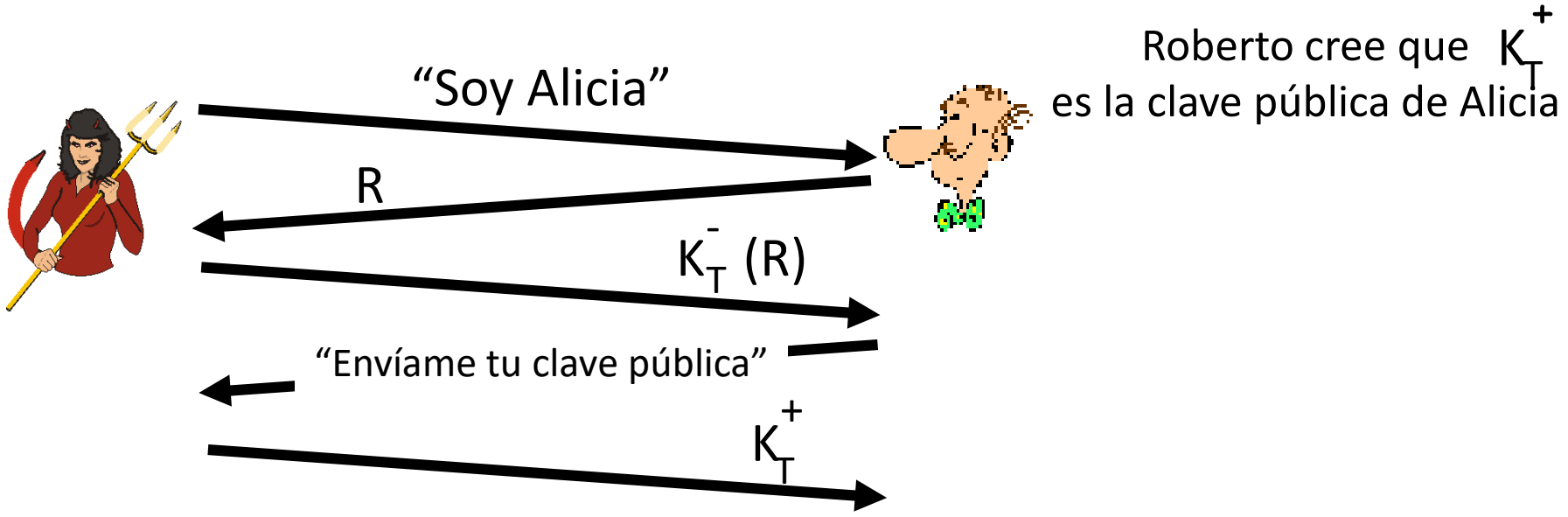
y sabe que sólo Alicia puede tener la clave privada que encripta a R,

- **Problema**: ¿quién asegura que quien envía la clave pública a Roberto es realmente Alicia?

Protocolo de autenticación - pa5.0

Ataque: Gertrudis se hace pasar por Alicia

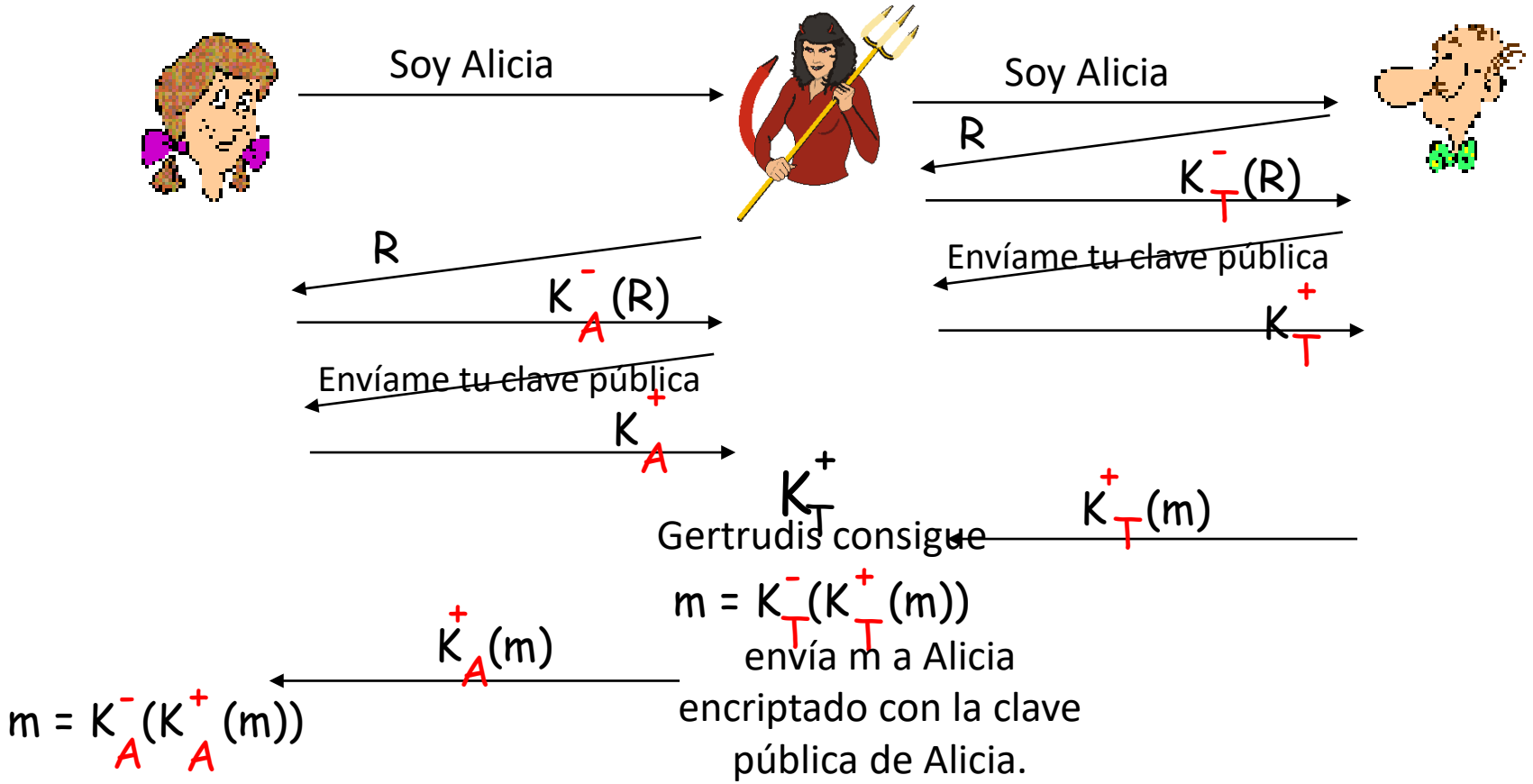
- Este problema (y el de la siguiente diapositiva) se podría solucionar mediante algún mecanismo que nos permitiera garantizar que la clave pública de alguien es realmente de ese alguien



Protocolo de autenticación - pa5.0

Ataque: "Man in the middle", interceptar la comunicación

- Getrudis se interpone y actúa como Alicia de cara a Roberto y como Roberto de cara a Alicia



Autenticación segura
Autoridades de certificación

Autoridades de certificación

Problema

*La criptografía basada en clave pública tiene un problema fundamental: cuando se obtiene la clave pública de Roberto (de un sitio web, por correo electrónico u otro medio), ¿**cómo puede saberse** que es realmente la clave pública de Roberto y no la de Gertrudis?*

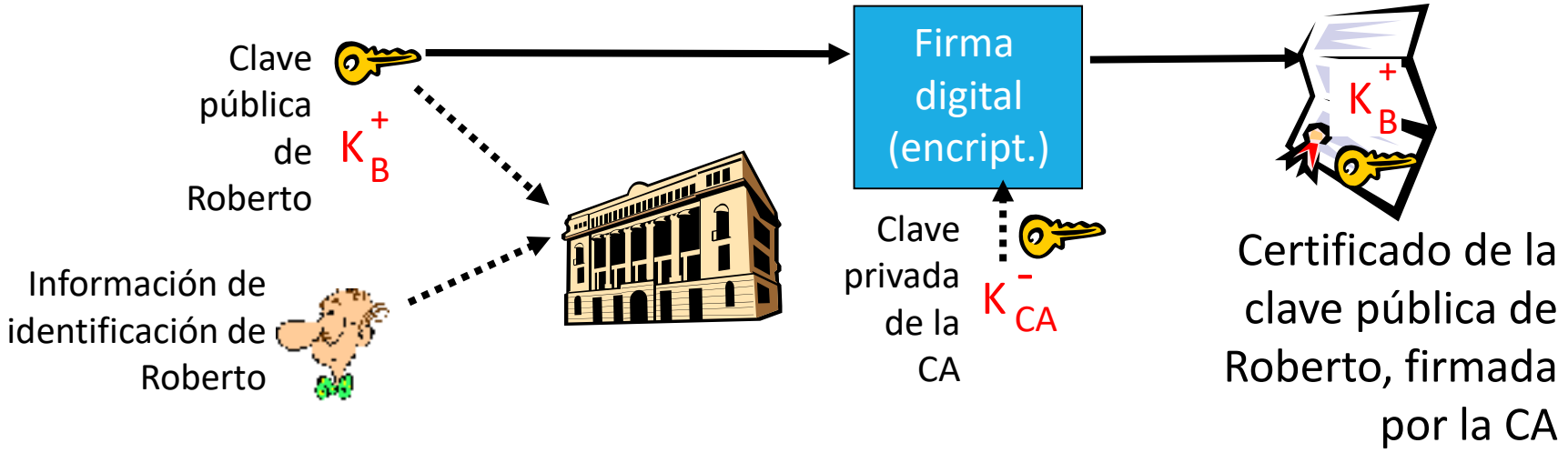
Solución: autoridad de certificación de confianza (CA)

- Las CA (*Certification Authority*) son organizaciones públicas y empresas
- Una CA **vincula la clave** pública a una entidad en particular
- Hay muchas CA actualmente, habría que **elegir una de confianza**
- En nuestro país una de las CA más habituales es la FMNT

Autoridades de certificación

Funcionamiento

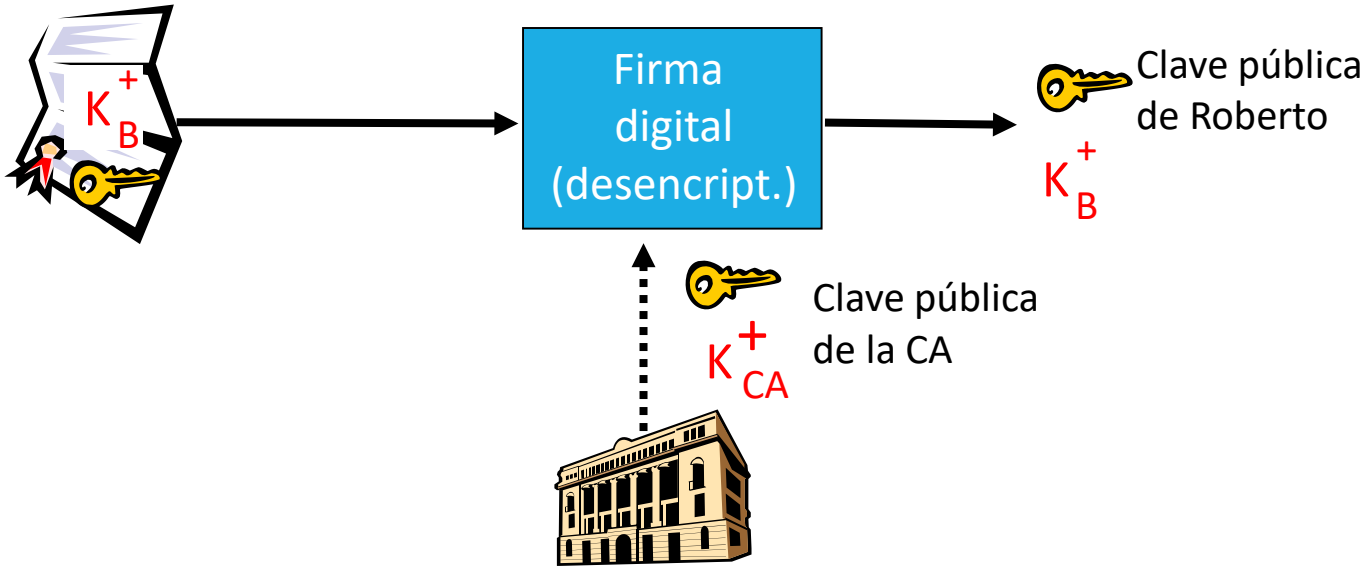
- La CA vincula una clave pública a una entidad **E** (persona, empresa, servidor) que previamente la ha registrado:
  - **E** proporciona "prueba de identidad" a la CA
  - La CA crea certificado que vincula a **E** con su clave pública
  - Se genera un certificado que contiene la clave pública de **E** firmada digitalmente por la CA. Con él, la CA dice "Esta es la clave pública de E"



Autoridades de certificación

Funcionamiento

- Cuando Alicia quiere la clave pública de Roberto para cifrar el núnico:
  - Obtiene el certificado digital de Roberto, ya sea del propio Roberto o cualquier otra fuente
  - Aplica al certificado de Roberto la clave pública de la CA y obtiene la clave pública de Roberto



  - Los certificados de las CA raíz, con su clave pública, se encuentran generalmente preinstalados en la mayoría de sistemas

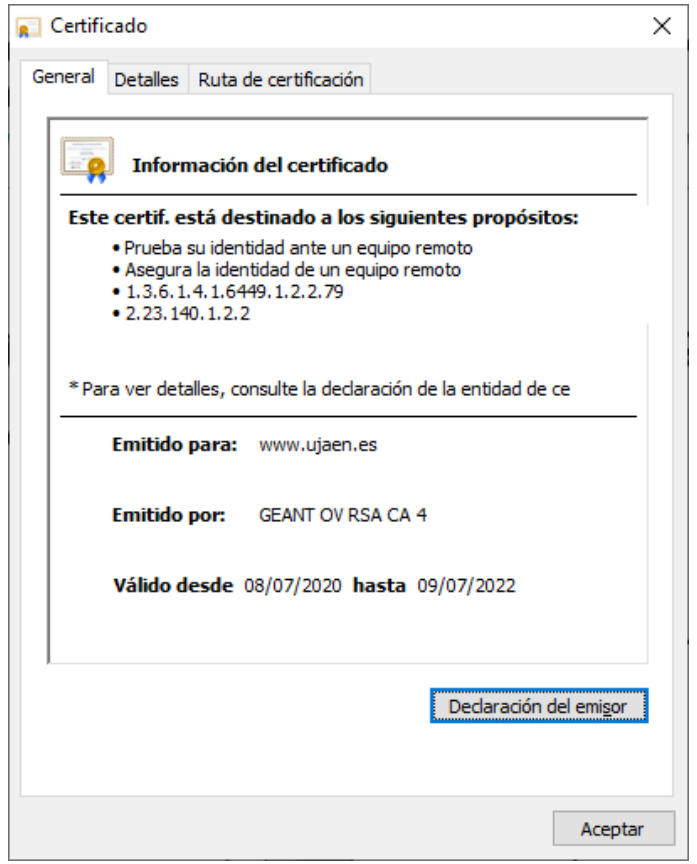
Actividad - Certificados HTTPS

Objetivo

Comprobar el uso de certificados digitales y el papel de las autoridades de certificación en la comunicación entre clientes y servidores web

Descripción

- Usa tu navegador web para conectar con el servidor web www.ujaen.es
- Haz clic en el icono en forma de candado que aparece a la izquierda del URL
- Selecciona la opción que te permite examinar el certificado y contesta a las **siguientes cuestiones**:
  - ¿Qué **autoridad de certificación** garantiza que este certificado contiene la clave pública del servidor web de la Universidad de Jaén?
  - ¿Hasta **cuándo es válido** este certificado digital?
  - ¿Quién certifica que la anterior autoridad de certificación **es quién dice** ser en realidad?
    - Consulta los certificados de la ruta de certificación hasta la raíz



Integridad de los mensajes

Firma digital

Firma digital - Introducción

Objetivo

Ofrecer un método que permita verificar que los datos proceden de una fuente fiable (persona, organismo, …) y que no han sido modificados por el camino

Funcionamiento

- Emisor (Roberto): firma digitalmente un mensaje o documento estableciendo que es el creador o propietario e incorporando la información necesaria para detectar alteraciones

Características

- Verificable: se puede probar que corresponde a una persona/entidad sin ambigüedad
- No falsificable: nadie puede reproducir la firma sin el certificado original
- No repudiable: el emisor no podrá decir que no es su firma

Algoritmos de resumen - Conceptos

Definición

Un algoritmo de resumen (hashing) genera una huella digital única y de longitud fija a partir de un mensaje de longitud arbitraria

Finalidad

- Los **algoritmos de resumen** del mensaje se emplean para aportar integridad
- En un **algoritmo de resumen del mensaje**:
  - A partir de un mensaje `m` de longitud arbitraria se calcula un resumen o *huella digital* de longitud fija `H(m)`
  - A `H(·)` se le denomina función de dispersión
- A priori algunas funciones como (sumas de comprobación, métodos CRC, etc.) podrían definirse como funciones de dispersión
- Para lograr la integridad una característica que debe cumplir `H` es que "sea **imposible encontrar dos mensajes** distintos `m` y `n` tales que `H(m) = H(n)`", propiedad que hay que verificar si cumplen las funciones anteriores
- Esto implica que cuando un emisor envía `(m, H(m))` a un receptor, es imposible sustituir un mensaje por otro

Algoritmos de resumen - Funciones de dispersión

- MD5 (*Message Digest Algorithm*, RFC 1321):
  - Calcula un resumen de mensaje de **128 bits** en un proceso de cuatro pasos
  - Cadena x arbitraria 128-bit, parece difícil construir mensaje m cuya dispersión MD5 sea igual a x

- SHA-1 (*Secure Hash Algorithm*):
  - Estándar de EE.UU. [NIST, FIPS PUB 180-1]
  - Resumen de mensaje de **160 bits**

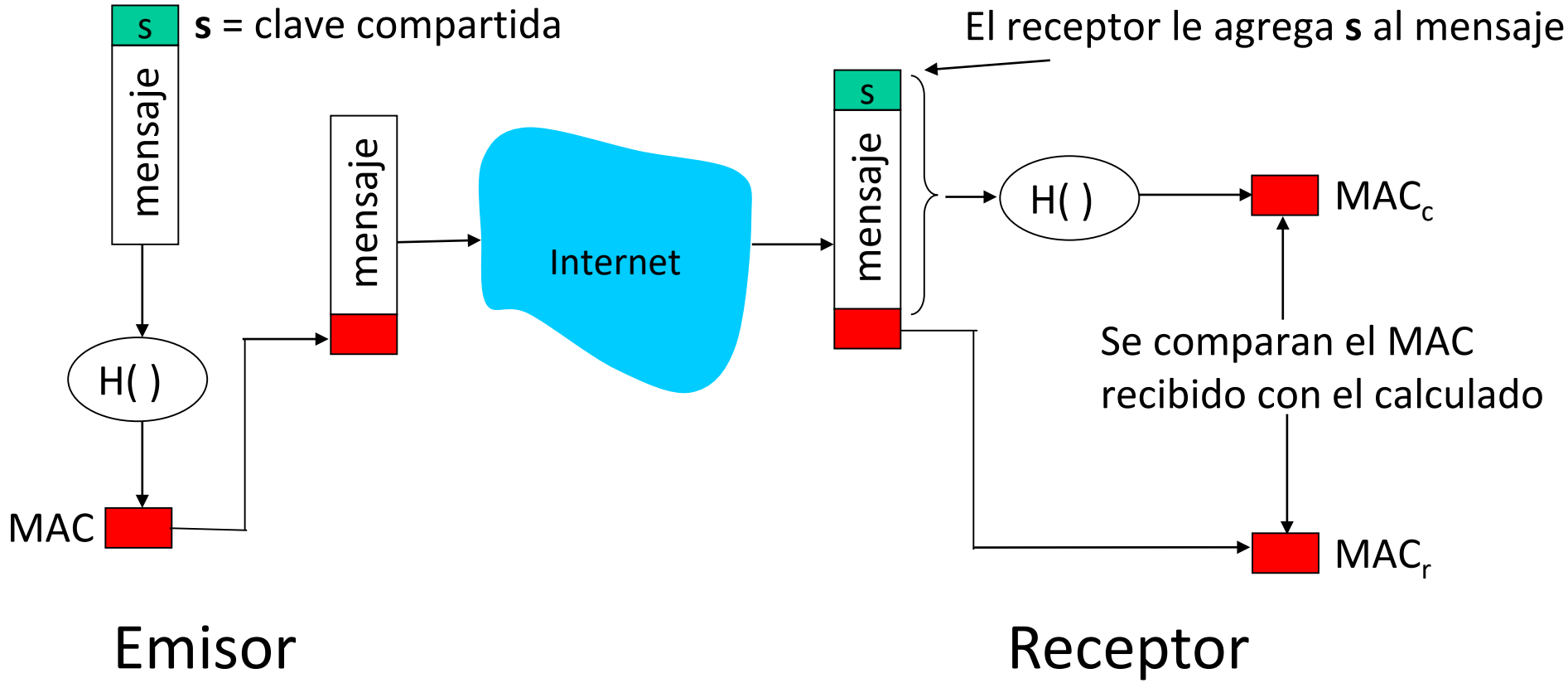
- SHA-2 y SHA-3:
  - Tanto MD5 como SHA-1 son vulnerables a distintos tipos de ataques
  - Las nuevas versiones de SHA generan resúmenes de entre **224 y 512 bits**
  - Algoritmo de mayor complejidad y más difícil de romper mediante ataques

Código de autenticación de mensaje (MAC)

- El problema es que aunque no se pueda sustituir un mensaje por otro para la tupla `(m, H(m))`, **sí se podría sustituir completamente** por otra:
  - `(m', H(m'))` → Sustitución del mensaje y de su resumen
- Para asegurar la integridad del mensaje hace falta una **clave secreta** compartida (*shared secret*) `s` o clave de autentificación
- Esta clave (que al igual que antes es una cadena de bits) se añadirá de la siguiente manera:
  - `(m, H(m+s))` → No es posible la sustitución sin conocer `s`

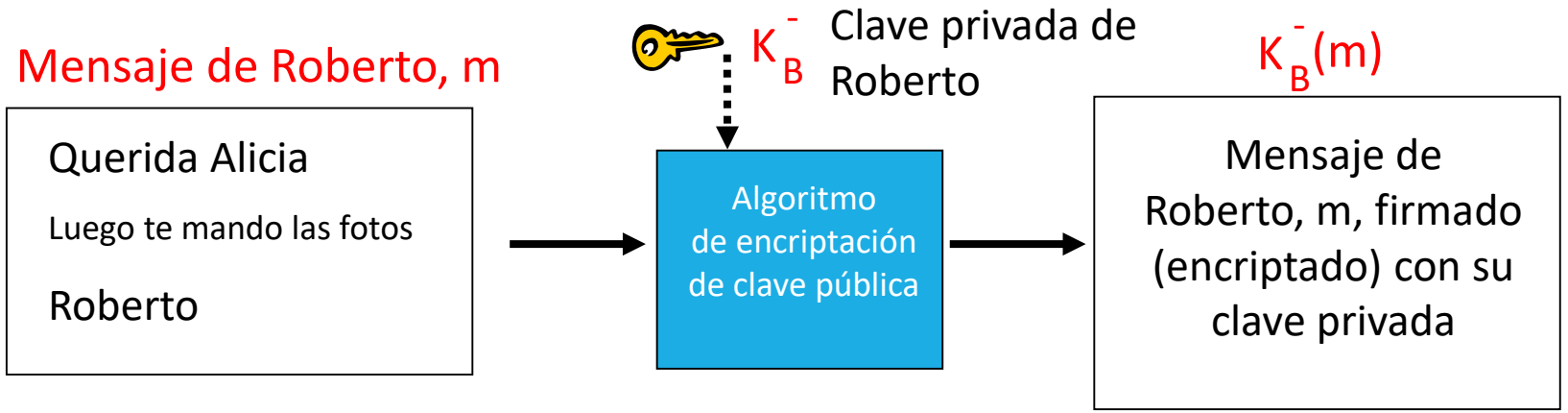
Código de autenticación de mensaje (MAC)

- Emisor y receptor agregan al mensaje la misma clave **s** y usan la misma función H( ) para generar el MAC
- El emisor envía el mensaje y su MAC, el receptor calcula su MAC y verifica que coinciden



Firma digital - Funcionamiento

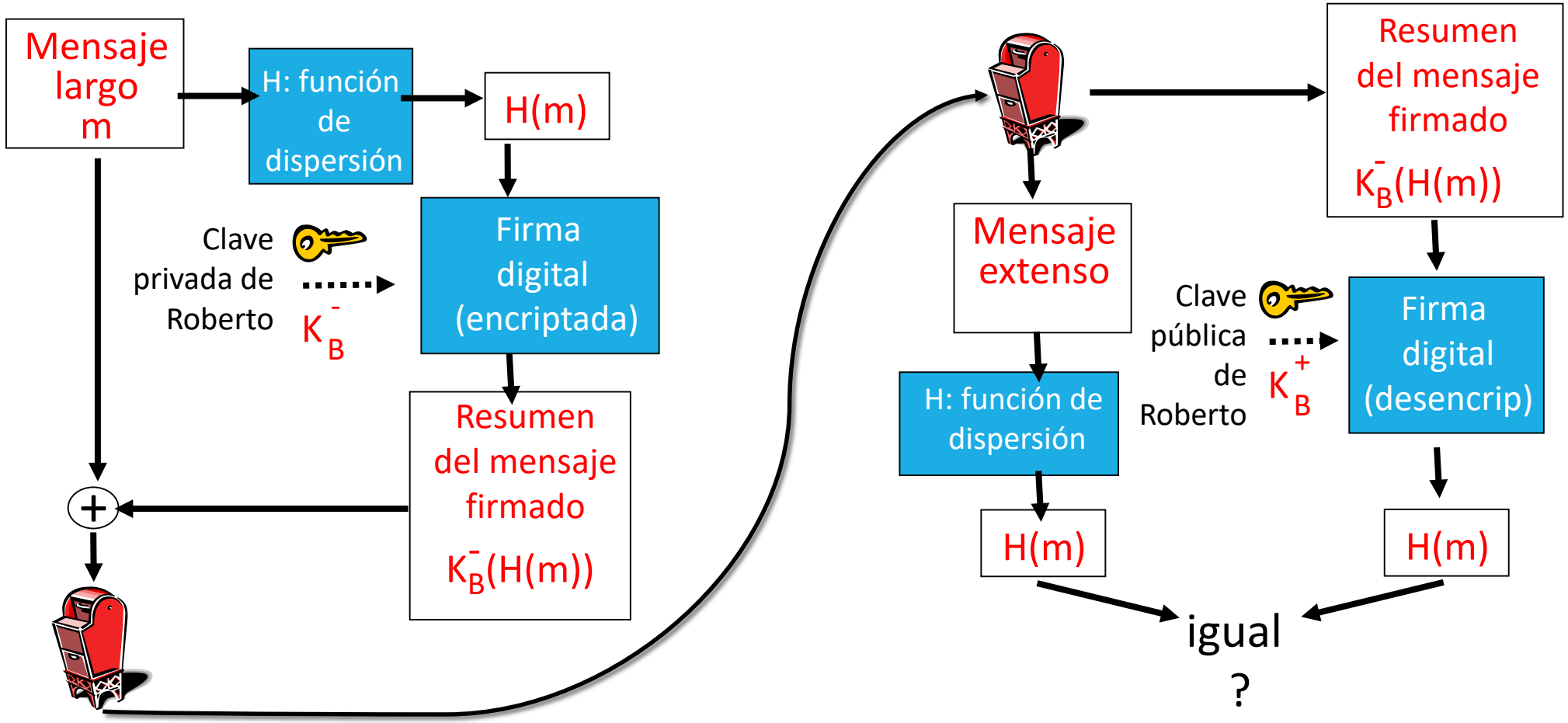
- Mecanismo que permita **firmar digitalmente** un documento. Esta firma debe ser verificable, no falsificable y no repudiable. Ejemplo con clave pública:



- Alicia verifica m firmado por Roberto aplicando la clave pública de Roberto K+B y comprueba que $K^+_B(K^-_B(m)) = m$
- Por las características de una clave pública se cumple:
  - Se puede verificar que Roberto ha firmado m
  - Nadie más (con otra clave) ha firmado m
  - Roberto ha firmado m y no m'
  - Roberto no puede negar que ha firmado m

Firma digital - Funcionamiento

- La opción anterior es costosa computacionalmente. Es mejor firmar solo el resumen del mensaje. Ejemplo:



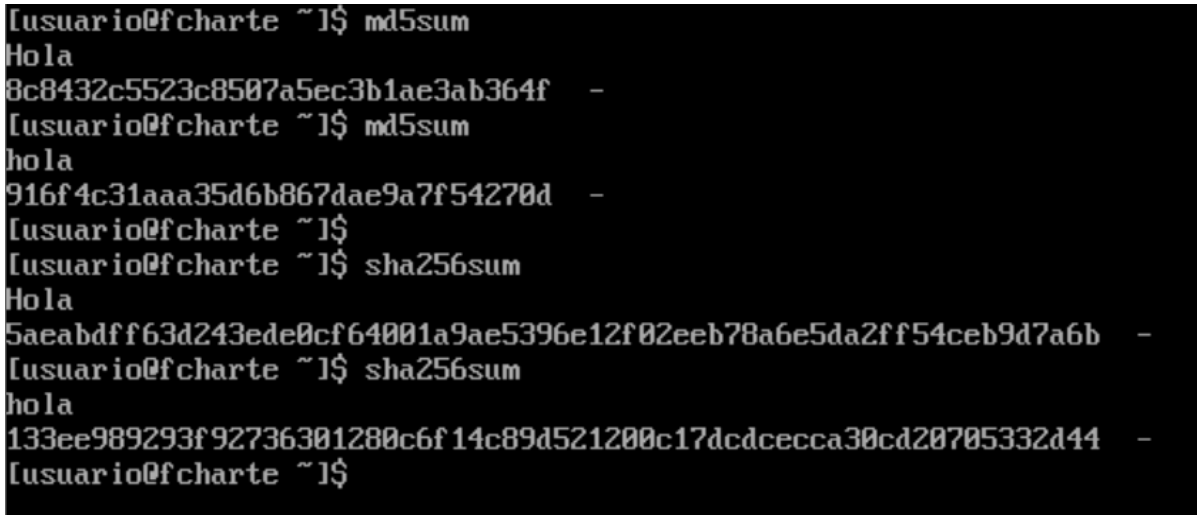
Actividad - Vulnerabilidades de los algoritmos

Objetivo

Verificar que los algoritmos, en este caso los de resumen, pueden ser vulnerables a ataques y los peligros que ello conlleva

Descripción

- En tu sistema tienes instalados los programas `md5sum` y `sha256sum`, cuya finalidad es obtener el resumen de cualquier mensaje facilitado como entrada. Puedes ejecutarlos, introducir un mensaje y pulsar **Control+D**
- Comprueba, como se hace en la imagen adjunta, cómo el **cambio de un solo carácter** genera un resumen totalmente diferente
- En www.mscs.dal.ca/~selinger/md5collision encontrarás **dos cadenas de caracteres** que hacen que MD5 genere exactamente el mismo resumen
- La misma web facilita **dos programas**, con una finalidad totalmente distinta, para los que MD5 también genera el mismo resumen
- ¿Dirías que MD5 es un algoritmo de *hash* fiable?



Servicio de transporte seguro
TLS

Comunicación segura - Cómo aunar todo

Objetivo

Contar con un mecanismo que ofrezca simultáneamente confidencialidad en la transmisión, autenticación de los interlocutores e integridad de los mensajes. Es lo que nos ofrece TLS (Transport Layer Security)

Además ...

El cifrado, autenticación e integridad no serán suficientes ante otros tipos de ataques, por lo que será necesario:

- **Controlar el acceso**: permitiendo acceder a nuestra red únicamente a quien nos interesa
- **Garantizar la disponibilidad**: el servicio debería estar disponible siempre que los interlocutores lo necesiten
- **Preveer medidas de contingencia**: prepararse para lo inesperado con planes de recuperación

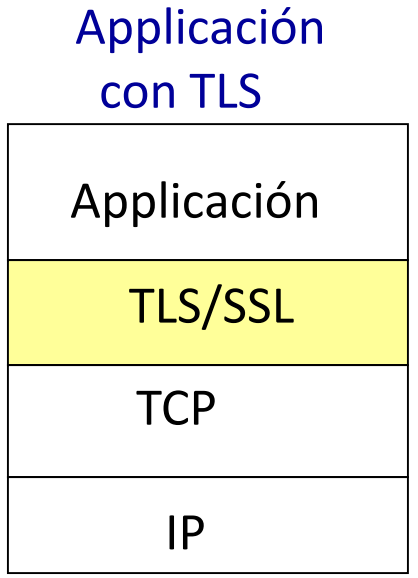
Servicio de transporte seguro - TLS

Definición

TLS actúa como una capa adicional de seguridad que se dispone entre la capa de transporte y la de aplicación

Características

- **Finalidad**: ofrecer mecanismos de autenticación, cifrado e integridad
- **Uso**: solución ampliamente usada, p.e. con HTTPS
- **Origen**: TLS es una versión moderna y abierta de SSL (*Secure Sockets Layer*), diseñado por Netscape en 1993
- **Disponibilidad**: TLS actúa como una API (*Application Program Interface*) accessible desde diferentes lenguajes de programación



A continuación se describe de forma resumida el funcionamiento de TLS

TLS - Fases de funcionamiento

Resumen

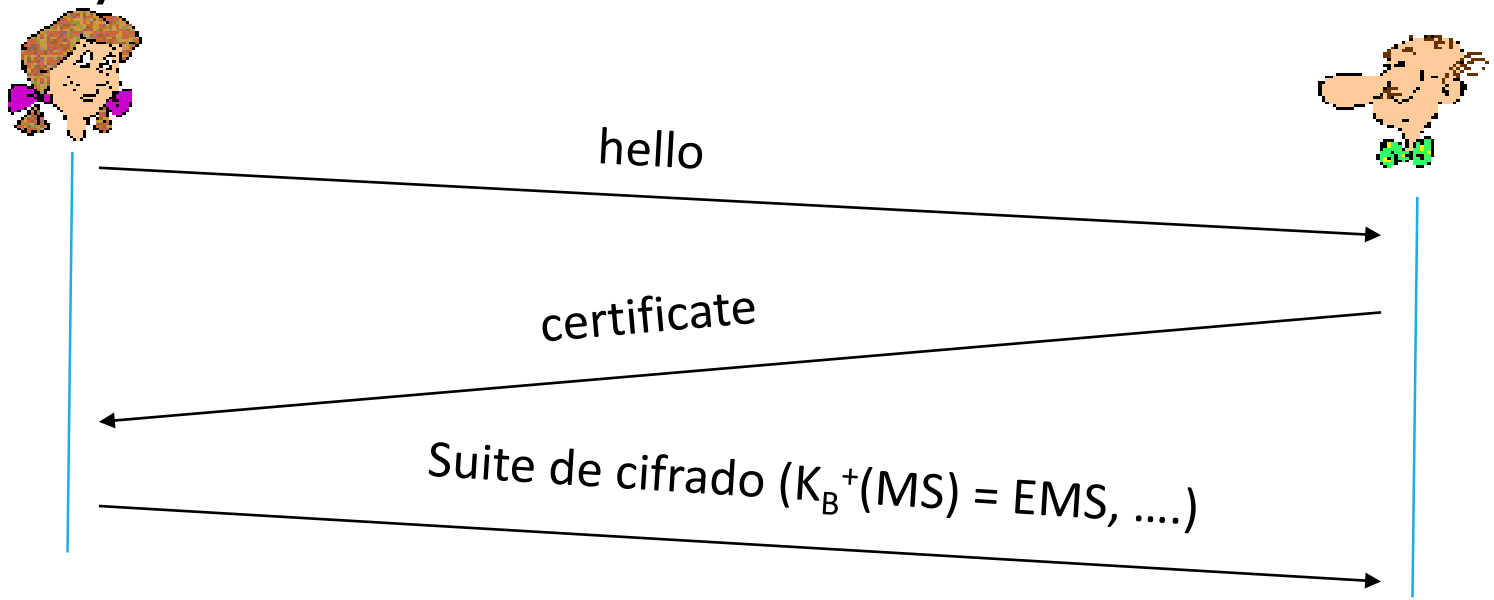
El funcionamiento de TLS implica unas fases previas a la comunicación propiamente dicha y una fase posterior

Fases

- ***Handshake***: establecimiento de conexión, autenticación, negociación sobre algoritmos y claves a usar
- **Derivación de claves**: a partir de una clave compartida se generan las claves definitivas a usar
  - Para aumentar la seguridad se generan claves derivadas de MS tanto para encriptación como para integridad
- **Transferencia de datos**: envío de datos encriptados y divididos en registros
- **Cierre de la conexión**: con mensajes especiales que aseguren un cierre seguro de la conexión

TLS - Fase de *handshake* (simplificado)

Finalidad: intercambiar información para acordar los algoritmos de cifrado que se emplearán y la clave de cifrado simétrico

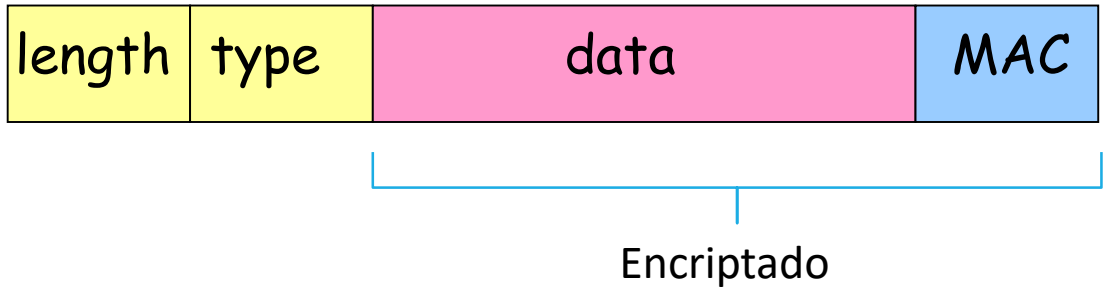


- **Suite de cifrado** que contiene algoritmos de cifrado a utilizar, claves, etc.
- **MS** – *Master Secret*: clave simétrica a partir de la que derivaremos las claves que usaremos en la transmisión
- **EMS**: *Encripted Master Secret*

TLS - Registros de datos (simplificado)

Finalidad: facilitar la comprobación de integridad de cada paquete de datos intercambiado entre los interlocutores

- Necesidad de dividir los datos a enviar en registros/paquetes con una estructura interna
- De esta manera se permite hacer comprobaciones inmediatas de integridad de cada paquete
- **Campo MAC** - código de autenticación del mensaje
- **Registro TLS**:



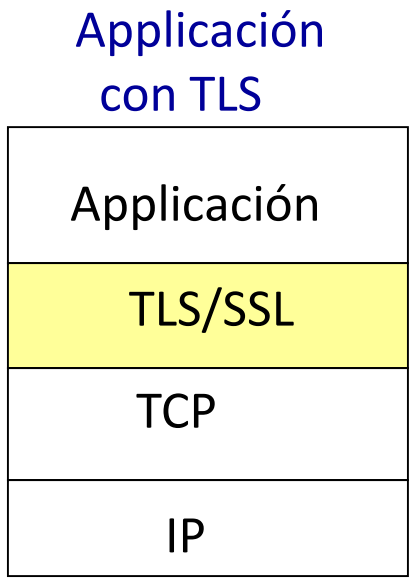
TLS - Números de secuencia

Problema

TLS no define un mecanismo de seguridad sobre los segmentos TCP en sí mismos, por lo que sus cabeceras podrían sufrir ataques

Ataques y soluciones

- En un primer ataque se podrían desordenar, suprimir, etc. estos segmentos
  - **Solución**: introducir un número de secuencia incluido en el código MAC
- Otro posible ataque (*truncation attack*) consistiría en **terminar la conexión TCP** enviando un segmento de cierre de conexión
  - **Solución**: introducción del campo `Type` que indique el tipo de registro que es: de datos o de cierre



TLS - Ejemplo de funcionamiento (simplificado)

hello

certificate

Handshake

Cipher Suite

type 0, seq 1, data

type 0, seq 2, data

type 0, seq 1, data

Intercambio de datos

Comunicación cifrada

type 0, seq 3, data

type 1, seq 4, close

Cierre

type 1, seq 2, close

Control de acceso
Cortafuegos

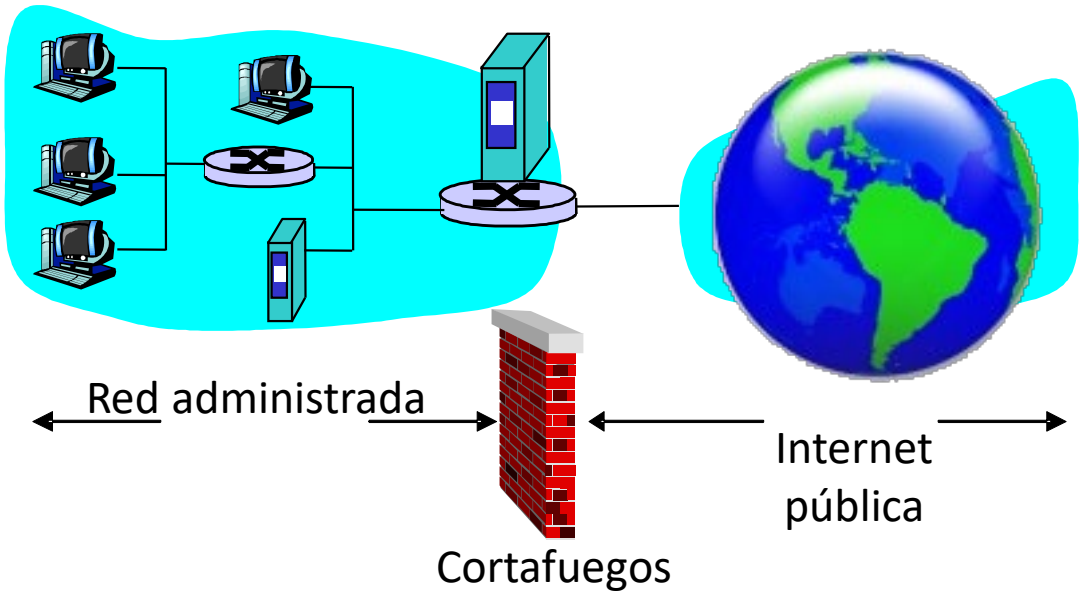
Cortafuegos - Introducción

Definición

Un cortafuegos es un componente hardware o software que tiene por finalidad controlar el tráfico entre una red privada e Internet

Características

- Normalmente la entrada/salida a una red **se produce por un punto**, el *gateway*, en el que se colocaría el cortafuegos
- En entornos domésticos y PYME el *router* que conecta la LAN a Internet suele incorporar un cortafuegos
- Un cortafuegos puede ser un también un dispositivo hardware con esa única función
- Existen **dos tipos** de cortafuegos:
  - De nivel de aplicación
  - De filtrado de paquetes

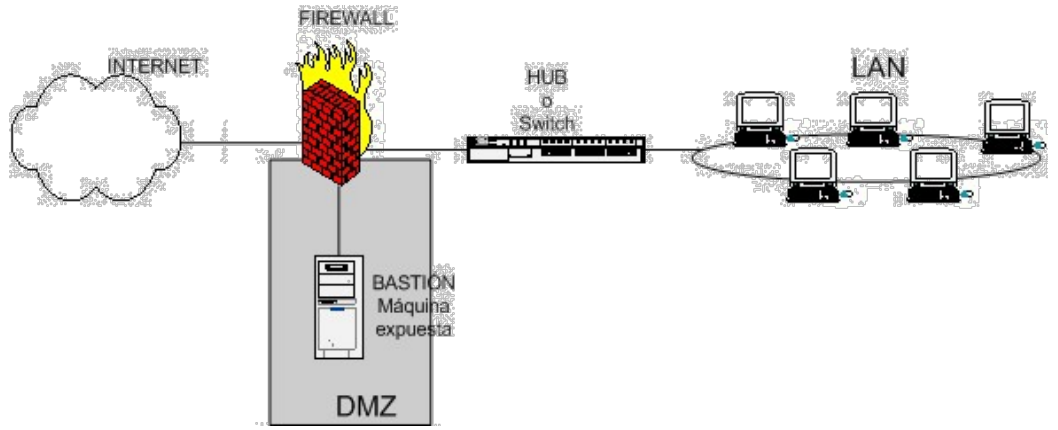


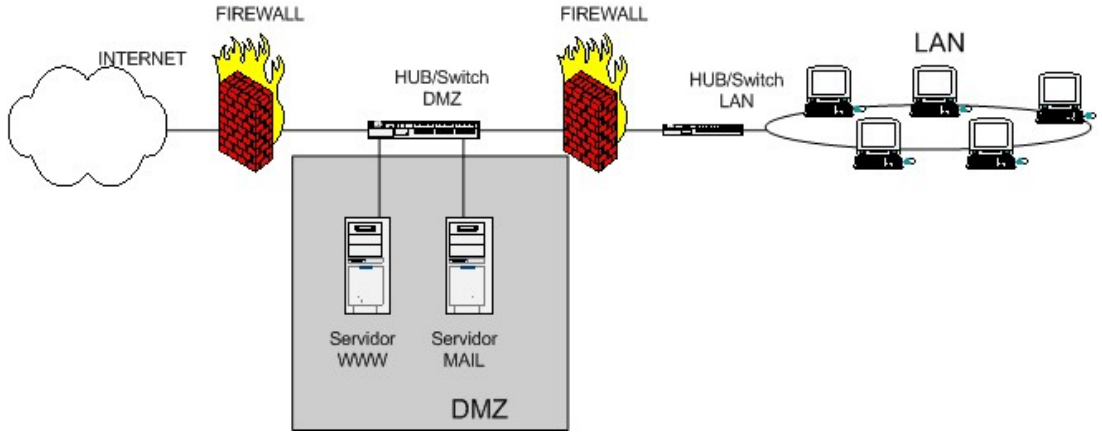
Cortafuegos - Filtrado de paquetes

- Red interna conectada a Internet a través de un *router* cortafuegos
- El *router* opera en modo filtrado paquete-por-paquete
- La decisión de enviar/desechar paquetes se basa en:
  - Direcciones IP origen y destino
  - Puerto TCP o UDP origen y destino
  - Tipo de mensaje ICMP
  - Bits TCP SYN o ACK
- Ejemplos
  - Si queremos bloquear todos los flujos UDP de entrada/salida y las conexiones telnet $\Rightarrow$ bloquear datagramas de entrada y de salida con campo de protocolo IP = 17 y con puerto de origen o de destino = 23.
  - Si queremos impedir que clientes externos realicen conexiones TCP con clientes internos, pero permitir a los clientes internos conectarse con el exterior $\Rightarrow$ bloquear segmentos TCP de llegada con SYN=1 y ACK=0.

Cortafuegos - Configuraciones típicas de seguridad

- La topología básica de red al introducir un cortafuegos se puede mejorar de la siguiente manera:
  - Los equipos de la empresa que **no dan servicios al exterior** estarán en una red
  - Los equipos de la empresa que dan algún tipo de servicio al exterior están en **otra red independiente y aislada**, tradicionalmente denominada zona desmilitarizada o DMZ
  - De esta manera si algún *hacker* entra a esos equipos que dan servicios externos quedaría aislado





Cortafuegos - Pasarelas de aplicación

Problema

Los cortafuegos de filtrado de paquetes no permiten, por ejemplo, autenticar al nivel de usuario para permitir el uso de ciertos servicios, sino que los cierran o permiten para todos

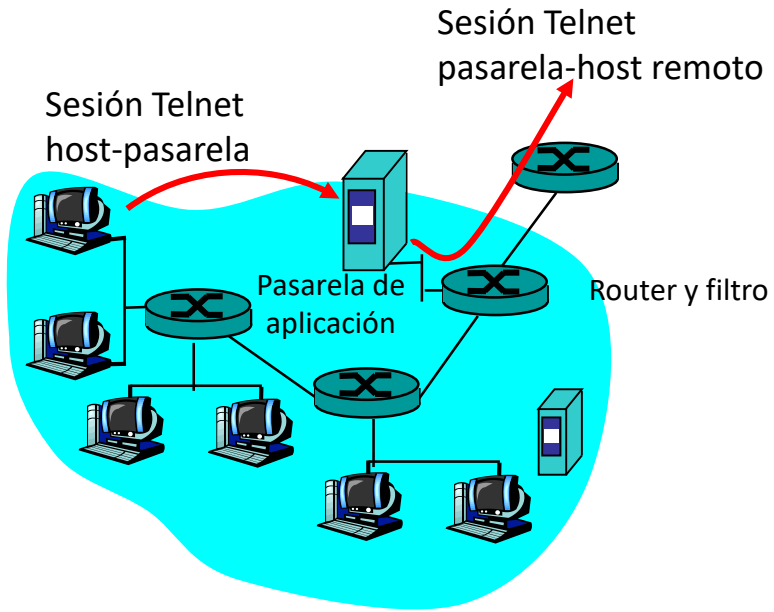
Solución

Contar con una pasarela de aplicación, un cortafuegos que realiza el filtrado al nivel de la capa de aplicación en lugar de basándose únicamente en las características de los paquetes

- **Configuración**: el servicio ofrecido por una pasarela de aplicación también se suele conjugar con un cortafuegos de filtrado de paquetes a nivel de las capas IP, TCP o UDP (ya explicado)
- **Aplicaciones**: hay ejemplos de pasarelas para Telnet, HTTP (servidor proxy), FTP, correo electrónico, etc.

Pasarelas de aplicación - Ejemplo

- **Ejemplo**: permitir la selección de un conjunto de usuarios internos para realizar un `telnet` hacia el mundo exterior:
  - Requiere que todos los usuarios de `telnet` accedan por la pasarela
  - Para los **usuarios autorizados**, la pasarela establece la conexión `telnet` con el host de destino. La pasarela transmite datos entre las dos conexiones
  - El **filtro del router bloquea** todas las conexiones `telnet` que no se han originado en la pasarela



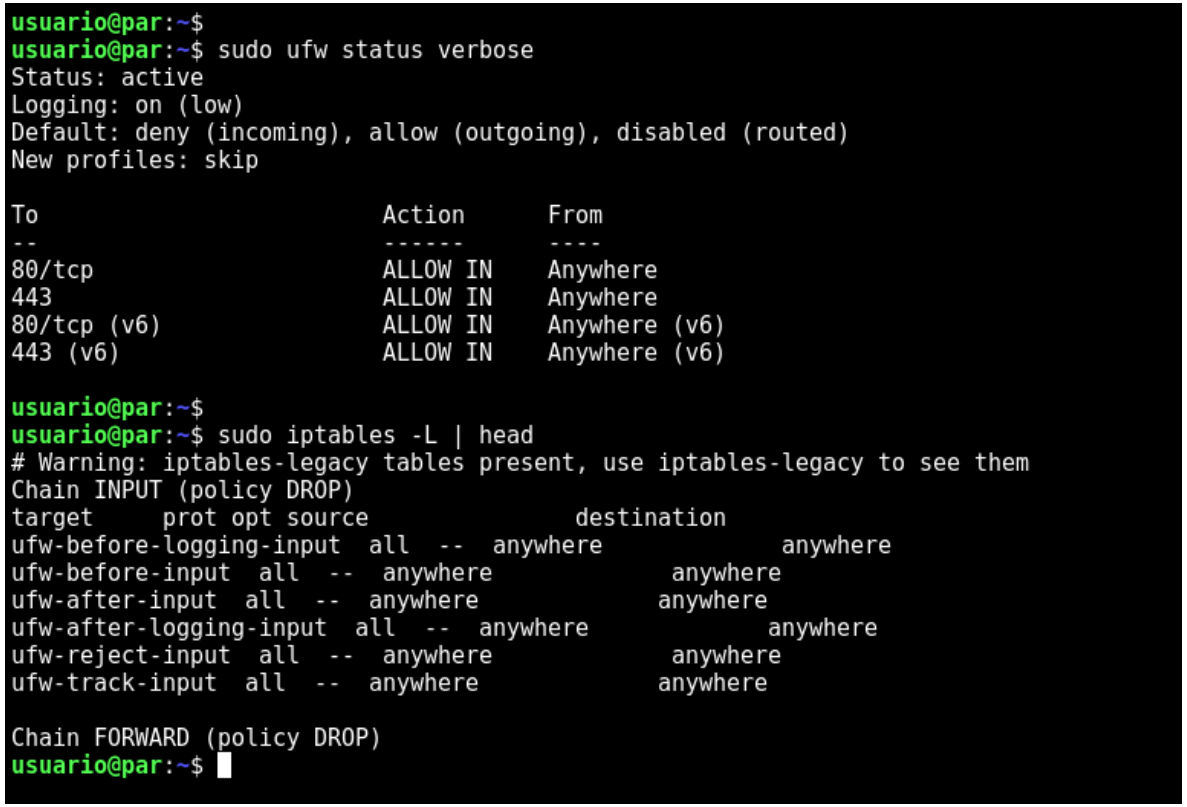
Actividad - Cortafuegos

Objetivo

Aprender a gestionar el cortafuegos integrado en Ubuntu

Descripción

- Usa la orden `sudo ufw status verbose` para verificar que el cortafuegos está instalado y en funcionamiento
- Consulta la ayuda de la orden `ufw` para saber cómo consultar y modificar la configuración del cortafuegos
- Introduce la orden sudo `iptables -L` para obtener una lista de las reglas que rigen el control de puertos
- Consulta la ayuda de la orden `iptables` para saber cómo obtener información sobre las reglas existentes y cómo definir nuevas reglas
- Asumiendo que tuvieses un servidor web en tu red, define una regla que permita acceder al mismo desde el exterior



Ataques y contramedidas

Ataques y contramedidas

Resumen

Ya conocemos los componentes básicos para garantizar la comunicación segura en redes: cifrado, autenticación, integridad y control de acceso. Ahora nos interesa conocer qué tipos de ataques podemos sufrir y qué contramedidas podemos planificar contra ellos

Ataques posibles y medidas de prevención

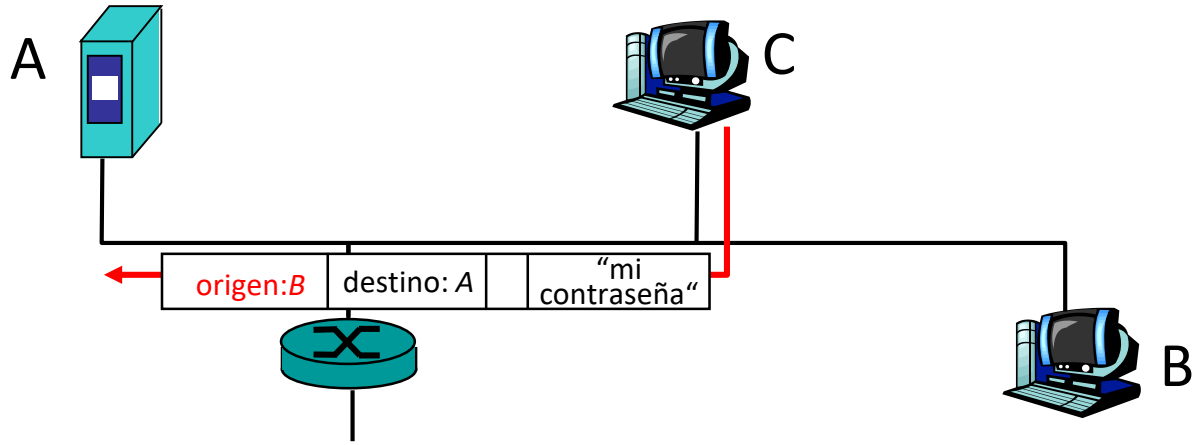
- Antes de atacar **planean el ataque** adquiriendo información de direcciones IP activas, puertos, servicios de la red, etc. Por ejemplo:
  - **Uso de `ping`:** para determinar qué *host* tienen direcciones en la red
  - **Escaneado de puertos:** intentar establecer conexión TCP para cada puerto secuencialmente (y ver qué respuesta se obtiene)
- **Contramedidas**: por ejemplo usando la aplicación **nmap** (`http://www.insecure.org/nmap/`): "exploración de la red e información de actividades maliciosas"
  - Registrar el tráfico entrante en la red
  - Observar actividades sospechosas (direcciones IP , puertos escaneados secuencialmente)

Ataques y contramedidas - Husmeo de paquetes

- Un programa husmeador de paquetes (*sniffer*) **lee el contenido de todos los paquetes** que pasan a través de su controladora de red (tarjeta en modo promiscuo)
  - El peligro aumenta en redes de canal de difusión (Ethernets con *hubs*, inalámbricas, etc.) o si consiguen introducir el software en un servidor
  - Puede leer todos los datos sin encriptar (por ejemplo, las contraseñas)
- **Contramedidas**:
  - Detección mediante software en los ordenadores de interfaces que funcionan en este modo promíscuo
  - Asegurar el hardware de redes
    - Cambiando en redes Ethernet los *hub* por *switches*
    - Utilizando **encriptación fuerte** en redes inalámbricas
      - Evitar protocolos con vulnerabilidades conocidas como WEP

Ataques y contramedidas - Falsificación

- La más usual es la **falsificación de tu dirección IP** dando otra dirección que te interese
- **Contramedidas**:
  - **Filtrado de entrada**: los *routers*, y más concretamente el de salida a Internet, deben filtrar y eliminar paquetes que vengan de su LAN y que no tengan como dirección IP origen una perteneciente a esa LAN
    - No es una técnica muy difundida
    - No sería válida si el ataque procede de la propia LAN



Ataques y contramedidas - Denegación de servicio

- **Funcionamiento**: consiste en sobrecargar de alguna manera la red o el propio servidor para que no pueda ofrecer un servicio.
- **Ejemplos**:
  - **Inundación** de paquetes SYN (de establecimiento de conexión)
    - Estos paquetes tienen direcciones IP origen diferentes.
    - El servidor no puede distinguir a verdaderos clientes del ataque
    - El servidor agotará sus recursos con conexiones inútiles.
  - Envío de **fragmentos IP** que nunca completan un datagrama
    - El servidor agota sus *buffers* acumulando fragmentos
  - **Ataque *smurf***: envío de solicitud de eco (*ping*) a múltiples máquinas con la dirección del servidor. Esto propicia un gran tráfico hacia este
- **Procedencia**: puede venir simultáneamente de diferentes fuentes (DDoS Denegación de servicio distribuida)
  - Puede que estas fuentes sean "inocentes" p. ej: si se han robado contraseñas, si responden a un *ping*, etc.
- **Contramedidas**: complicadas y algunas poco eficaces:
  - Filtrado de paquetes: suele ser inútil porque no se sabe la naturaleza del origen
  - Rastreo de paquetes para detectar el origen, el problema es que el origen puede no saber nada

Ataques y contramedidas - Secuestro

- **Funcionamiento**: este ataque consiste en que un interlocutor **C** toma el control y suplanta a uno de los interlocutores (**A** o **B**) de una conexión
- **Preparación**: se consigue tras el *husmeo* de la conexión con el que se consigue toda información sobre esta (direcciones IP y puertos, números de secuencia, números de reconocimiento, etc.)
- **Contramedidas**: dependen del ámbito (internet/intranet) pueden funcionar otras contramedidas ya citadas en los apartados previos:
  - Cortafuegos, software antí-husmeo, etc.
  - También se puede firmar digitalmente los datos

Actividad - Ataque de denegación de servicio

Objetivo

Conocer una técnica básica de ataque de denegación de servicio

Descripción

- Ciertos tipos de ataques pueden llevarse a cabo con herramientas básicas disponibles públicamente, como es el caso del comando `hping3` que ya conocemos
- Con la opción `--flood` de dicha herramienta es posible inundar de paquetes al objetivo a atacar
- Esos paquetes pueden llevar activado el bit SYN, con la opción `-S`, e ir dirigidos a un puerto concreto, por ejemplo `-p 80` para atacar un servidor web
- Incluso podemos ocultar la dirección IP origen del ataque, usando para ello la opción `--rand-source` que produce direcciones aleatorias de origen
- Realiza pruebas pero usando **siempre un servidor local**
- Ten en cuenta que en muchos países la realización de un ataque de este tipo se **considera un delito**

```
[usuario@fcharte ~]$ sudo hping3 --rand-source -p 80 -S --flood 192.168.2.132
using enp0s3, addr: 192.168.2.176, MTU: 1500
HPING 192.168.2.132 (enp0s3 192.168.2.132): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.2.132 hping statistic ---
411301 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[usuario@fcharte ~]$ s
```

Cuestiones clave de este tema

Cuestiones clave

Qué deberías saber

Al inicio de este tema se planteaban unos objetivos específicos que deberían permitirte ***responder a las siguientes cuestiones*** *clave*

Cuestiones

- ¿Cuáles son las características esenciales que se demandan de una comunicación para que esta sea considerada segura?
- ¿Qué papel juegan los algoritmos de cifrado en la comunicación entre un cliente y un servidor?
- ¿En qué consiste un protocolo de autenticación y cómo los diferentes tipos de ataque vulneran los protocolos de autenticación básicos?
- ¿Cuál es la función de los certificados digitales actualmente en Internet?
- ¿De qué manera se puede garantizar la integridad del contenido de los mensajes?
- ¿Cómo TLS aporta mecanismos de cifrado, autenticación e integridad?
- ¿Cómo controlar el acceso a servicios mediante cortafuegos?

Material adicional

Descripción

Para ampliar tus conocimientos sobre los contenidos de esta semana te recomendamos que consultes los recursos indicados a continuación.

Recursos

- **Autenticación en Redes informáticas**, vídeo disponible en https://youtu.be/cBM8ShsvnnU y en PLATEA sobre los algoritmos de autenticación
- **Capítulo 8** Seguridad en las redes de computadores, del libro Redes de computadoras 7ED disponible en formato digital en la BUJA (recuerda identificarte para poder acceder a leerlo desde tu navegador), concretamente las **secciones 8.1 a 8.4**
- **RFC 1704**, de octubre de 1994, el documento On Internet Authentication describe diferentes tipos de autenticación y tipos de ataques a los que deben enfrentarse
- **Cryptographic hash function** en la Wikipedia enumera múltiples funciones de resumen con sus características básicas
- **Digital Attack Map**, es un sitio web en el que podemos encontrar información sobre los ataques de tipo DDoS a nivel mundial, con representaciones sobre mapas del origen y destino de los ataques