



## Fundamentos de Redes

# Tema 4

## Seguridad en Redes

Antonio M. Mora García



# Bibliografía

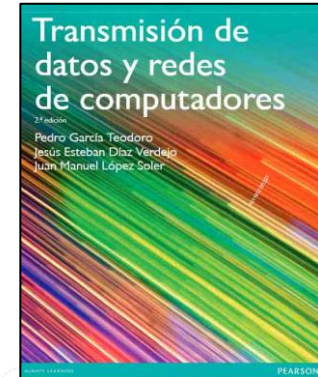
## Básica

- © James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7º Edición. Editorial Pearson S.A., 2017.

### **CAPÍTULO 8**



- © P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler. Transmisión de datos y redes de computadores, 2ª Edición. Editorial Pearson, 2014. **CAPÍTULO 12**



# Índice

◎ **4.1.** Introducción a la seguridad en redes

◎ **4.2.** Mecanismos de seguridad:

- Cifrado
- Autenticación
- Clave secreta
- Funciones Hash
- Firma digital
- Certificados digitales

◎ **4.3.** Implementación de mecanismos de seguridad

# TEMA 4. Seguridad en Redes

## © 4.1. Introducción a la seguridad en redes

## © 4.2. Mecanismos de seguridad:

- Cifrado
- Autenticación
- Clave secreta
- Funciones Hash
- Firma digital
- Certificados digitales

## © 4.3. Implementación de mecanismos de seguridad

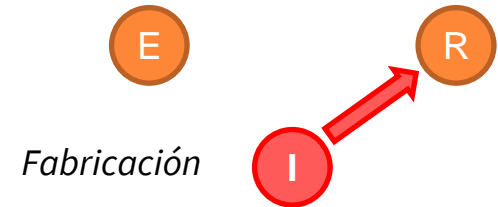
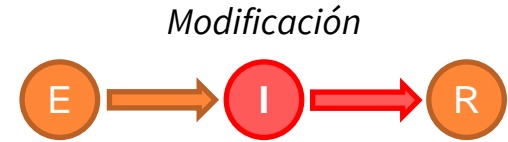
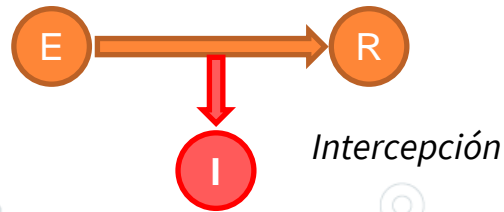
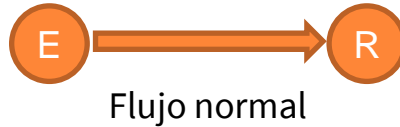
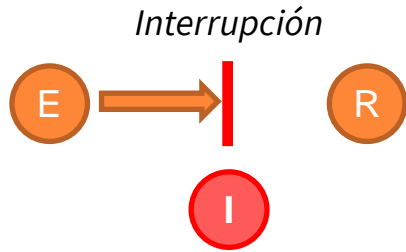
# Introducción

- Una **red de comunicaciones** es **segura** cuando se **garantizan todos los aspectos** de la misma ⇔ no hay protocolos ni redes 100% seguros.
- ¿Qué es la seguridad? → múltiples aspectos:
  - **Confidencialidad/privacidad**: el contenido de la información es comprensible sólo para entidades autorizadas.
  - **Autenticación**: las entidades son quienes dicen ser.
  - **Control de accesos**: los servicios son accesibles sólo para entidades autorizadas.
  - **No repudio o irrenunciabilidad**: el sistema impide la renuncia de la autoría de una determinada acción.
  - **Integridad**: el sistema detecta todas las alteraciones (intencionadas o no) de la información.
  - **Disponibilidad**: el sistema mantiene las prestaciones de los servicios con independencia de la demanda.

# Introducción

- ¿En qué **nivel/capa** se debe situar la **seguridad**? en **TODOS**... el grado de *seguridad lo determina el punto más débil*.
- **Ataque de seguridad**: cualquier **acción** intencionada o no **que menoscaba** cualquiera de los **aspectos de la seguridad**.

## TIPOS DE ATAQUES



# Introducción

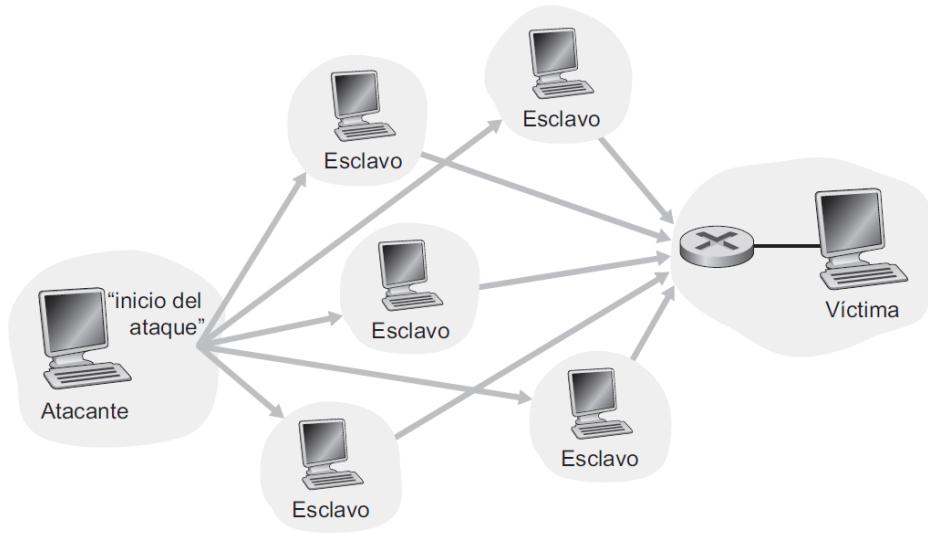
- **Ejemplos de Tipos de ataques:**

- **Sniffing** → vulneración a la confidencialidad, escuchar (husmear). **[Intercepción]**
- **Poofing (phishing)** → suplantación de la identidad de entidades. **[Fabricación]**
- **Man in the Middle (MitM)** → hombre/máquina en medio. **[Intercepción/Modificación]**
- **Distributed Denial of Service (DDoS)** → denegación de servicio distribuido,  
Ej: *Flooding* (inundación) **[Interrupción]**
- **Malware** → **troyano** (software oculto con la apariencia de otro programa), **gusano** (virus que se replica), **spyware** (programa que captura datos privados), **backdoor** (punto desconocido de acceso a nuestra máquina), **rootkit** (software que proporciona acceso remoto), **ransomware** (captura o modificación de datos), **keylogger** (captura las pulsaciones de teclas que hacemos y las envían).

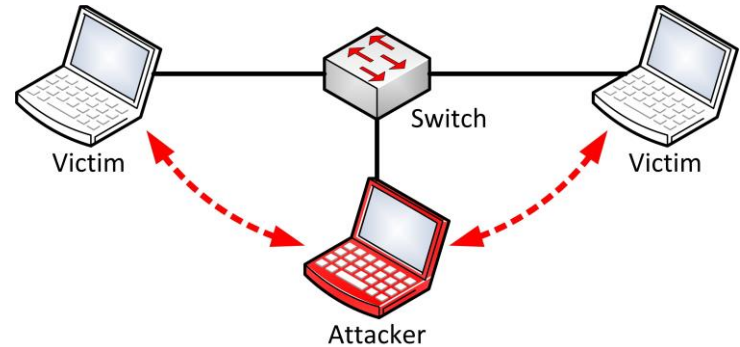
# Introducción

## EJEMPLOS DE ATAQUES

### Ataque DDoS



### Ataque MitM





# Introducción

## MECANISMOS DE SEGURIDAD

- De **prevención**:
  - mecanismos de autenticación e identificación.
  - mecanismos de control de acceso.
  - mecanismos de separación (física, temporal, lógica, criptográfica y fragmentación).
  - mecanismos de seguridad en las comunicaciones (cifrado de la información).
- De **detección**:
  - IDS (*Intrusion Detection System*)
- De **recuperación**:
  - copias de seguridad (*backup*).
  - mecanismos de análisis forense: averiguar alcance, las actividades del intruso en el sistema y cómo entró.

# TEMA 4. Seguridad en Redes

© **4.1.** Introducción a la seguridad en redes

© **4.2. Mecanismos de seguridad:**

- Cifrado
- Autenticación
- Clave secreta
- Funciones Hash
- Firma digital
- Certificados digitales

© **4.3.** Implementación de mecanismos de seguridad

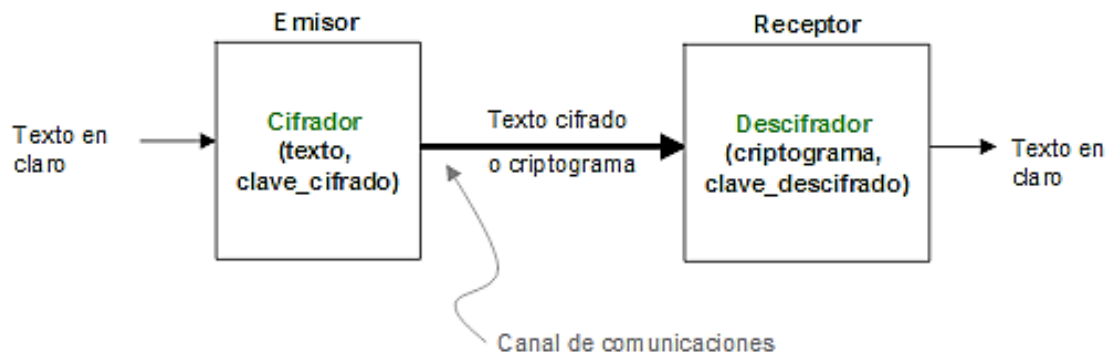
# Mecanismos de seguridad

- **Mecanismos más utilizados:**
  - Cifrado (simétrico y asimétrico)
  - Autenticación con clave secreta (reto-respuesta)
  - Intercambio de Diffie-Hellman (establecimiento de clave secreta)
  - Funciones Compendio o Hash. Hash Message Authentication Code (HMAC).
  - Firma Digital.
  - Certificados digitales.

# Cifrado

- Se basa en la **criptografía** y en la definición de un **criptosistema**:

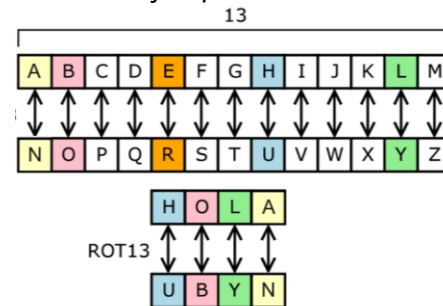
- Alfabeto de partida
- Espacio de claves
- Conjunto de transformaciones de cifrado
- Conjunto de transformaciones de descifrado



- Tipos de criptosistema:

- **Simétricos** o de clave privada (DES, *Data Encryption Standard*)
- **Asimétricos** o de clave pública (RSA, *Rivest-Shamir-Adleman*)

Ejemplo ROT13



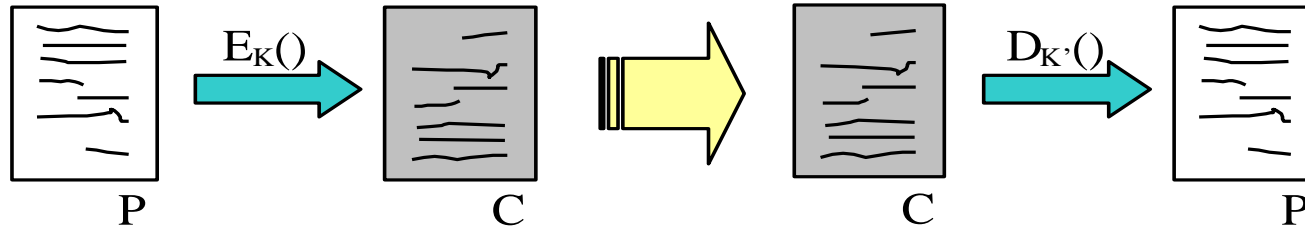
Ejemplo ASCII

A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
J	01001010

# Cifrado

- El **cifrado** es un procedimiento para garantizar la **confidencialidad**:
  - Se parte de un **Texto llano/claro** (*plain text*)
  - Se aplica un **algoritmo de cifrado** conocido como  $E_K()$
  - Y un **algoritmo de descifrado** llamado  $D_{K'}()$
  - Ambos **dependen** respectivamente de una **clave de cifrado  $K$**  y de una **clave de descifrado  $K'$** .

Cifrado  $\Leftrightarrow$  Encriptación  
 Descifrado  $\Leftrightarrow$  Desencriptación



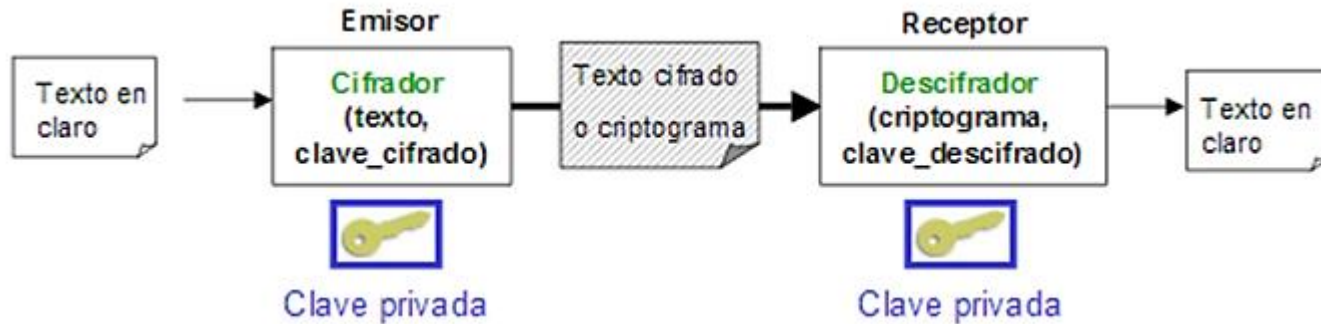
- El texto plano  $P$  se cifra y se convierte en  $C$ , se transmite y posteriormente se descifra  $C$  para obtener  $P$  de nuevo.

# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

- **Emisor y receptor** comparten la **misma clave**.
- La **clave** sólo es conocida por ellos (**privada/secret**a).
- **Emisor encripta** con ella y **receptor descripta con ella**.
- La clave deben compartirla por un canal seguro.

$$K = K'$$

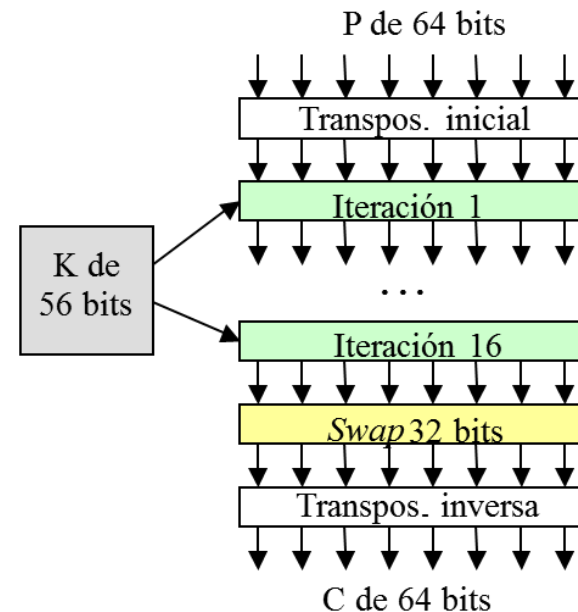


# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

- Algoritmo **DES** (*Data Encryption Standard, IBM 1975*):
  - 1) Se hace una transposición al bloque inicial de bits P
  - 2) 16 iteraciones aplicando la clave K de 56 bits  
[ver transparencia siguiente]
  - 3) Intercambio de 32 bits de orden más alto por los más bajos
  - 4) Transposición inversa de 1)

[http://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Data_Encryption_Standard)



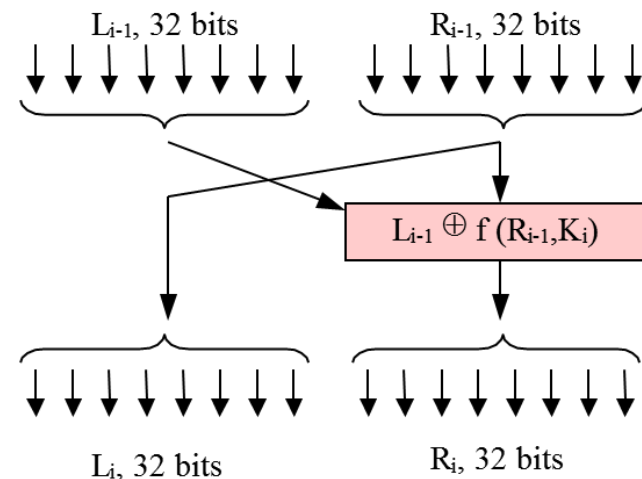
(a)

# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

- Algoritmo **DES** (*Data Encryption Standard, IBM 1975*):
  - 2) 16 iteraciones aplicando la clave K de 56 bits (cada iteración (b))
    - a) 32 bits de la derecha pasan a ser los de la izquierda para la iteración siguiente
    - b) 32 bits de la derecha se obtienen haciendo XOR con los de la izquierda, junto con la aplicación de una función de transposición y duplicación de bits sobre R y K de la iteración actual,  $i$ .  
En dicha función también se utilizan módulos de sustitución para cada grupo de 6 bits (8 grupos) y se obtienen 4 bits por cada bloque. Por último se hace una nueva transposición del resultado.

[http://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Data_Encryption_Standard)



(b)



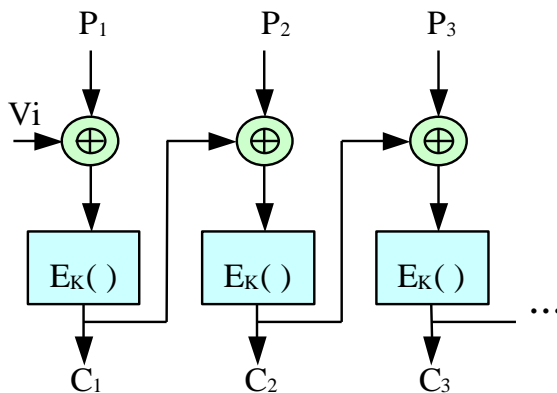
# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

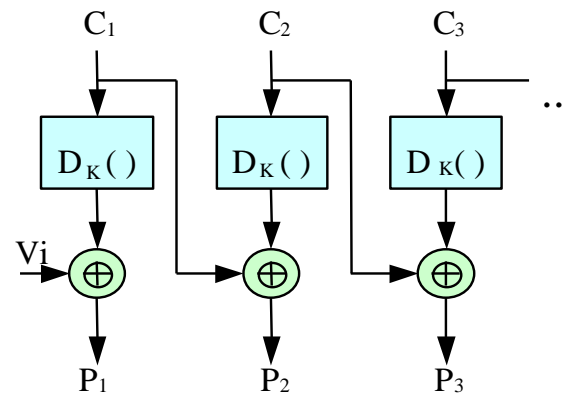
- Encadenamiento **DES**:

- Se realizan varios encriptamientos consecutivos y se combinan los resultados.
- Con cada encriptamiento se aumenta en  $2^{56}$  la dificultad para descubrir la clave.

DES: Complejidad  
para descubrir la  
clave "sólo"  $2^{56}$



(a)



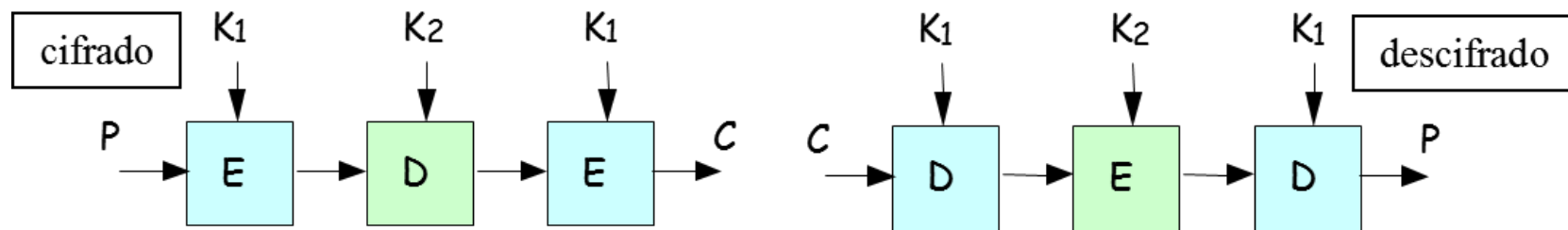
(b)

# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

- **3DES:**

- Se hacen dos fases de encriptado y una de descryptado entre ellas, usando cada vez una clave diferente.



# Cifrado Simétrico

## ALGORITMOS DE CLAVE SECRETA

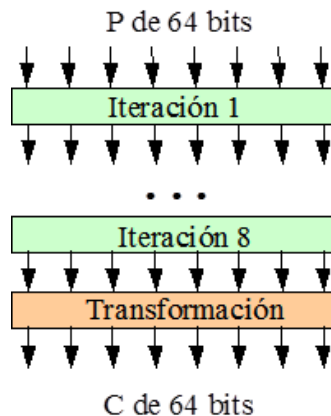
- **IDEA** (*International Data Encryption Algorithm*):

- Utiliza claves de 128 bits.
- Puede operar en tiempo real.
- Fácil de implementar en hardware.
- 8 iteraciones
- Aplica operaciones:

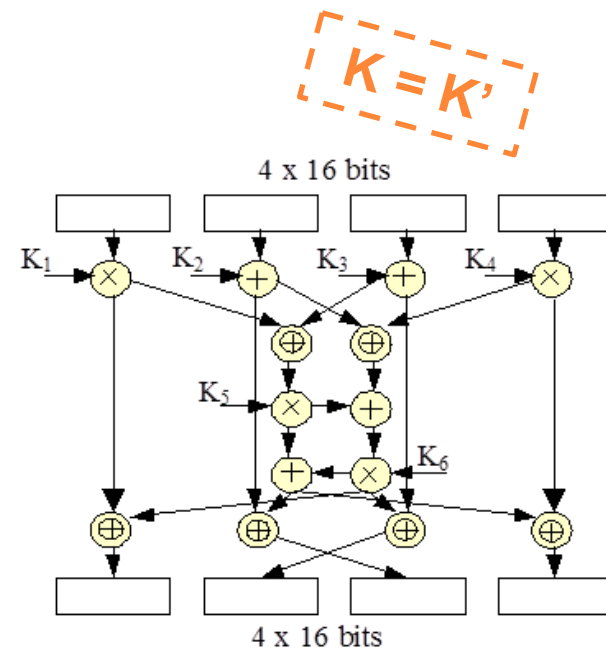
XOR

Suma módulo  $2^{16}$

Multiplicaciones módulo  $2^{16}+1$



(a)



(b)

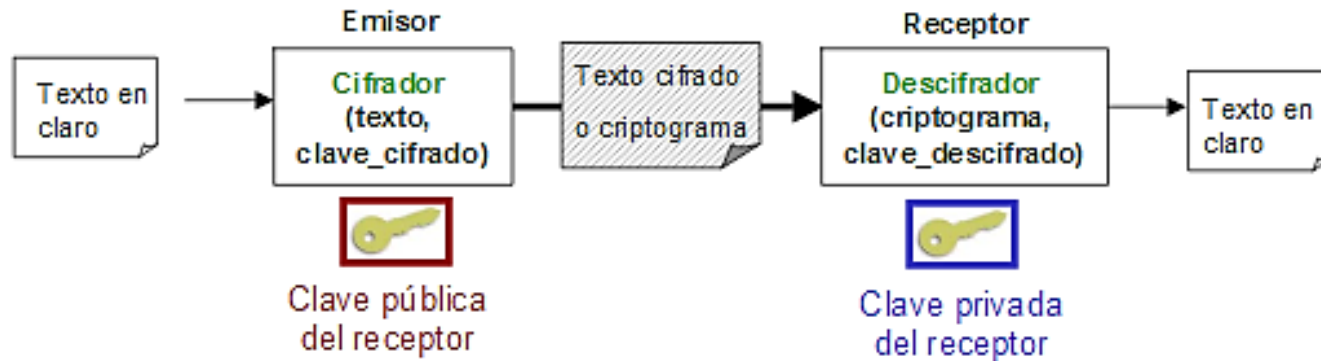
[https://es.wikipedia.org/wiki/International\\_Data\\_Encryption\\_Algorithm](https://es.wikipedia.org/wiki/International_Data_Encryption_Algorithm)

# Cifrado Asimétrico

## ALGORITMOS DE CLAVE PÚBLICA

- El **receptor** tiene una **clave pública** y una **clave privada** (de la que deriva la pública).
- Envía la **clave pública** a los **emisores** potenciales (por cualquier medio).
- **Emisor encripta** con la clave pública del receptor.
- **Receptor descrypta** con su **clave privada**.

$K \neq K'$



# Cifrado Asimétrico

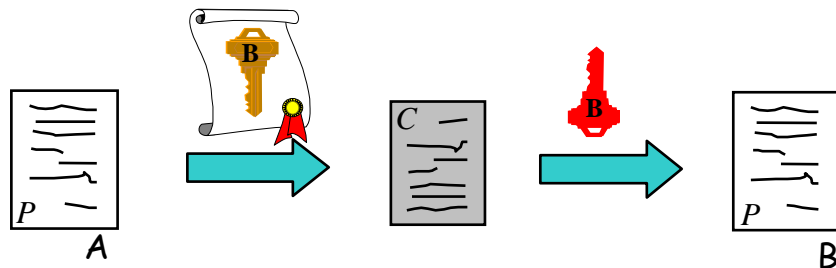
## ALGORITMOS DE CLAVE PÚBLICA

- **Dos claves por usuario (B):** una **pública**  $K_{PUB_B}$  y otra **privada**  $K_{PRI_B}$  **distintas**
- Conocida  $K_{PUB_B}$  es **imposible conocer**  $K_{PRI_B}$
- **Claves diferentes** para **cifrar y descifrar**:

$K \neq K'$

Cifrar  $\rightarrow C = E_{K_{pubB}}(P)$

Descifrar:  $P = D_{K_{priB}}(C)$



- ¿Y si enviamos  $C = E_{K_{privA}}(P)$ ?  $\rightarrow$  **autenticación**

# Cifrado Asimétrico

## ALGORITMOS DE CLAVE PÚBLICA

- **RSA** (*Rivest, Shamir, Adleman*)

1) Elegimos  $p$  y  $q$  primos grandes ( $>10^{100}$ )

2)  $n = (p \cdot q)$  y  $z = (p-1) \cdot (q-1)$  (*función de Euler*)

3) Elegimos  $d$  coprimo con  $z$  (no tienen factores primos en común)

4) Calculamos  $e$  tal que  $e \cdot d \bmod z = 1$  (*algoritmo de Euclides*)

5)  $K_{pub} = (e, n)$  y  $K_{pri} = (d, n)$ , de modo que:

$$* C = P^e \bmod n$$

$$* P = C^d \bmod n$$

$K \neq K'$

<https://es.wikipedia.org/wiki/RSA>

# Cifrado Asimétrico

## ALGORITMOS DE CLAVE PÚBLICA

### • EJEMPLO RSA

$$p = 3, q = 11$$

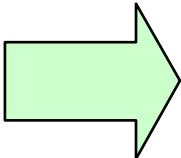



$$n = p \cdot q = 33, \quad z = (p-1) \cdot (q-1) = 20 \quad (= 5 \cdot 2 \cdot 2 \text{ en factores primos})$$

$$d = 7, \text{ coprimo respecto a } z$$

$$e = 3, e \cdot d \bmod z = 1$$

$$K_{pub} = (3, 33) \text{ y } K_{pri} = (7, 33)$$

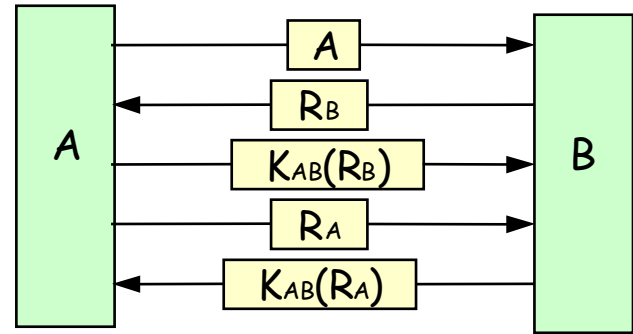
$$\begin{aligned} C &= P^e \bmod n \\ P &= C^d \bmod n \end{aligned}$$

<u>Simbólico</u>	<u>Numérico</u>	<u><math>P^3</math></u>	<u><math>P^3 \bmod 33</math></u>		<u><math>C^7</math></u>	<u><math>C^7 \bmod 33</math></u>	<u>Simbólico</u>
S	19	6859	28		13492928512	19	S
U	21	9261	21		1801088541	21	U
Z	26	17576	20		1280000000	26	Z
A	01	1	1		1	01	A
N	14	2744	5		78125	14	N
N	14	2744	5		78125	14	N
E	05	125	26		8031810176	05	E
 P		 C			 P		

# Autenticación

## AUTENTICACIÓN Y CIFRADO DE CLAVE SECRETA

- **Esquema reto-respuesta (criptográfica):**
  - A desea autenticarse en B
  - B le plantea un “reto” a A
  - A responde al reto encriptándolo con la clave privada/secreta compartida entre A y B
  - B comprueba si la respuesta es correcta y si lo es A se autentica
  - El proceso se puede repetir para autenticar a B.
- **Variante no criptográfica:**
  - La respuesta es la contraseña → ataque replay
  - Contraseña con identificador → ataque replay con id
  - Contraseña de un solo uso



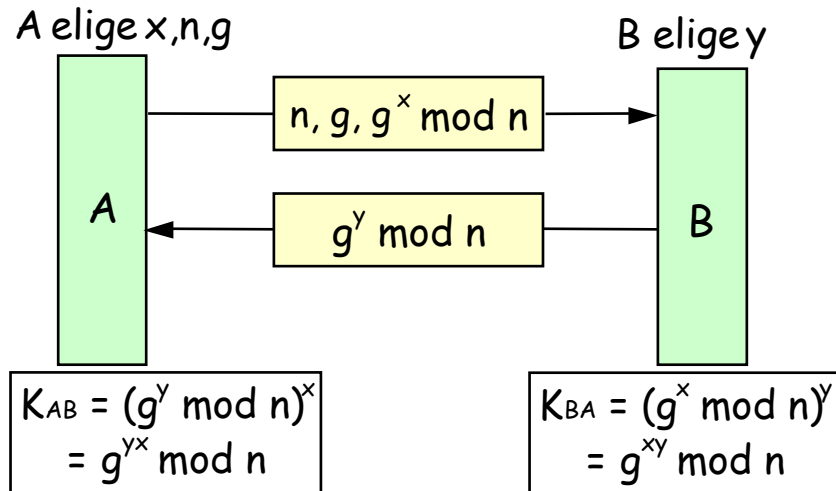
**Mensaje nonce**  
(sólo se genera una vez)



# Clave secreta

## ESTABLECIMIENTO DE CLAVE SECRETA

- **Intercambio de Diffie-Hellman:** permite establecer una clave secreta entre dos entidades a través de un canal no seguro.



### EJEMPLO:

$g=7, n=23$

1. A elige  $x = 3$  y calcula  $R1 = 7^3 \bmod 23 = 21$ .
2. A envía el número 21 a B.
3. B elige  $y = 6$  y calcula  $R2 = 7^6 \bmod 23 = 4$ .
4. B envía el número 4 a A.
5. A calcula la clave privada/simétrica  $K = 4^3 \bmod 23 = 18$ .
6. B calcula la clave privada/simétrica  $K = 21^6 \bmod 23 = 18$ .

**El valor de K es el mismo para A y B:**

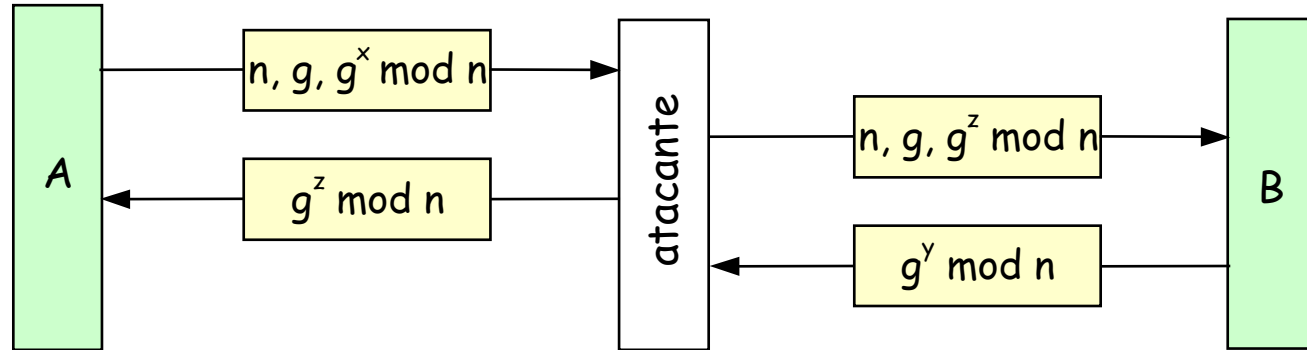
$$g^{xy} \bmod n = 7^{18} \bmod 23 = 18.$$

Usando números grandes no es vulnerable a escucha del canal

# Clave secreta

## ESTABLECIMIENTO DE CLAVE SECRETA

- **Intercambio de Diffie-Hellman:** permite establecer una clave secreta entre dos entidades a través de un canal no seguro.



Vulnerable a ataque MitM

# Funciones Hash

## FUNCIONES COMPENDIO (RESUMEN O DIGEST)

- **Funciones unidireccionales** (irreversibles) de cálculo sencillo.
- Texto de **entrada (M)** de **longitud variable**.
- $M \rightarrow H(M)$  siendo  **$H(M)$  de longitud fija** (256 ó 512 bits)
- Imposible obtener M a partir de su resumen  $H(M)$ .
- **Invulnerables a ataques de colisión**, dado M es imposible encontrar  $M'$  tal que

$$M \neq M' \text{ y } H(M) = H(M')$$

- Ejemplos de funciones Compendio/Digest/Hash: MD5, SHA-1, SHA-512
- Las funciones **Hash** se pueden usar para **garantizar integridad + autenticación** (clave K):

**Hash Message Authentication Code (HMAC):**  $M + H(K|M)$  pero para evitar ataques de extensión se usa  $M + H(K | H(K | M))$

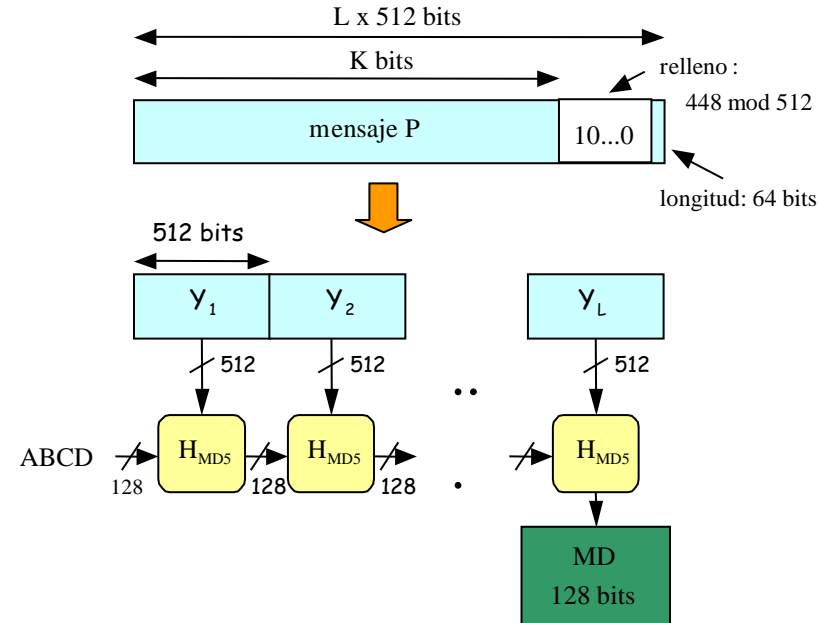
Integridad →  
datos sin alterar  
Autenticación →  
generar hash  
correcto

# Funciones Hash

## MD5 (Message Digest 5, RFC 1321)

- Relleno bits “100..0” por la derecha, de longitud máxima 448 bits
- Adición de campo de longitud de 64 bits
- División del mensaje en bloques de 512 bits
- Procesamiento secuencial por bloques.
- De cada bloque se obtiene un digest de 128 bits.

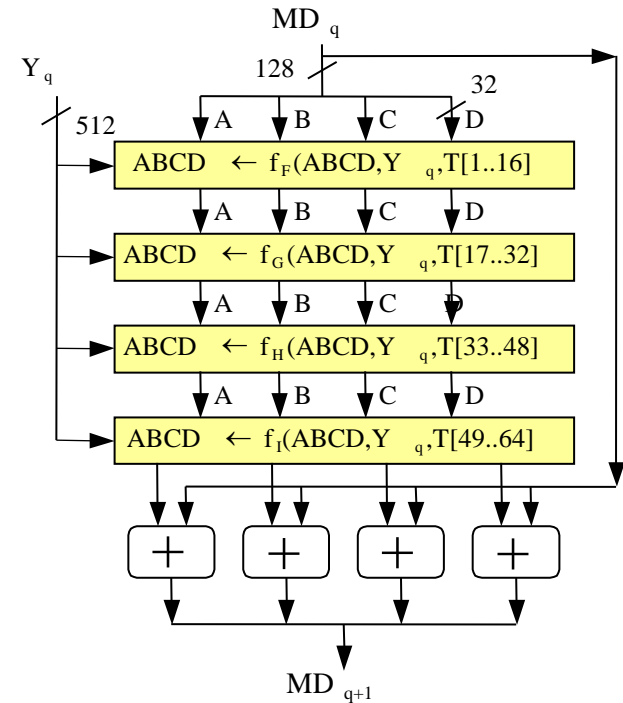
ABCD son 4 registros de 32 bits con valores constantes hexadecimales



# Funciones Hash

## MD5 (Message Digest 5, RFC 1321)

- Cada bloque se procesa:
  - Se usan varias funciones (F, G, H, I) de operadores binarios (XOR, AND, OR, NOT) combinadas.
  - Se aplican los valores de los registros A, B, C, D.
  - Se hacen desplazamientos de bits.
  - Se hacen varias pasadas.
  - Se hace una suma final módulo  $2^{32}$ .
  - La salida de un bloque será la entrada del siguiente.

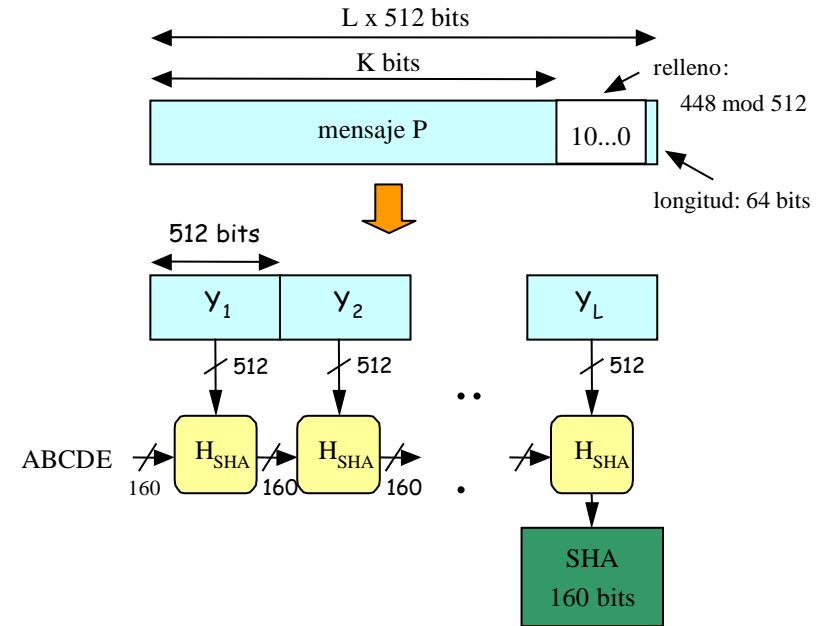


# Funciones Hash

## SHA-1 (Secure Hash Algorithm 1, RFC 3174)

- Relleno bits “100..0” por la derecha, de longitud máxima 448 bits
- Adición de campo de longitud de 64 bits
- División del mensaje en bloques de 512 bits
- Procesamiento secuencial por bloques.
- De cada bloque se obtiene un digest de 160 bits.

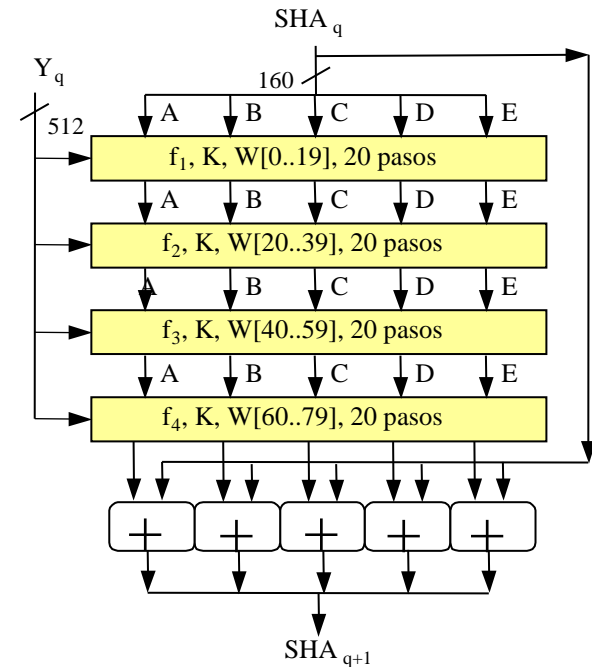
ABCDE son 5 registros de 32 bits con valores constantes hexadecimales (diferentes de los de MD5)



# Funciones Hash

## SHA-1 (*Secure Hash Algorithm 1*, RFC 3174)

- Cada bloque se procesa:
  - Se divide el bloque en palabras de 32 bits.
  - Se extienden las palabras combinándolas hasta tener 80.
  - Se agrupan de 20 en 20 y se combinan usando funciones.
  - Se usan varias funciones de operadores binarios (XOR, AND, OR, NOT) combinadas entre sí.
  - Se aplican los valores de los registros A, B, C, D, E.
  - Se hacen 4 pasadas de este proceso.
  - Se hace una suma final módulo  $2^{32}$ .
  - La salida de un bloque será la entrada del siguiente.

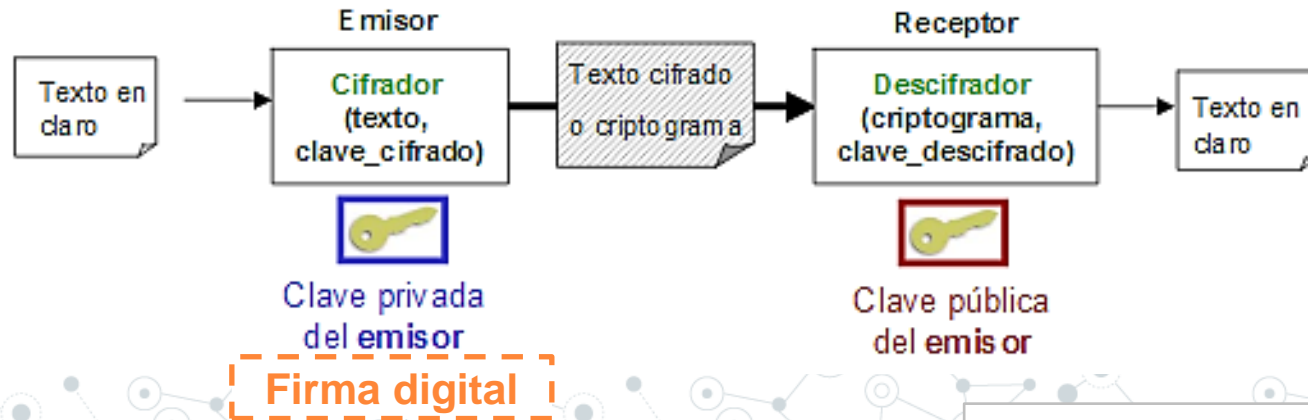


# Firma Digital

- Una **firma digital** es un conjunto de **datos** que, consignados junto a otros o asociados con ellos, pueden **ser utilizados como** medio de **identificación del firmante**.

## OBJETIVOS

- Que el **receptor** pueda **autenticar al emisor**.
- Que **no** haya **repudio** (que el emisor no pueda alegar que él no envió el mensaje).
- Que el **emisor** tenga **garantías de no falsificación** de su mensaje (integridad).

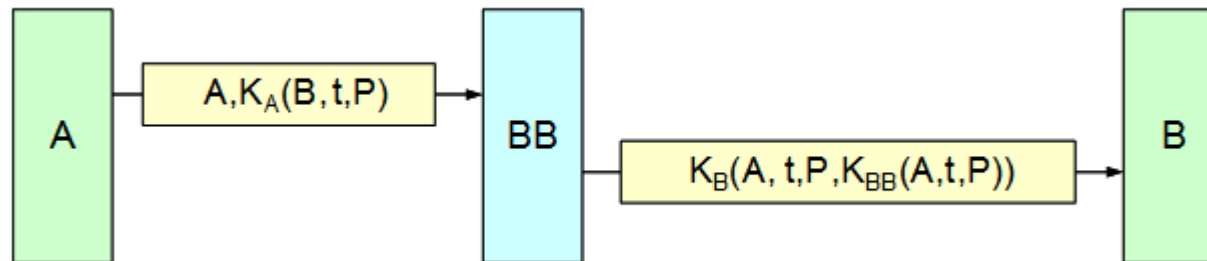




# Firma Digital

## FIRMA DIGITAL. BIG BROTHER

- Entidad central (BB) que interviene en el proceso de firma digital para la transmisión de un mensaje  $P$  entre  $A$  y  $B$ .
- $A$  envía el mensaje cifrado con una clave que comparte con BB,  $K_A$ , incluyendo además el propio destino del mensaje,  $B$ , y una marca de tiempo  $t$ .
- BB envía a  $B$  el mensaje cifrado con la clave que comparte con él,  $K_B$ , la identidad de  $A$ , el mensaje  $P$ , su propia marca de tiempo  $t$  y su firma digital. La firma serán estos mismos valores encriptados con su propia clave  $K_{BB}$ .

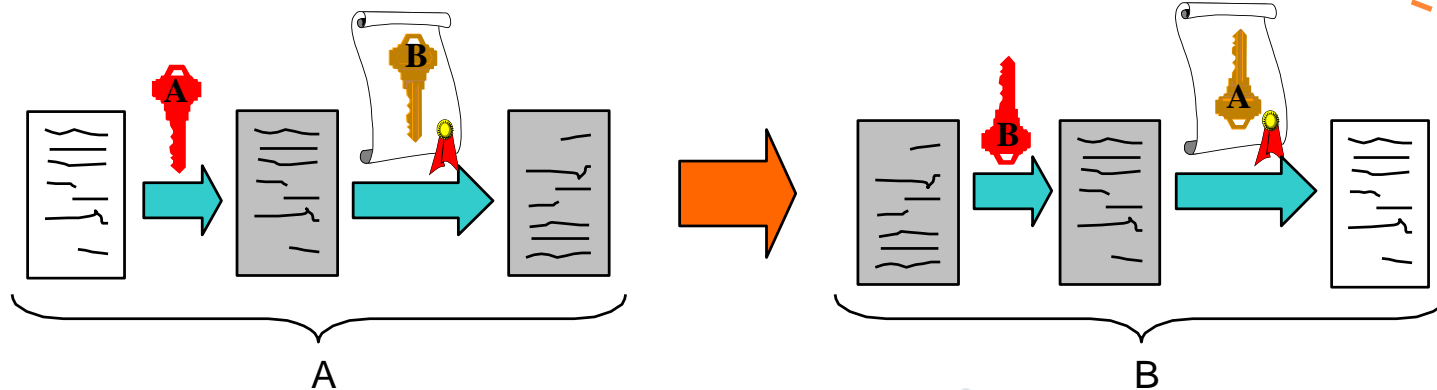


# Firma Digital

## FIRMA DIGITAL CON CLAVE ASIMÉTRICA. DOBLE CIFRADO

- Un cifrado para autenticación, con  $K_{\text{priA}}$
- Otro, para proporcionar privacidad, con  $K_{\text{pubB}}$
- Para firmar, enviar  $K_{\text{pubB}}(K_{\text{priA}}(P))$
- En el receptor se descripta:  $K_{\text{pubA}}(K_{\text{priB}}(K_{\text{pubB}}(K_{\text{priA}}(P))))=P$

**PROBLEMA**  
Garantizar el no repudio:  
Asociación fehaciente e  
indisoluble de A con su clave  
pública  $K_{\text{pubA}}$



# Certificados digitales

- Un **certificado digital** sirve para **garantizar** la asociación **identidad-clave**.
- Para que un usuario no pueda **corromper una clave pública** (de otro) y decir que es suya.

## AUTORIDADES DE CERTIFICACIÓN (AC)

- **Entidad** para **garantizar** la **asociación** entre **identidad** y **claves**.
- **Funcionamiento:**
  - El usuario obtiene sus claves pública y privada
  - Éste envía una solicitud, firmada digitalmente, a la AC indicando su identidad y su clave pública
  - AC comprueba la firma y emite el certificado solicitado:
    - \* Identidad de AC, identidad del usuario, clave pública del usuario y otros datos como, por ejemplo, el período de validez del certificado.
    - \* Todo ello se firma digitalmente con la clave privada de AC con objeto de que el certificado no pueda falsificarse .
- **Formato** de certificados: principalmente **X.509**.

# Certificados digitales

## AUTORIDADES DE CERTIFICACIÓN (AC)

- Las AC son responsables de:
  - emitir los certificados
  - asignarles una fecha de validez
  - revocarlos antes de esta fecha (en casos determinados)
- AC reconocidas:
  - ☐ ACE ([www.ace.es](http://www.ace.es))
  - ☐ VeriSign ([www.verisign.com](http://www.verisign.com))
  - ☐ CAMERFIRMA ([www.camerfirma.es](http://www.camerfirma.es))
  - ☐ CERES ([www.cert.fnmt.es](http://www.cert.fnmt.es))

*No es lo mismo Firma Digital  
(un uso en una transmisión)  
que Certificado Digital (muchos usos,  
acreditar nuestra identidad)*

# Certificados digitales

## TIPOS DE CERTIFICADOS

- **Certificados firmados localmente:**
  - Firmados por un servidor local.
  - De uso interno en una red privada (intranet).
  - Para garantizar los intercambios confidenciales y para autenticar usuarios.
- **Certificados firmados por una autoridad de certificación:**
  - Válidos en todo Internet.
  - Para garantizar los intercambios seguros con usuarios anónimos.
  - Para acreditar la identidad de un usuario.

# Certificados digitales

## CERTIFICADO X.509

<i>Field</i>	<i>Explanation</i>
Version	Version number of X.509
Serial number	The unique identifier used by the CA
Signature	The certificate signature
Issuer	The name of the CA defined by X.509
Validity period	Start and end period that certificate is valid
Subject name	The entity whose public key is being certified
Public key	The subject public key and the algorithms that use it

# Certificados digitales

## CERTIFICADO X.509

Certificate:

Data:

Version: 1 (0x0)  
 Serial Number: 7829 (0x1e95)  
 Signature Algorithm: md5WithRSAEncryption  
 Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,  
 OU=Certification Services Division,  
 CN=Thawte Server CA/Email=server-certs@thawte.com

**Autoridad  
Certificadora**

Validity

Not Before: Jul 9 16:04:02 1998 GMT

Not After : Jul 9 16:04:02 1999 GMT

**Validez**

Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,  
 OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org

**Datos del  
usuario**

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:  
 33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:  
 66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:  
 70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:  
 16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:  
 c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:  
 8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:  
 d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:  
 e8:35:1c:9e:27:52:7e:41:8f

**Algoritmo y  
clave pública  
del usuario**

Exponent: 65537 (0x10001)

Signature Algorithm: md5WithRSAEncryption

93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:  
 92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:  
 ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:  
 d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:  
 0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:  
 5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:  
 8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:  
 68:9f

**Algoritmo y  
clave privada  
de la AC**

# Resumen

- Relación entre los mecanismos de seguridad y los servicios de seguridad:
  - **Confidencialidad:** Se consigue mediante Cifrado (simétrico o asimétrico).
  - **Autenticación:** Se consigue con los mecanismos de autenticación (reto-respuesta), y Firma digital (big brother, doble cifrado: cifrado en el emisor con clave privada y descifrado en receptor con clave pública).
  - **No repudio o irrenunciabilidad:** Se consigue mediante firma digital (big brother, doble cifrado), Certificado digital.
  - **Integridad:** Se consigue añadiendo resúmenes generados con funciones hash/digest.
  - **Disponibilidad:** Los mecanismos no proporcionan disponibilidad por sí mismos.  
Serían necesarios sistemas antiataque, redundancia en las líneas de acceso, en los servidores, etc.



# TEMA 4. Seguridad en Redes

- ◎ **4.1.** Introducción a la seguridad en redes
- ◎ **4.2.** Mecanismos de seguridad:
  - Cifrado
  - Autenticación
  - Clave secreta
  - Funciones Hash
  - Firma digital
  - Certificados digitales
- ◎ **4.3.** Implementación de mecanismos de seguridad

# Implementación de mecanismos de seguridad

- **Seguridad perimetral**
  - Firewalls, IDS (sistemas detección intrusiones), IRS (sistemas respuesta intrusiones)

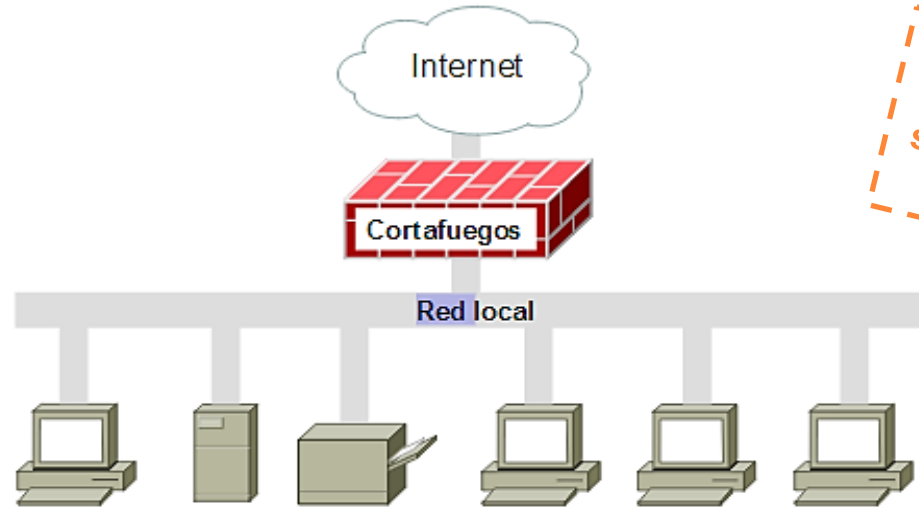
## PROTOCOLOS DE SEGURIDAD

- **Capa de Aplicación**
  - Pretty Good Privacy (**PGP**)
  - Secure Shell (**SSH**)
- **Capa de Transporte**
  - Secure Socket Layer (**SSL**) → HTTPS, IMAPS, SSL-POP
  - Transport Layer Security (**TLS**)
- **Capa de Red** → IPsec (VPN)
- **Capas inferiores** → PAP, CHAP, MS\_CHAP, EAP...

# Implementación de mecanismos de seguridad

## CORTAFUEGOS (*FIREWALL*)

- Es una combinación de técnicas, políticas de seguridad y tecnologías (hardware y software).
- Proporciona seguridad en la red, controlando el tráfico que entra y sale (normalmente entre una red privada e Internet).



*Debe combinarse  
con Protocolos  
seguros (seguridad  
dentro de la red)*

# Implementación de mecanismos de seguridad

## CORTAFUEGOS (*FIREWALL*) – FUNCIONES

- **Controlar** (permitiendo o denegando) los **accesos** desde la **red local** hacia el exterior **y** los **accesos desde el exterior** hacia la red local.
- **Filtrar los paquetes** que circulan, de modo que **sólo los servicios permitidos** puedan pasar.
- **Monitorizar** el **tráfico**, supervisando **destino, origen** y cantidad de **información** recibida y/o enviada.
- **Almacenar** total o parcialmente los **paquetes** que circulan a través de él **para analizarlos** en caso de problemas.
- Establecer un **punto de cifrado** de la información si se pretende comunicar dos redes locales a través de Internet.

# Implementación de mecanismos de seguridad

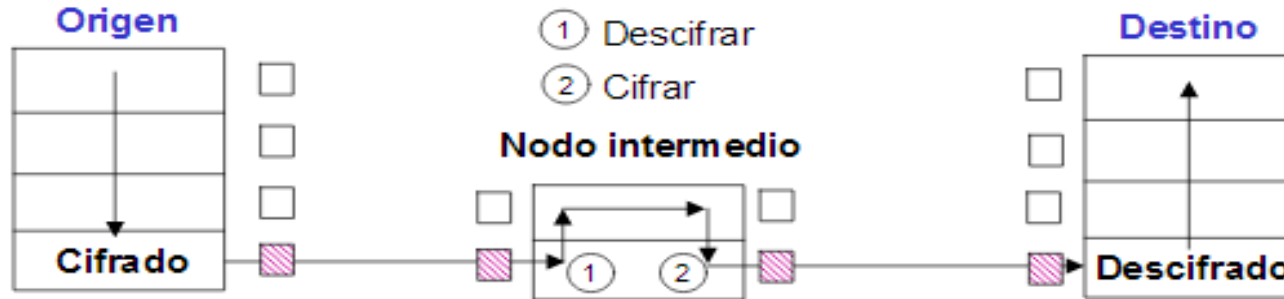
## CORTAFUEGOS (*FIREWALL*) – TÉCNICAS APLICADAS

- **Filtrado de paquetes:**
  - Reglas que especifican qué tipos de paquetes pueden circular en cada sentido y cuáles se bloquearán.
  - Las reglas se basan en las cabeceras de los paquetes.
- **Servicios de proxy:**
  - Son aplicaciones especializadas que funcionan en un cortafuegos.
  - Hacen de intermediarios entre los servidores y los clientes reales.
  - Reciben las peticiones de servicios de los usuarios, las analizan y en su caso modifican, y las transmiten a los servidores reales .
  - Son transparentes al usuario.

# Implementación de mecanismos de seguridad

## CIFRADO EN REDES

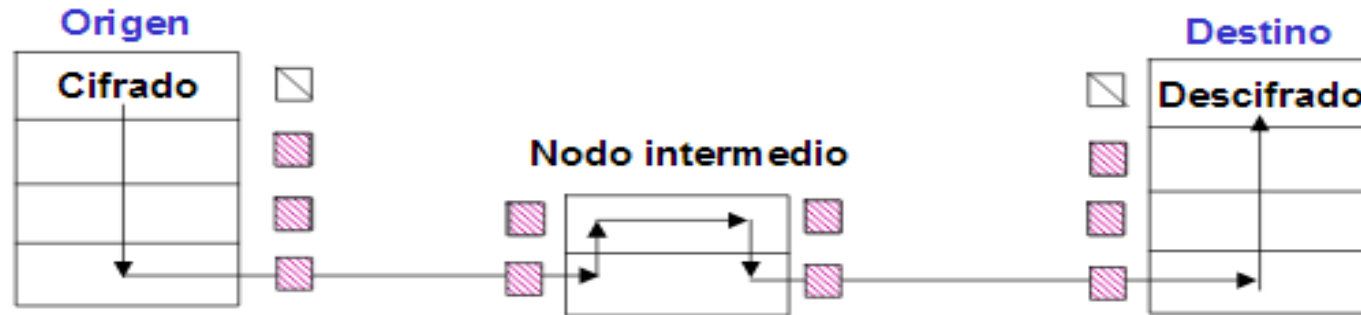
- **Cifrado de enlace:**
  - Capa 2 de OSI
  - Cifra todo el mensaje, incluidas las cabeceras de niveles superiores
  - Requiere nodos intermedios con capacidades de cifrado/descifrado
  - La información está protegida entre cada par de nodos consecutivos (distintas claves para cada par)
  - Es necesario descifrarla, aunque sea parcialmente, para procesos de encaminamiento, control de errores, etc



# Implementación de mecanismos de seguridad

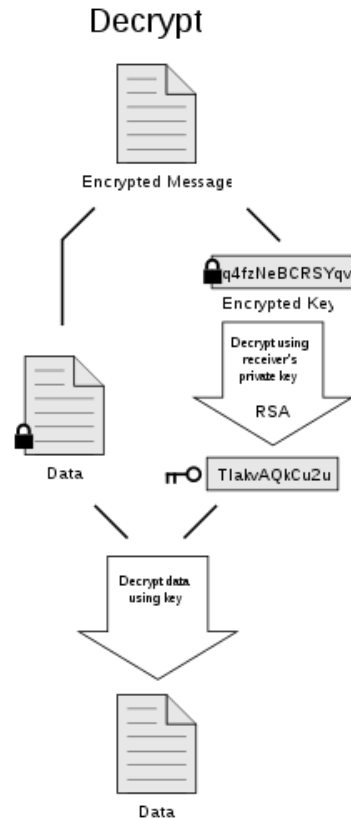
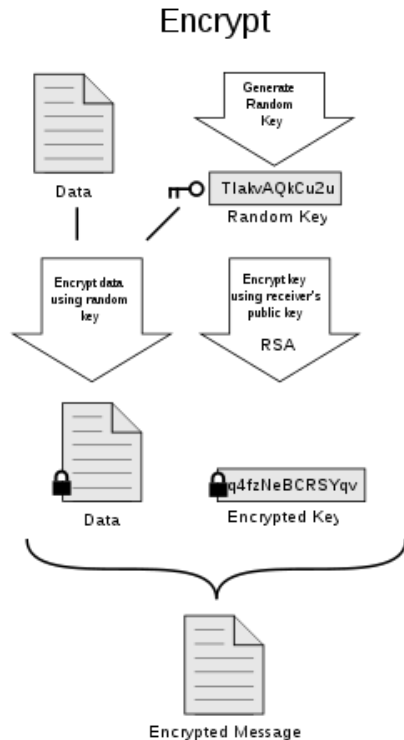
## CIFRADO EN REDES

- **Cifrado extremo a extremo:**
  - Capa 7 de OSI
  - Sólo se cifran los datos, las cabeceras se añaden y se transmiten sin cifrar.



# Implementación de mecanismos de seguridad

**PRETTY GOOD PRIVACY (PGP)** (Usado para correo electrónico seguro y otros documentos en Internet)

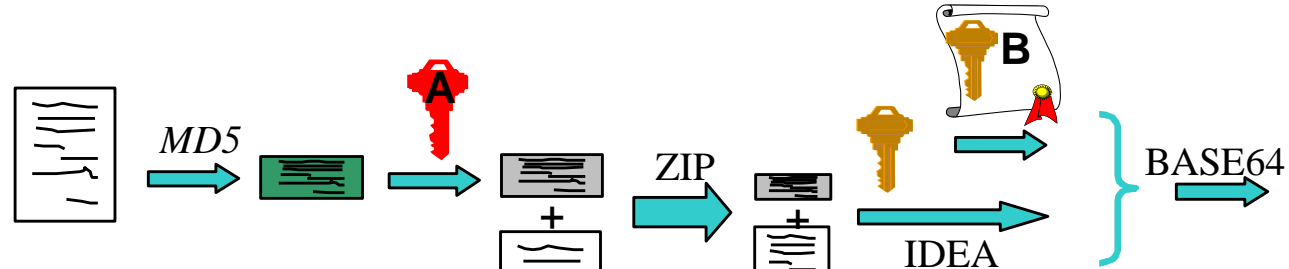


*Se encripta tanto los datos, como la clave para descryptar en el destino esos datos*



# Implementación de mecanismos de seguridad

**PRETTY GOOD PRIVACY (PGP)** (Usado para correo electrónico seguro y otros documentos en Internet)



- | . Resumen/Hash (integridad)
- | . Firma Digital (identidad de A)
- | . Agrupar datos y comprimirlos
- | . Encriptar con clave aleat. K (seguridad)
- | . Encriptar con clave publica de B (confidencialidad)
- | . Codificación adicional B64

Emisor:

- $R = \text{MD5}(P)$
- $\text{FD} = \text{Kpr}_A(R)$
- $Z = \text{ZIP}(\text{FD} + P)$
- $C = \text{IDEA}_K(Z) + \text{Kpu}_B(K)$
- $M = \text{B64}(C)$

Receptor:

- $C = \text{B64}^{-1}(M)$
- $K = \text{Kpr}_B(\text{Kpu}_B(K))$
- $Z = \text{IDEA}_K^{-1}(\text{IDEA}_K(Z))$
- $\text{FD} + P = \text{ZIP}^{-1}(Z)$
- $R = \text{Kpu}_A(\text{FD})$
- $R' = \text{MD5}(P)$
- $R' = R ??$

# Implementación de mecanismos de seguridad

## SSH (Secure Shell)

- SSH es un protocolo de **nivel de aplicación** para crear **conexiones seguras** entre dos sistemas **sobre redes no seguras**.
- Alternativa a programas de **acceso remoto** no seguros, como telnet, ftp, rlogin, rsh y rcp (slogin, ssh y scp).
- Proporciona un **terminal de sesión cifrada con autenticación** fuerte del servidor y el cliente, usando criptografía de clave pública.
- Incluye **características** como:
  - Variedad de mecanismos de autenticación de usuarios (incluyendo autenticación externa Kerberos).
  - Conexiones TCP arbitrarias de *tunneling* a través de la sesión SSH, protegiendo protocolos inseguros como IMAP y permitiendo el paso seguro a través de cortafuegos.
  - Transferencias seguras de ficheros.
  - Soporte para entorno gráfico.

# Implementación de mecanismos de seguridad

## Secuencia de eventos de una conexión SSH

1. Se crea una **capa de transporte segura** para que el cliente sepa que está efectivamente comunicándose con el servidor correcto. Luego **se cifra la comunicación** entre el cliente y el servidor por medio de una clave simétrica/privada.
2. Una vez conectado de forma segura, el **cliente se autentica ante el servidor** sin preocuparse de que la información de autenticación pudiese exponerse.
3. Con el cliente autenticado ante el servidor, se pueden **usar varios servicios diferentes** con seguridad a través de la conexión, como una sesión de terminal interactivo, aplicaciones y túneles TCP/IP.

# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (SSL/TLS)

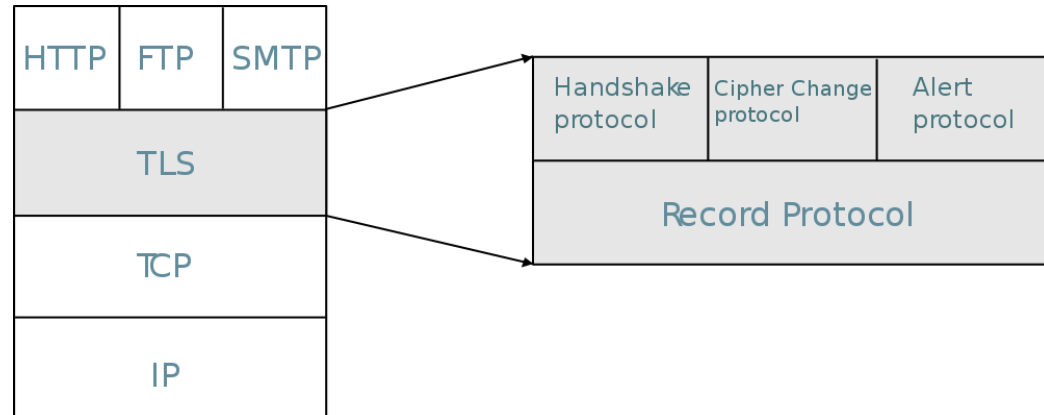
- **SSL (Secure Socket Layer)** → Desarrollado por Netscape en 1994 y puesto en dominio público para la **definición de canales seguros sobre TCP**.
- **TLS (Transport Layer Security)** → Sucesor y **mejora sobre SSL**.
  - **Corrige vulnerabilidades** de SSL y **permite la autenticación** de emisor y receptor.
  - Se basa en el uso de **certificados digitales** para establecer la conexión.
  - Posteriormente emisor y receptor comparten una clave privada.
- Ambos son **protocolos criptográficos** que permiten realizar **comunicaciones seguras sobre una red no segura**.

No funciona  
sobre UDP

# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (SSL/TLS) – Capas

- **SSL Record Protocol** encapsula los protocolos y ofrece un canal seguro con privacidad, autenticación e integridad
- **SSL Handshake Protocol**
  - Negocia el algoritmo de cifrado
  - Negocia la función Hash
  - Autentica al servidor con X.509
  - El cliente genera claves de sesión:
    - Aleatorias cifrada con  $K_{\text{PUB\_SERVER}}$  ó
    - Diffie-Hellman
- **SSL Alert protocol**
  - Informa sobre errores en la sesión
- **Cipher Change Espec Protocol**
  - Para notificar cambios en el cifrado



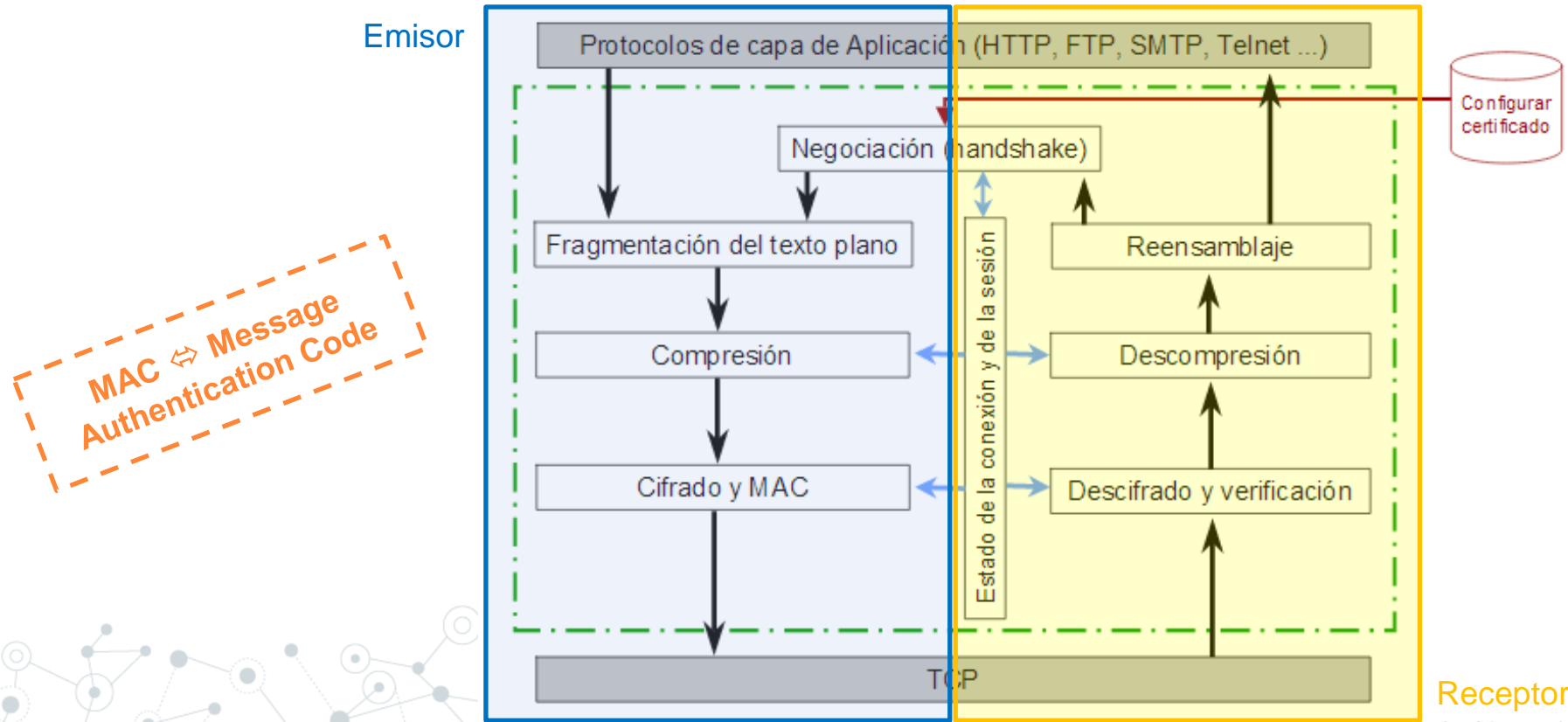
# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (SSL/TLS) – Funcionamiento

- El **cliente** al hacer la **conexión informa** sobre los **sistemas criptográficos que tiene** disponibles, y el **servidor** responde con un **identificador de la conexión**, su **clave certificada** e información sobre los **sistemas criptográficos** que soporta.
- El **cliente elegirá** un **sistema criptográfico** y **verificará** la **clave pública del servidor**.
- Entonces se **generará** una **clave privada** (de uso único) cifrada con la clave pública del servidor. Si alguien pudiese descifrar la información, sólo conseguiría romper esa conexión/sesión, ya que una sesión posterior requeriría una clave privada diferente.
- Una vez **finalizado este proceso**, los **protocolos** toman el control de **nivel de aplicación**, de modo que SSL/TLS nos asegura que:
  - Los mensajes que enviamos o recibimos no han sido modificados (integridad).
  - Ninguna persona sin autorización puede leer la información transmitida (confidencialidad).
  - Efectivamente envía/recibe la información quien debe enviarla/recibirla (autenticación).

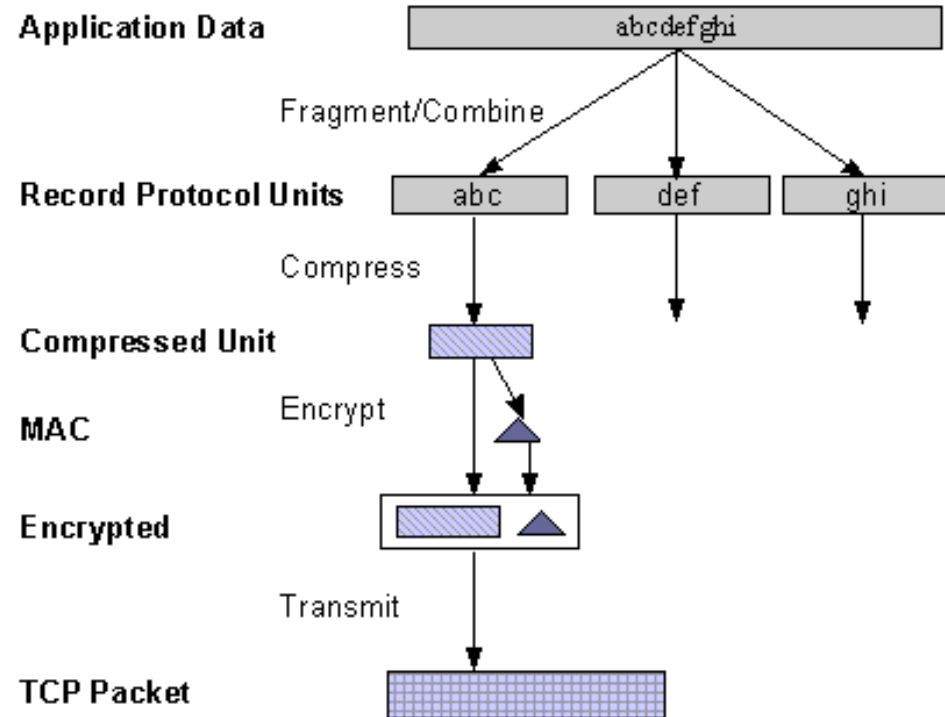
# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (SSL/TLS) – Arquitectura



# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (TLS/SSL)





# Implementación de mecanismos de seguridad

## TRANSPORT LAYER SECURITY (SSL/TLS)

- Versión actual SSL 3.0
- SSL es capaz de trabajar de forma transparente con todos los protocolos que trabajan sobre TCP
- Para ello el IANA tiene asignado un número de puerto por defecto a cada uno de ellos:

Identificador de protocolo	Puerto TCP	Descripción
https	443	HTTP sobre SSL
smtps	465	SMTP sobre SSL
nhttps	563	NTTP sobre SSL
ladps	646	LDAP sobre SSL
telnets	992	TELNET sobre SSL
imaps	993	IMAP sobre SSL
ircs	994	IRC sobre SSL
pop3s	995	POP3 sobre SSL
ftps-data	989	FTP-Datos sobre SSL
ftps-control	990	FTP-Control sobre SSL

# Implementación de mecanismos de seguridad

## IPSec (IP Security)

- Proporciona **seguridad en la capa de red** y a las superiores que se apoyen en IP (RFC 2401).
- Su objetivo es garantizar **autenticación**, **integridad** y (opcionalmente) **privacidad** a nivel IP.
- IPSec consiste en 3 procedimientos:
  - 1) Establecimiento de una “**Asociación de seguridad**”: IKE (Internet Key Exchange, RFC 2409)
    - Objetivo: establecimiento de clave secreta (**Diffie-Hellman**).
    - Incluye previamente **autenticación** (con certificados) para evitar el ataque de MitM.
    - Es **simplex**: la asociación de seguridad tiene un único sentido.
    - Se **identifica** con la IP origen + Security Parameter Index (32 bits).
    - **Vulnera** el carácter NO orientado a conexión de IP.
  - 2) Garantizar la **autenticación** e **integridad** de los datos:  
protocolo de “Cabeceras de autenticación” (RFC 2401)
  - 3) (Opcional) Garantizar la **autenticación** e **integridad** y **privacidad** de los datos:  
protocolo de “**Encapsulado de seguridad de la carga**” (RFC 2411)

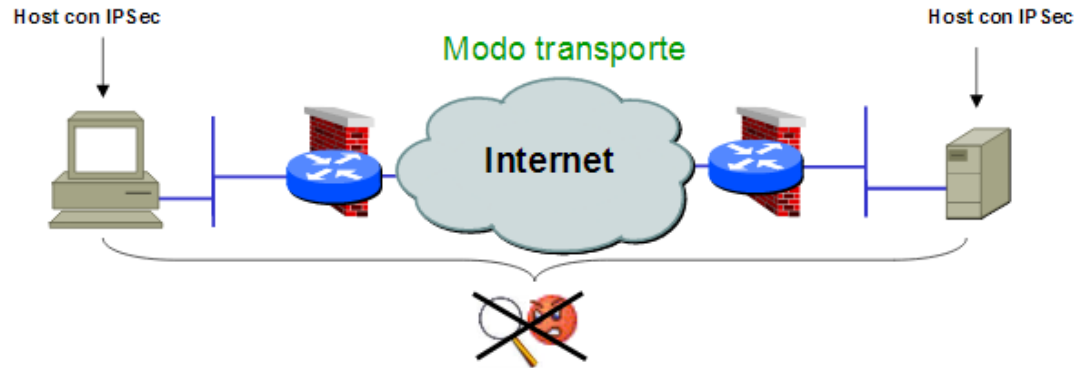
# Implementación de mecanismos de seguridad

## IPSec (IP Security)

- IPSec tiene 2 modos de operación:

1) **Modo Transporte:** la asociación se hace extremo a extremo entre en host origen y host destino.

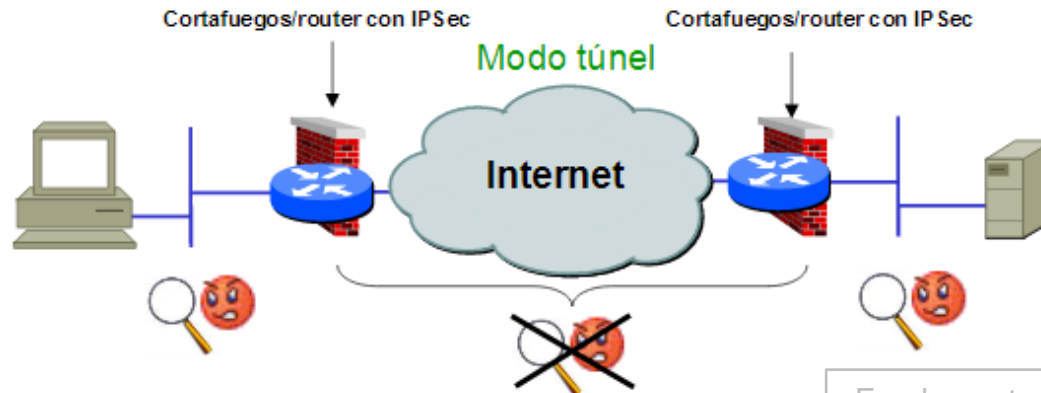
- se protege la carga útil IP (payload) (capa de transporte)
- comunicación segura extremo a extremo
- requiere implementación de IPSec en ambos hosts



# Implementación de mecanismos de seguridad

## IPSec

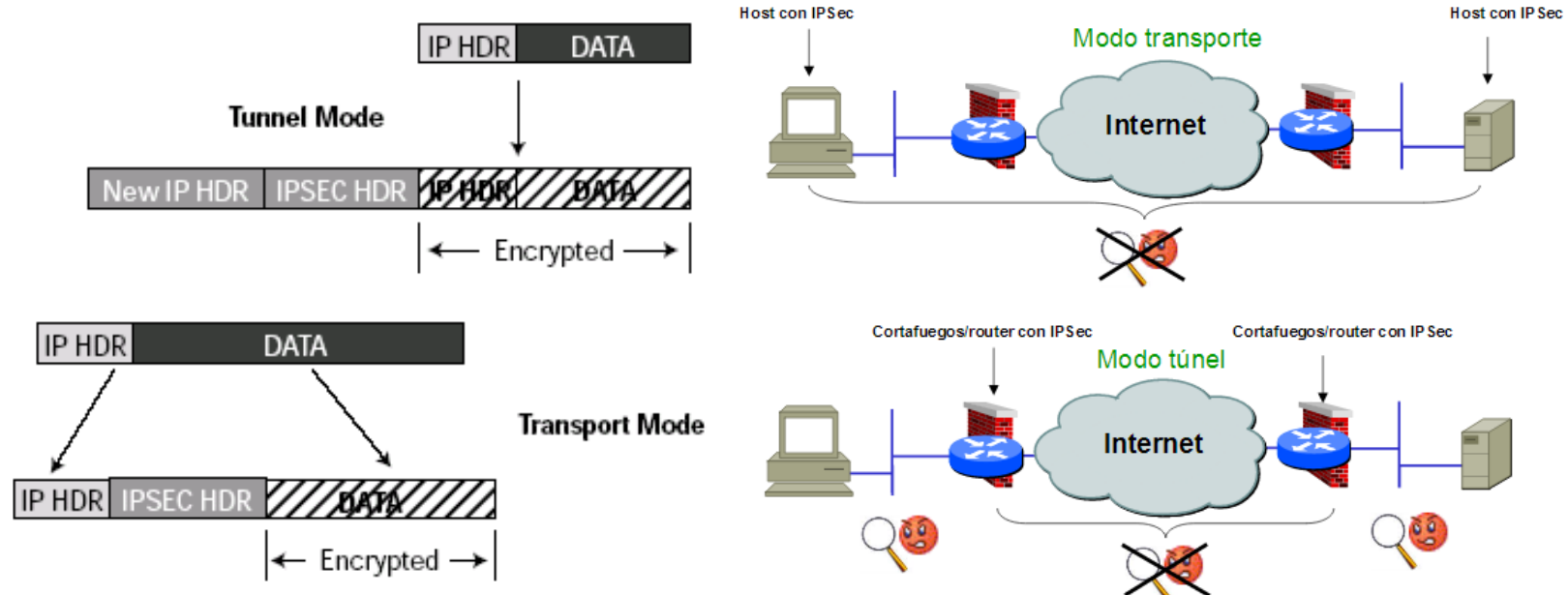
- IPSec tiene 2 modos de operación:
  - 1) **Modo Túnel:** la asociación se hace entre dos routers intermediarios.
    - se protegen paquetes IP (capa de red)
    - para la comunicación segura entre routers/gateways de seguridad sólo se puede usar este modo
    - permite incorporar IPSec sin afectar a los hosts
    - se integra fácilmente con VPNs



# Implementación de mecanismos de seguridad

## IPSec

- IPSec tiene 2 modos de operación:
  - 1) **Modo Transporte:** la asociación se hace extremo a extremo entre el host origen y el host destino
  - 2) **Modo Túnel:** la asociación se hace entre dos routers intermediarios



# ¿Preguntas?

0 comentarios, sugerencias, inquietudes