

# TEMA 4

# SEGURIDAD EN REDES

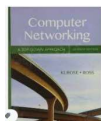
Fundamentos de Redes  
2021/2022



ugr

Universidad  
de Granada

## ➤ Bibliografía Básica:



Capítulo 8 James F. Kurose y Keith W. Ross. ***COMPUTER NETWORKING. A TOP-DOWN APPROACH***, 7ª Edición, Addison-Wesley, 2017, ISBN 9780133594140



Capítulo 12, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler.  
***TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES***, Ed. Pearson, 2017, ISBN:  
978-0-273-76896-8.

## ➤ Para saber más



➤Capítulo 8 Andrew S. Tanenbaum, David J. Wetherall: ***REDES DE COMPUTADORES***, 5ª Edición, PEARSON EDUCACIÓN, 2012 ISBN: 978-607-32-0817-8

## ➤ Agradecimientos:

Estas transparencias están inspiradas en las transparencias utilizadas por Kurose y Ross en de la Universidad de Massachusetts.

## Tema 4. SEGURIDAD EN REDES

### 1. Introducción

2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.

## 1. INTRODUCCIÓN

- Una red de comunicaciones es **segura** cuando se garantizan **todos** los aspectos → no hay protocolos ni redes 100% seguros
- ¿Qué es seguridad? → múltiples aspectos:
  - **Confidencialidad/privacidad:** el contenido de la información es comprensible sólo por entidades autorizadas.
  - **Autenticación:** las entidades son quien dicen ser.
  - **Control de accesos:** los servicios están accesibles sólo a 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.

## 1. INTRODUCCIÓN

- ¿En qué nivel/capa se debe situar la seguridad? en **TODOS**.....*el grado de seguridad lo fija el punto más débil*
- Ataque de seguridad: cualquier acción intencionada o no que menoscaba cualquiera de los aspectos de la seguridad
- Tipos de ataques:
  - Sniffing* = vulneración a la confidencialidad, escuchas (husmear)
  - Spoofing (phishing)* = suplantación de la identidad de entidades
  - Man\_in\_the\_middle* = hombre en medio (interceptación)
  - Distributed Denial\_of\_Service (DDoS)* = denegación de servicio distribuido, ejemplo *Flooding* (inundación)
  - Malware* = troyanos, gusanos, *spyware*, *backdoors*, *rootkits*, *ransomware*, *keyloggers*

## 1. 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 (Intruder Detection System)
- De **recuperación**:
  - copias de seguridad (backup).
  - mecanismos de análisis forense: averiguar alcance, las actividades del intruso en el sistema y cómo entró.

## 1. INTRODUCCIÓN

### ➤ Herramientas para la Seguridad

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

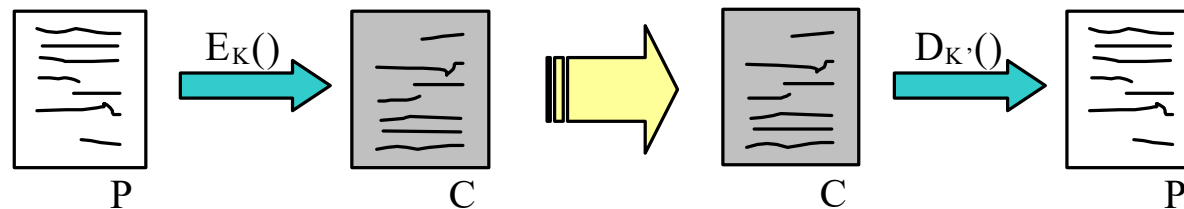
## Tema 4. SEGURIDAD EN REDES

1. Introducción
- 2. Cifrado (simétrico y asimétrico)**
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.



## 2. CIFRADO

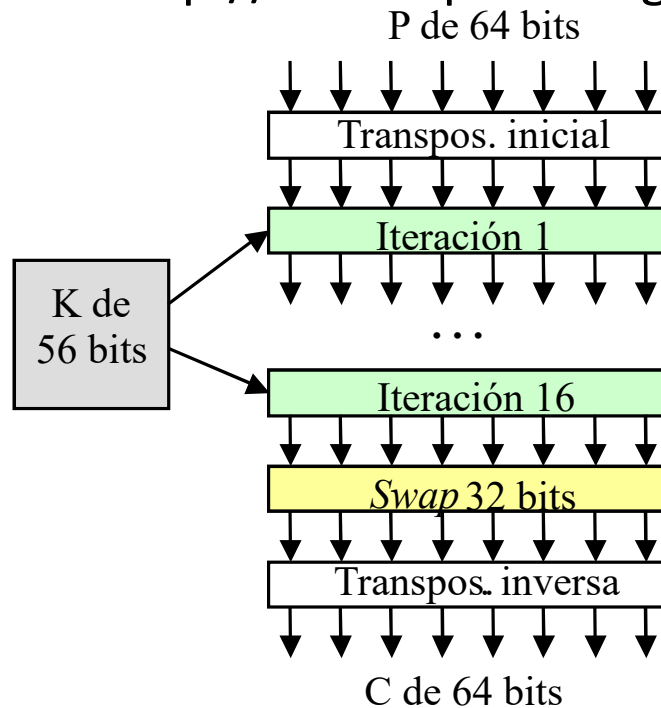
- **Cifrado de datos:**
  - Basado en Criptografía
  - Procedimiento para garantizar la **confidencialidad**
  - Texto llano/claro,  $P \rightarrow$  texto cifrado,  $C$
  - Se basa en la existencia de un **algoritmo** de cifrado/descifrado, **normalmente conocido**  $E_K()$  y  $D_{K'}()$ . La dificultad reside en la existencia de unas **claves de cifrado ( $K$ ) y descifrado ( $K'$ ) desconocidas.**



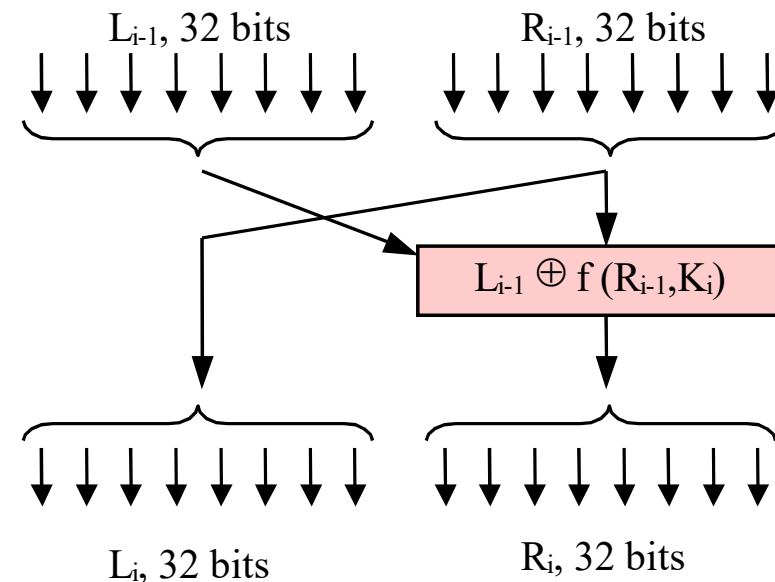
- Tipos:
  - Cifrado simétrico o con clave secreta
  - Cifrado asimétrico o con clave pública/privada

## 2. CIFRADO

- Cifrado simétrico, algoritmos de clave secreta:
  - Existe una sola clave para cifrar y descifrar ( $k=k'$ )
  - **DES** ("Data Encryption Standard", IBM 1975):  
[http://en.wikipedia.org/wiki/Feistel\\_network](http://en.wikipedia.org/wiki/Feistel_network)  
[http://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Data_Encryption_Standard)



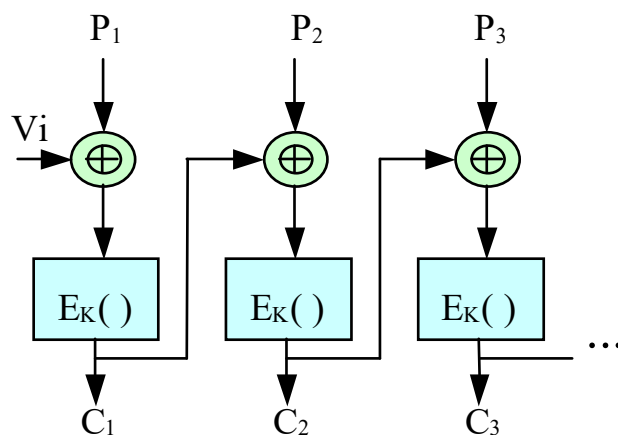
(a)



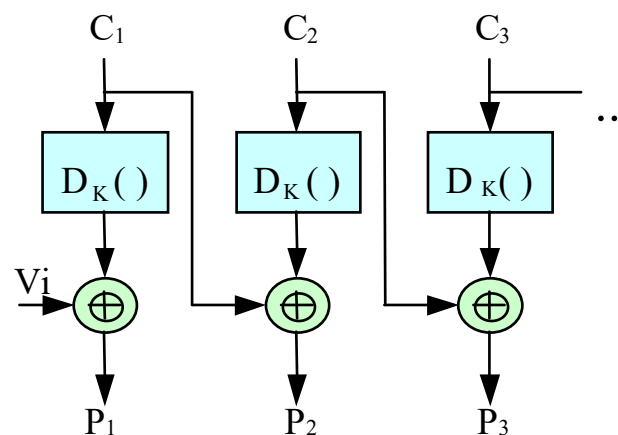
(b)

## 2. CIFRADO

- **DES**: Es un esquema de sustitución monoalfabético
- Encadenamiento DES (para evitar que DES sea un algoritmo de sustitución):

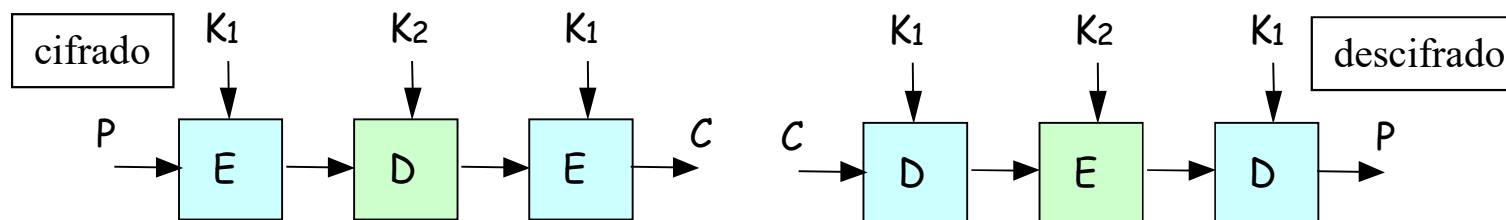


(a)



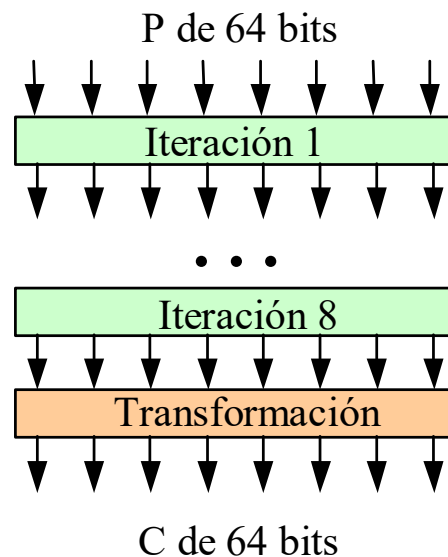
(b)

- Mejorar la robustez: **DES doble y 3DES**:

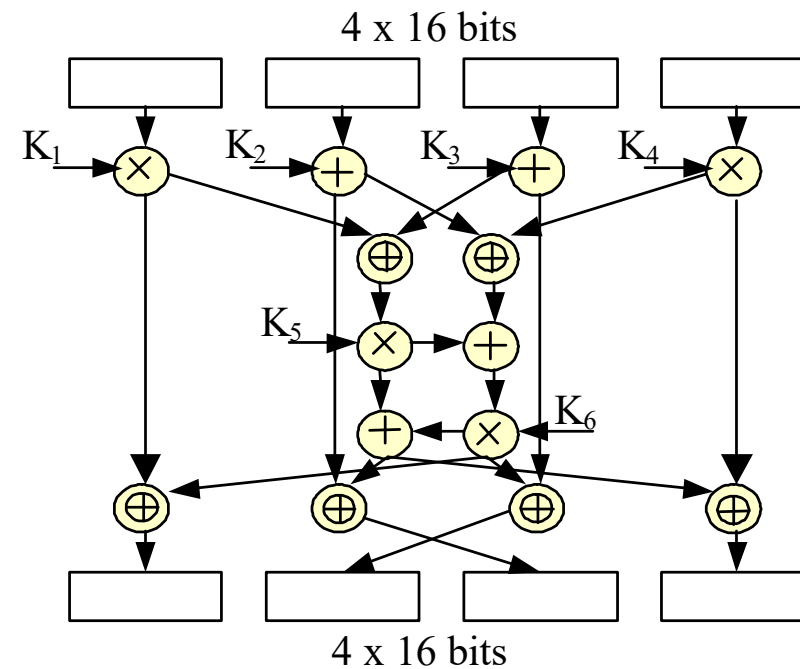


## 2. CIFRADO

- **IDEA** (“International Data Encryption Algorithm”):
  - Simétrico: misma clave para cifrar y para descifrar
  - Claves de 128 bits
  - Opera en tiempo real (VLSI).



(a)



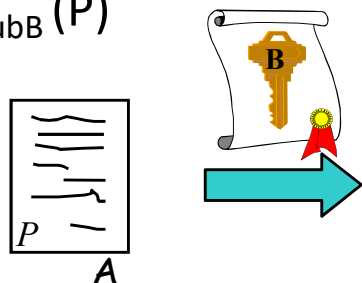
(b)

## 2. CIFRADO

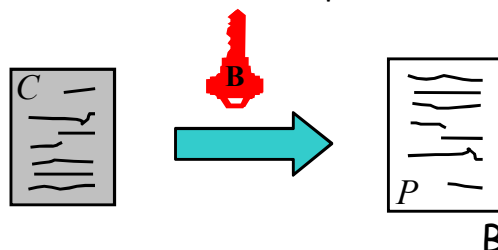
### Cifrado asimétrico, algoritmos de clave pública/privada:

- Dos claves por usuario (A): una pública  $K_{PU_A}$  y otra privada  $K_{PRI_A}$  distintas
- Conocida  $K_{PUB_A}$  es imposible conocer  $K_{PRI_A}$
- Claves diferentes para cifrar y descifrar:

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



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

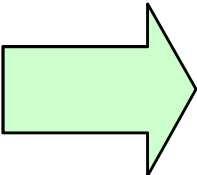





- ¿Y si enviamos  $C = E_{K_{privA}}(P)$  ?  $\rightarrow$  ¿sirve como autenticación ?
- **RSA** (Rivest, Shamir y Adleman):
  - Elegimos  $p$  y  $q$  primos grandes ( $>10^{100}$ )
  - $n = (p \cdot q)$  y  $z = (p-1) \cdot (q-1)$  (función de Euler)
  - Elegimos  $d$  primo respecto de  $z$
  - Calculamos  $e$  tal que  $e \cdot d \bmod z = 1$  (algoritmo de Euclides)
  - $K_{pub} = (e, n)$  y  $K_{pri} = (d, n)$ , de modo que:
    - \*  $C = P^e \bmod n$
    - \*  $P = C^d \bmod n$

## 2. CIFRADO

## Ejemplo RSA:

- $p = 3, q = 11$
- $n = p.q = 33, z = (n-1)(p-1) = 20 = 5 \times 2 \times 2$
- $d = 7$ , primo respecto de  $z$
- $e = 3, e \times d \bmod z = 1$
- $K_{pub} = (3, 33)$  y  $K_{pri} = (7, 33)$

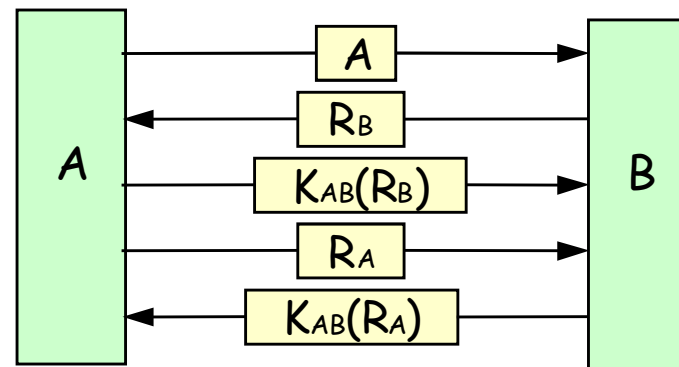
<u>Simbólico</u>	<u>Numérico</u>	<u>P<sup>3</sup></u>	<u>P<sup>3</sup> mod 33</u>		<u>C<sup>7</sup></u>	<u>C<sup>7</sup> mod 33</u>	<u>Simbólico</u>
S	19	6859	28		13492928512	19	S
U	21	9261	21		1801088541	21	U
Z	26	17576	20		1280000000	26	Z
A	01	1	1		1	01	A
N	14	2744	5		78125	14	N
N	14	2744	5		78125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

## Tema 4. SEGURIDAD EN REDES

1. Introducción
2. Cifrado (simétrico y asimétrico)
- 3. Autenticación con clave secreta (reto-respuesta)**
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.

### 3. AUTENTICACIÓN CON CLAVE SECRETA (RETO-RESPUESTA)

- **Autenticar** consiste en verificar de forma fehaciente (sin fisuras) la identidad de las entidades involucradas: “A es quien dice ser”.
- Un procedimiento de autenticación y cifrado basado en clave secreta: el esquema **reto-respuesta**



- Propuesta de ejercicio → ¿Es posible simplificar el procedimiento a 3 mensajes? → ¿Es vulnerable a un Ataque por reflexión? → Usar espacios de claves disjuntos
- Recomendación: usar **nonces** → información que sólo se puede usar una vez

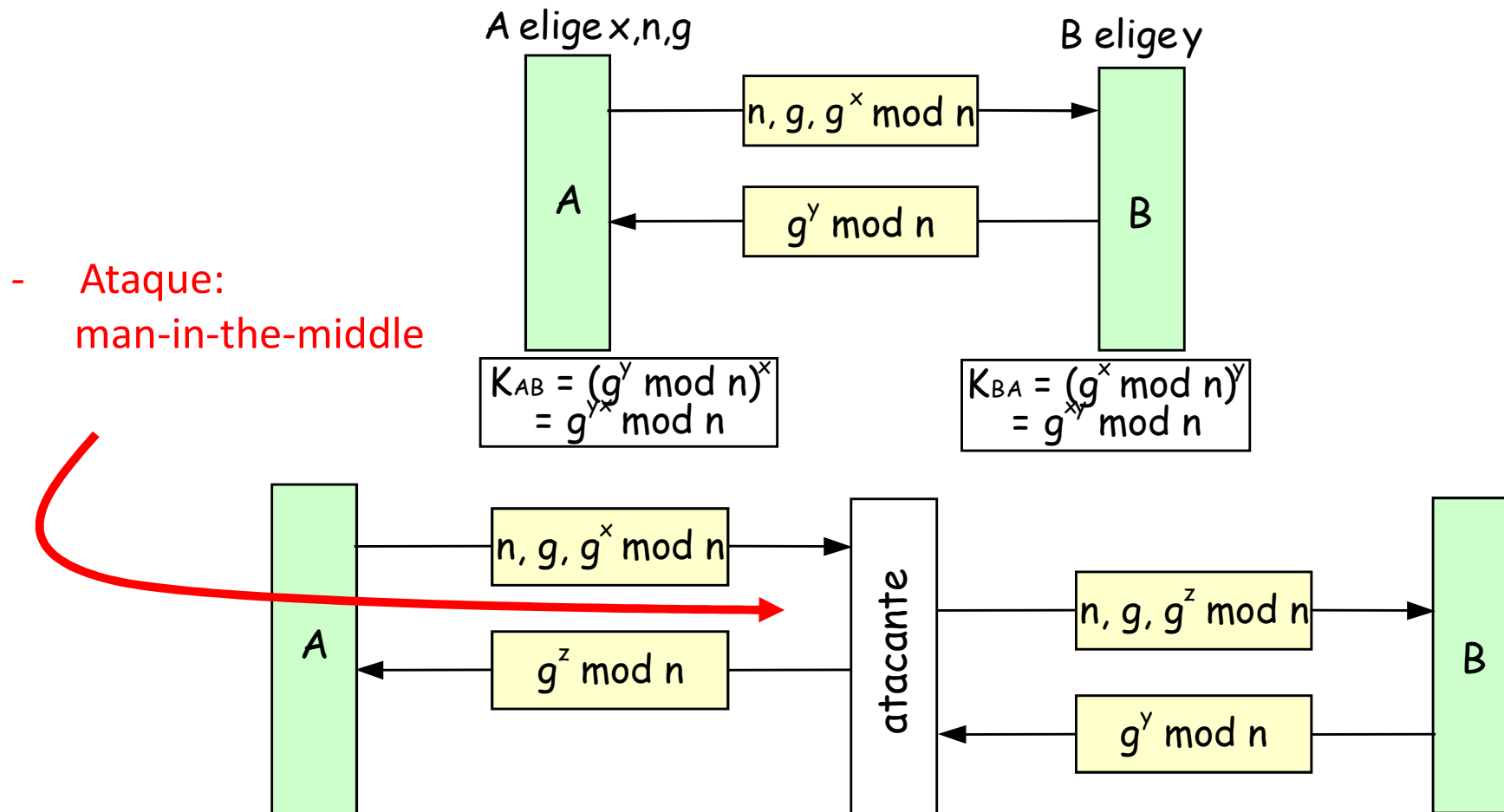


## Tema 4. SEGURIDAD EN REDES

1. Introducción
2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
- 4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)**
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.

#### 4. INTERCAMBIO DE DIFFIE-HELLMAN (ESTABLECIMIENTO DE CLAVE SECRETA)

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



## Tema 4. SEGURIDAD EN REDES

1. Introducción
2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
- 5. Funciones Hash. Hash Message Authentication Code (HMAC).**
6. Firma Digital.
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.

## 5. FUNCIONES HASH. HASH MESSAGE AUTHENTICATION CODE (HMAC).

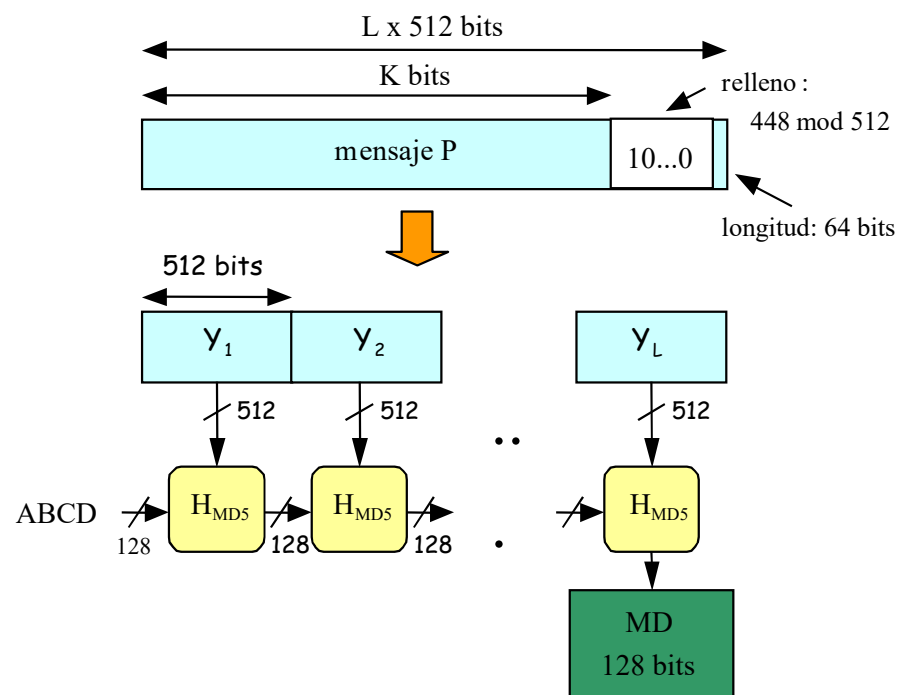
- **Funciones Hash (compendios).** Características de los compendios:
  - Funciones unidireccionales (irreversibles) de cálculo sencillo
  - Texto de entrada (M) de longitud variable
  - $M \rightarrow H(M)$  siendo  $H(M)$  de longitud fija (256 ó 512 bits)
  - Imposible obtener M a partir de su resumen  $H(M)$
  - Invulnerables a ataques de colisión, dado M es imposible encontrar  $M'$  /  $M \neq M'$  y  $H(M) = H(M')$
  - Ejemplos de funciones HASH: MD5, SHA-1, SHA-512
  - Las funciones Hash se usan para garantizar integridad + autenticación.  
Hash Message Authentication Code (HMAC):  $M + H(K|M)$  pero para evitar ataques de extensión se usa  $M + H(K | H(K | M))$

## 5. FUNCIONES HASH. HASH MESSAGE AUTHENTICATION CODE (HMAC).

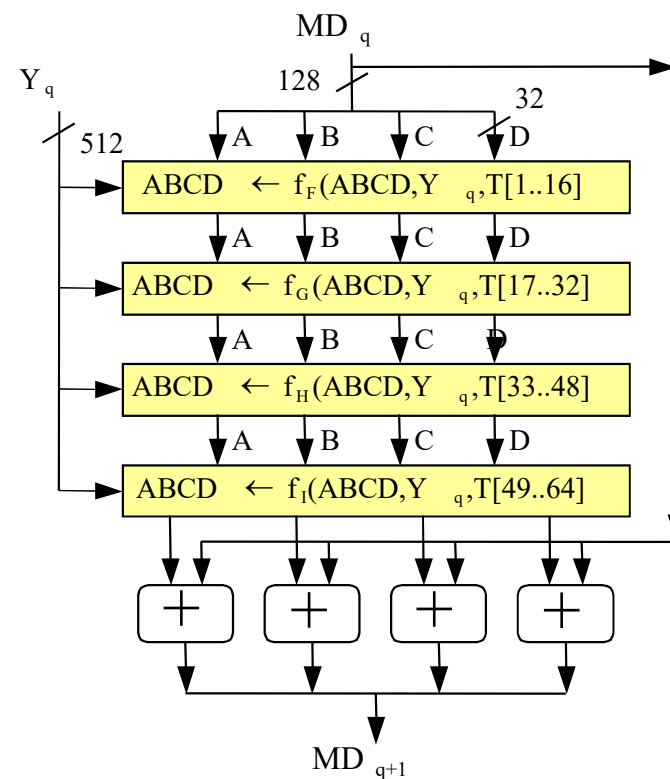
MD5 ("Message Digest 5", RFC 1321):

○ **Proceso** (resumen de 128 bits):

- Relleno 100..0 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



Proceso MD5 general

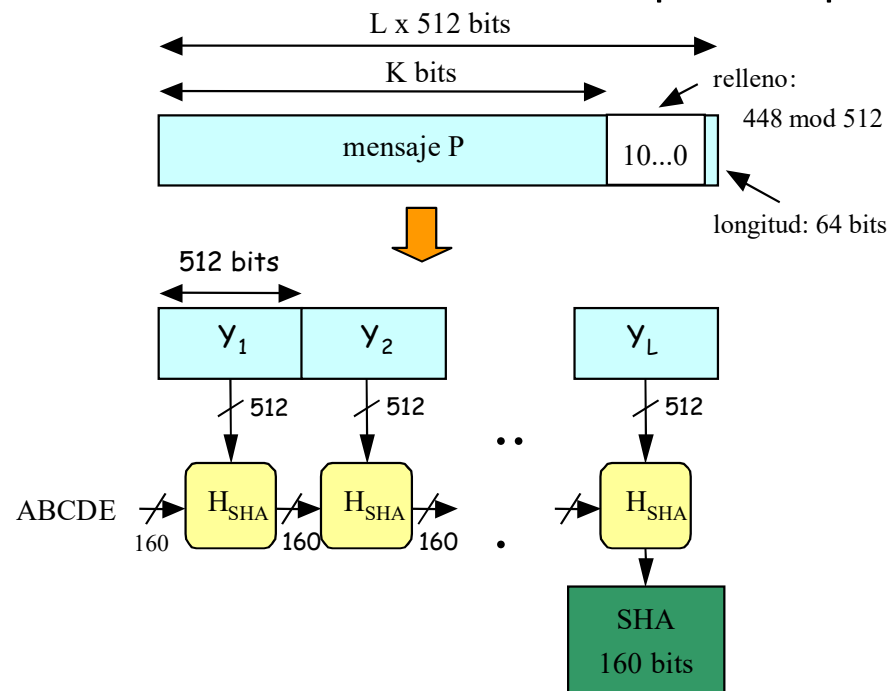
Proceso para un bloque  $q$

# PROTOCOLOS SEGUROS: MECANISMOS DE SEGURIDAD – FUNCIONES HASH

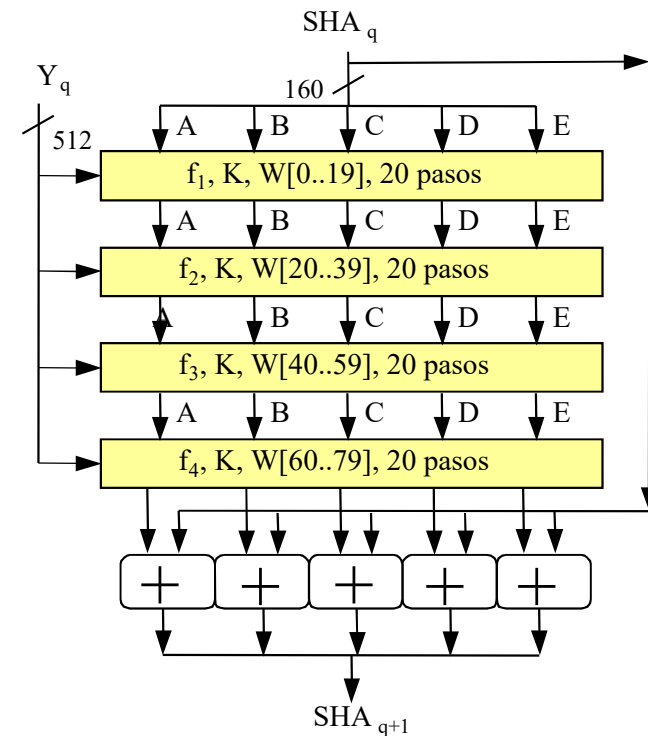
## SHA-1 ("Secure Hash Algorithm 1", NIST 1993):

### Proceso (resumen de 160 bits):

- Relleno 100..0 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



Proceso SHA-1 general



Proceso para un bloque  $q$

## Tema 4. SEGURIDAD EN REDES

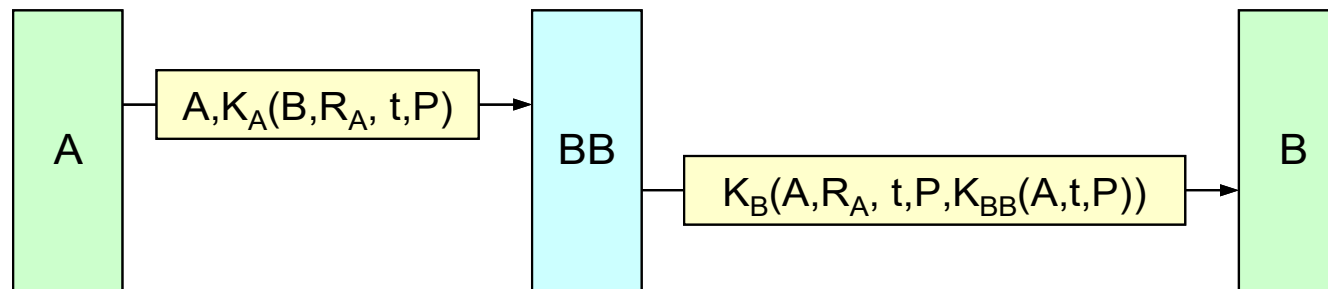
1. Introducción
2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
- 6. Firma Digital.**
7. Certificados digitales.
8. Protocolos seguros: implementación de mecanismos de seguridad.

## 6. FIRMA DIGITAL

### ➤ Objetivos de la firma digital

- El receptor pueda autenticar al emisor (el firmante)
- No haya repudio (irrenunciabilidad)
- El emisor (el firmante) tenga garantías de no falsificación (integridad) por parte del destinatario

### ➤ Firma digital con clave secreta: *Big Brother*:

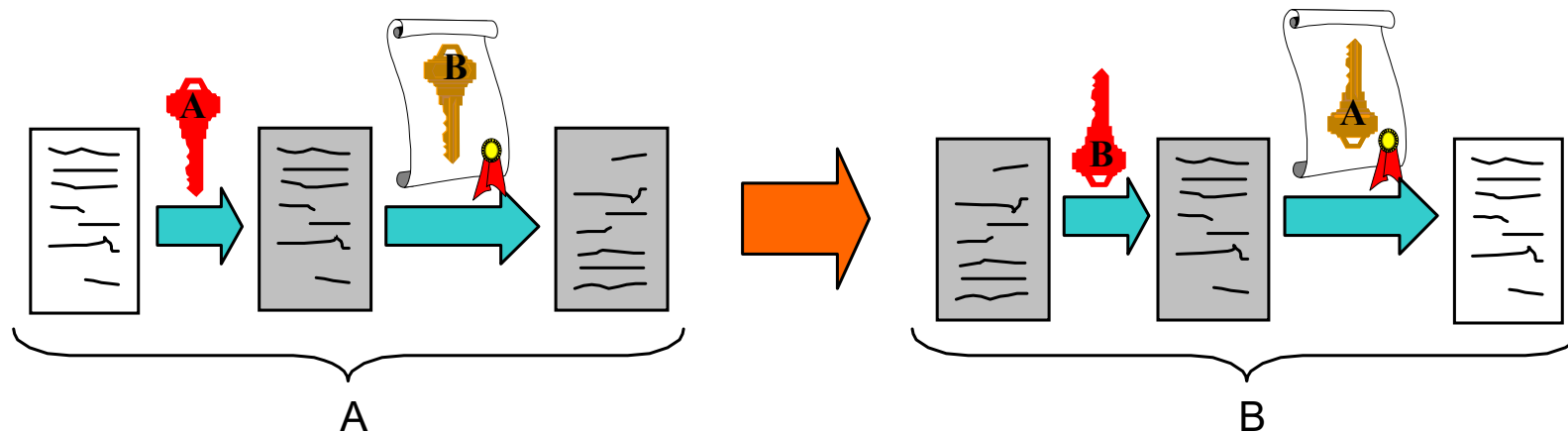




## 6. FIRMA DIGITAL

➤ Firma digital con clave asimétrica. *Doble cifrado*:

- Uno para proporcionar privacidad, con  $K_{pubB}$
- Otro, previo, para autenticación, con  $K_{priA}$
- Para firmar, enviar  $K_{pubB}(K_{priA}(T)) \rightarrow$
- En el receptor  $K_{pubA}(K_{priB}(K_{pubB}(K_{priA}(T))))=T$



- Debilidad: para garantizar el no repudio se necesita garantizar la asociación fehaciente e indisoluble de la “identidad A” con su “clave pública  $K_{pubA}$ ” ( $A \leftrightarrow K_{pubA}$ ) ... ? esto se consigue con un “certificado digital”

## Tema 4. SEGURIDAD EN REDES

1. Introducción
2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
- 7. Certificados digitales.**
8. Protocolos seguros: implementación de mecanismos de seguridad.

## 7. CERTIFICADOS DIGITALES

- Un **certificado digital** sirve para garantizar fehacientemente la asociación “identidad – clave pública”
- Se necesita la intervención de una **Autoridad de certificación (AC)**:
  - **AC** = Entidad para garantizar la **asociación** entre **identidad** y sus **claves**:
    - 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 norma **X.509**
- 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))

## 7. CERTIFICADOS DIGITALES

❑ Autoridades de certificación (AC):

### *Campos de un 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

## 7. CERTIFICADOS DIGITALES

### ❑ Autoridades de certificación (AC):

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

Validity

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

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

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

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

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

## Tema 4. SEGURIDAD EN REDES

1. Introducción
2. Cifrado (simétrico y asimétrico)
3. Autenticación con clave secreta (reto-respuesta)
4. Intercambio de Diffie-Hellman (establecimiento de clave secreta)
5. Funciones Hash. Hash Message Authentication Code (HMAC).
6. Firma Digital.
7. Certificados digitales.
- 8. Protocolos seguros: implementación de mecanismos de seguridad.**

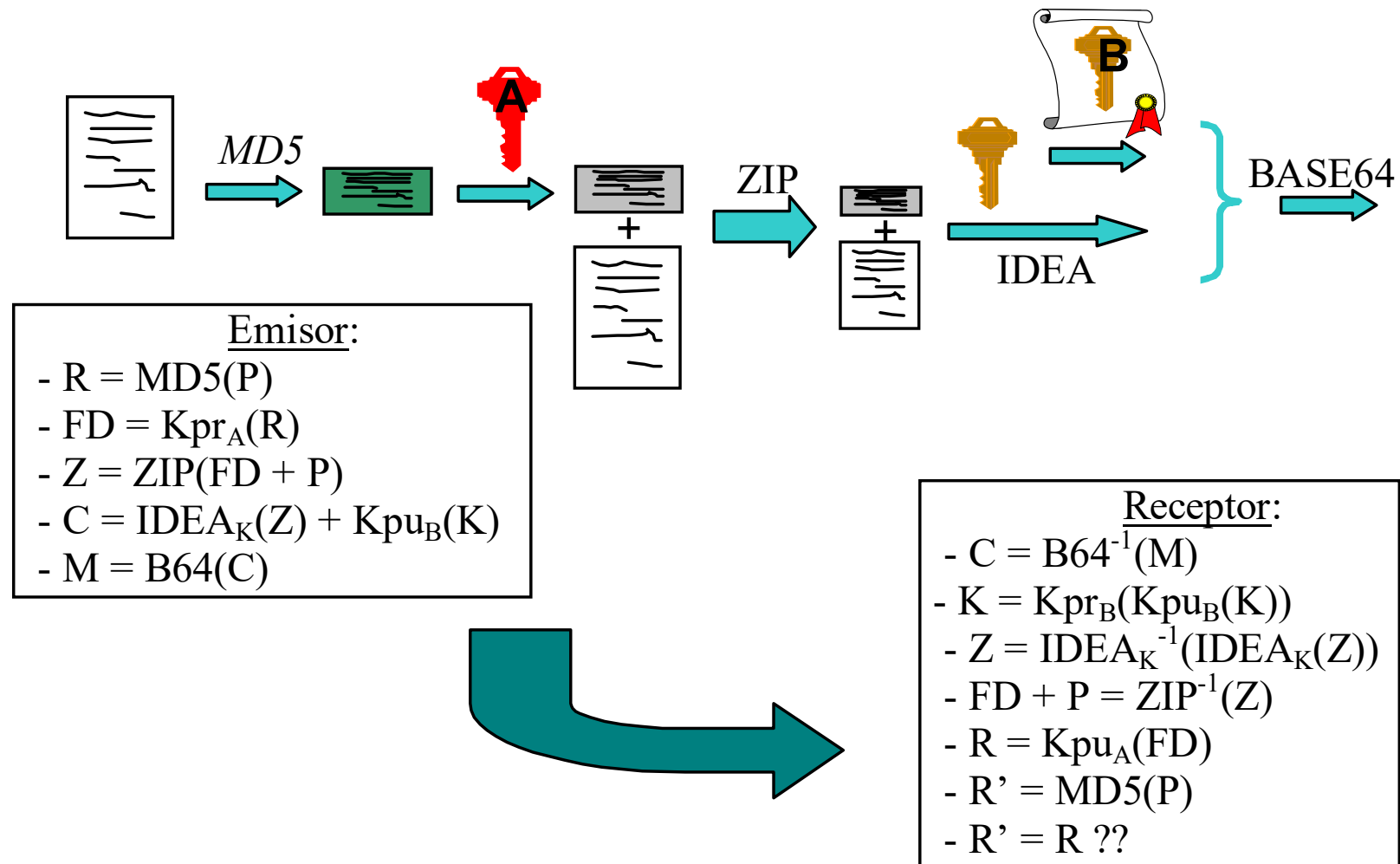
## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

### ➤ Seguridad:

- Seguridad Perimetral: *Firewalls* + sistemas de detección de intrusiones (IDS) y de respuesta (IRS)
- Seguridad en Protocolos (¿dónde poner la seguridad?):
  - Capa de aplicación (ejemplos) ➔
    - Pretty Good Privacy (PGP) – email seguro <https://www.openpgp.org>
    - Secure Shell (SSH) – acceso remoto seguro
    - DNSSEC – DNS con RR (RRSIG, DNSKEY) para firmar las respuestas
  - Capa de transporte ➔
    - Transport Secure Layer (TSL) (antes SSL) ➔ HTTPS, IMAPS, SSL-POP, VPN.  
TLS = Handshake (negociar) + Record Protocol (operación).  
TLS ➔ Confidencialidad ( $K_{\text{secreta}}$  negociada) + Autenticación (para el *server* por defecto con  $K_{\text{PUBLICA}}$ ) + integridad (Con HMAC)
  - Capa de red ➔ IPsec (VPN)
  - Capas inferiores ➔ PAP, CHAP, MS\_CHAP, EAP...

## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

### ➤ Pretty Good Privacy (PGP) – correo electrónico seguro





## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

➤ Transport Secure Layer (TSL) (SSL) ➔ HTTPS, IMAPS, SSL-POP, VPN.

➤ **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 Assert protocol**

- Informa sobre errores en la sesión

➤ **Change Cipher Spec Protocol**

- Para notificar cambios en el cifrado

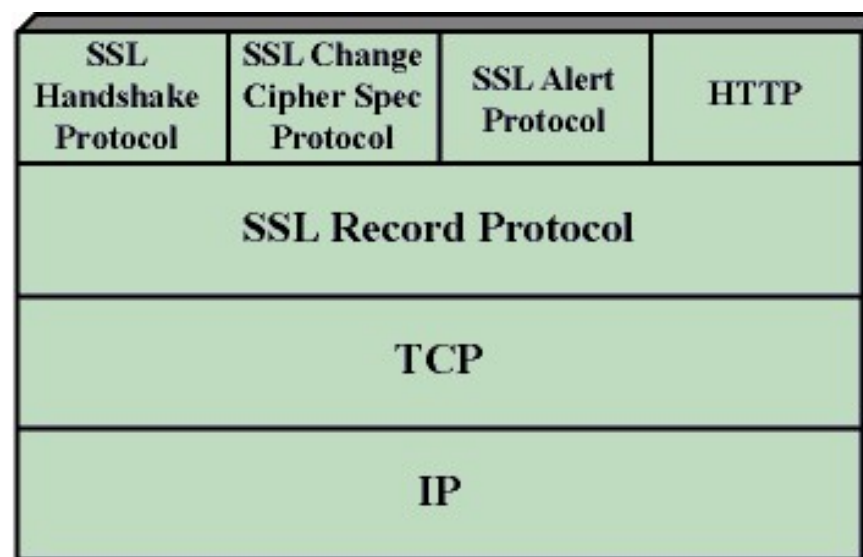
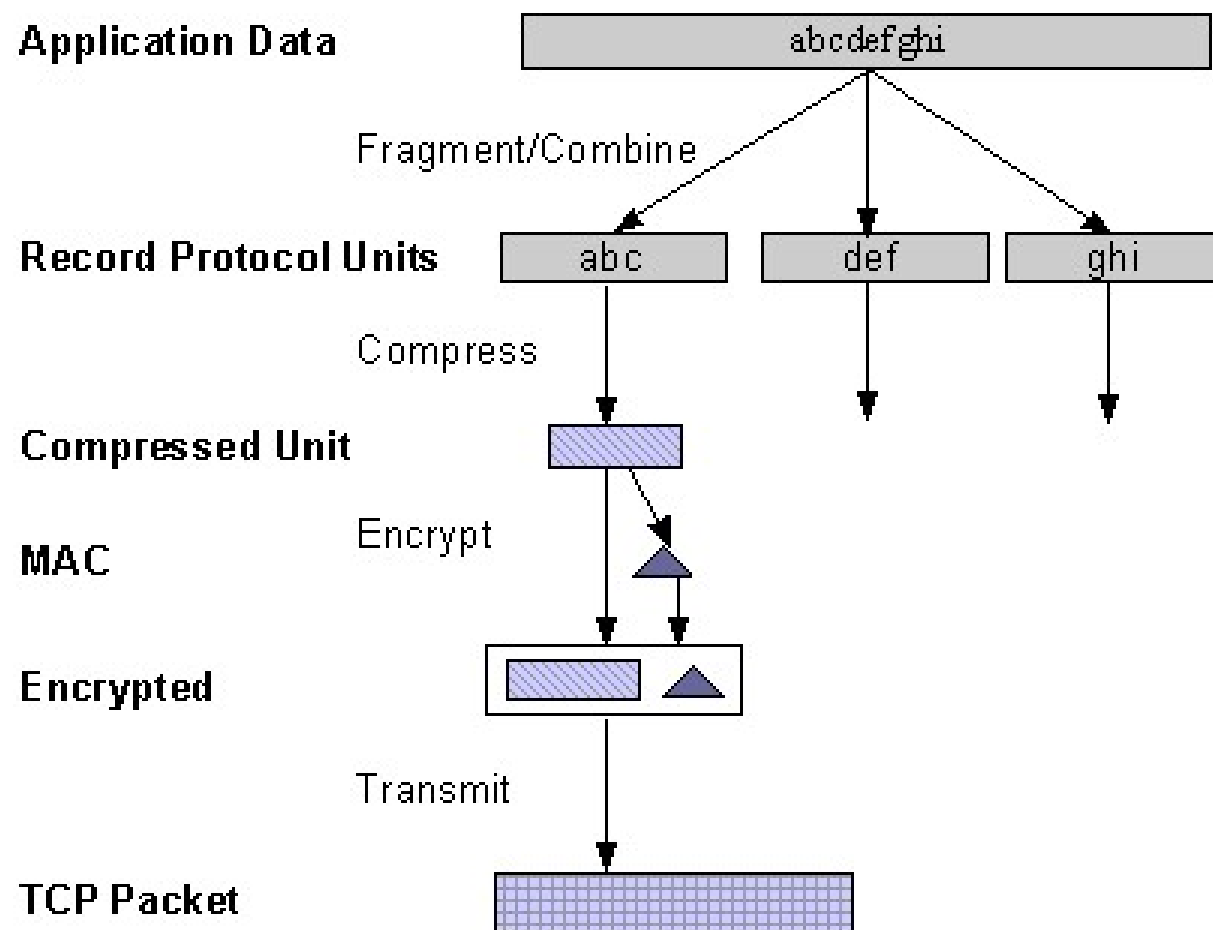


Figure 14.2 SSL Protocol Stack

## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

- Transport Secure Layer (TSL) (SSL) ➔ HTTPS, IMAPS, SSL-POP, VPN.
- **SSL Record Protocol**



## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

- IPsec: su objetivo es garantizar **autenticación**, **integridad** y (opcionalmente) **privacidad** a nivel IP.
- IPsec son 3 procedimientos:
  - 1) Establecimiento de una “**Asociación de seguridad**”: IKE =RFC 2409.
    - Objetivo: establecimiento de clave secreta (**Diffie-Hellman**)
    - Incluye previamente **autenticación** (con certificados) para evitar el ataque de persona en medio
    - 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

## 8. PROTOCOLOS SEGUROS: IMPLEMENTACIÓN DE MECANISMOS DE SEGURIDAD

➤ IPSec tiene 2 modos de operación →

- 1) **Modo Transporte:** la asociación se hace extremo a extremo entre en host origen y host destino
- 2) **Modo túnel:** la asociación se hace entre dos *routers* intermediarios

