PAR – Unidad 7

ADMINISTRACIÓN Y GESTIÓN DE REDES DE ÁREA LOCAL:

SEGURIDAD

Requisitos y ataques

- La **seguridad** en redes implica 4 requisitos :
  - **privacidad -** datos accesibles a quienes tienen permiso
  - **integridad -** datos se mantienen sin alteración
  - **disponibilidad -** datos están siempre accesibles
  - **autenticidad -** datos proceden de quienes dicen
- RFC 2828 - *Internet Security Glossary*:
  - un **ataque** es un asalto a la seguridad de un sistema que deriva de una amenaza inteligente. Puede ser:
    - **activo**, si intenta alterar funciones o recursos del sistema
    - **pasivo**, si sólo intenta averiguar o hacer uso de información
    - **interno** al perímetro de seguridad, con uso no aprobado
    - **externo** al perímetro, por un usuario sin autorización o ilegal

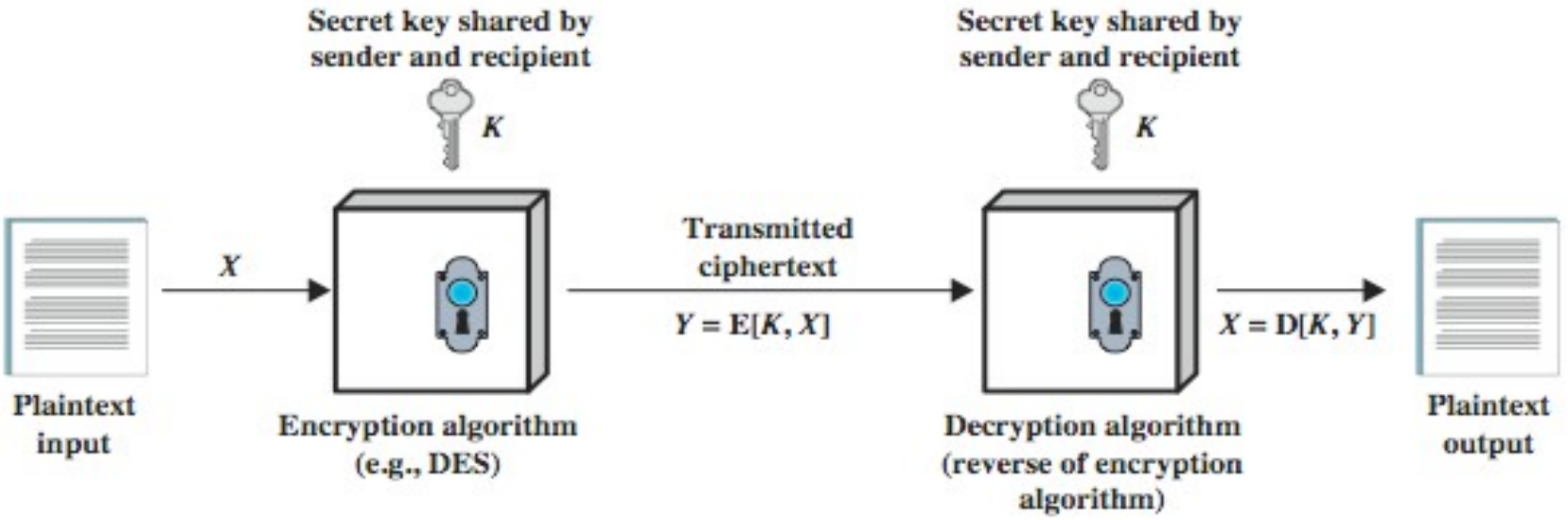
Defensas

- Cuestiones que debe cubrir la política de seguridad:
  - aceptación, registro, actualización y monitorización de dispositivos
  - educación y responsabilidad del usuario
  - seguridad física y en la capa física
  - despliegue y ubicación de la red:
    - pasarelas, vpn, cortafuegos, redundancia
  - contramedidas de seguridad:
    - mínima información al exterior y servicios, encriptar, repeler
  - monitorización de la red y respuesta ante incidentes:
    - IDS, equipo de respuesta y emergencia
  - auditorías de seguridad y estabilidad de la red

Criptografía

- **Criptografía**: ciencia matemática que trata de la transformación (usualmente reversible) de los datos para convertir su significado en ininteligible, prevenir alteraciones no detectadas o su uso no autorizado
- **Criptoanálisis**: ciencia matemática que trata del análisis de un sistema criptográfico para conseguir el conocimiento requerido para romper o saltarse la protección de un sistema criptográfico
- **Esteganografía**: método para ocultar la existencia de un mensaje u objetos dentro de otros, de modo que no se perciba su existencia
- **Criptología**: Criptografía + Criptoanálisis (+ Estegano.)

Criptografía simétrica



- Se basa en someter a sustituciones y trasposiciones el mensaje que se quiere cifrara en función de una misma y única **clave compartida** entre emisor y receptor
- Sistemas rudimentarios: Julio César
- Mejor: clave que dirija el algoritmo
- Kerchoff: secreto de las claves, no de los algoritmos

Requerimientos y ataques

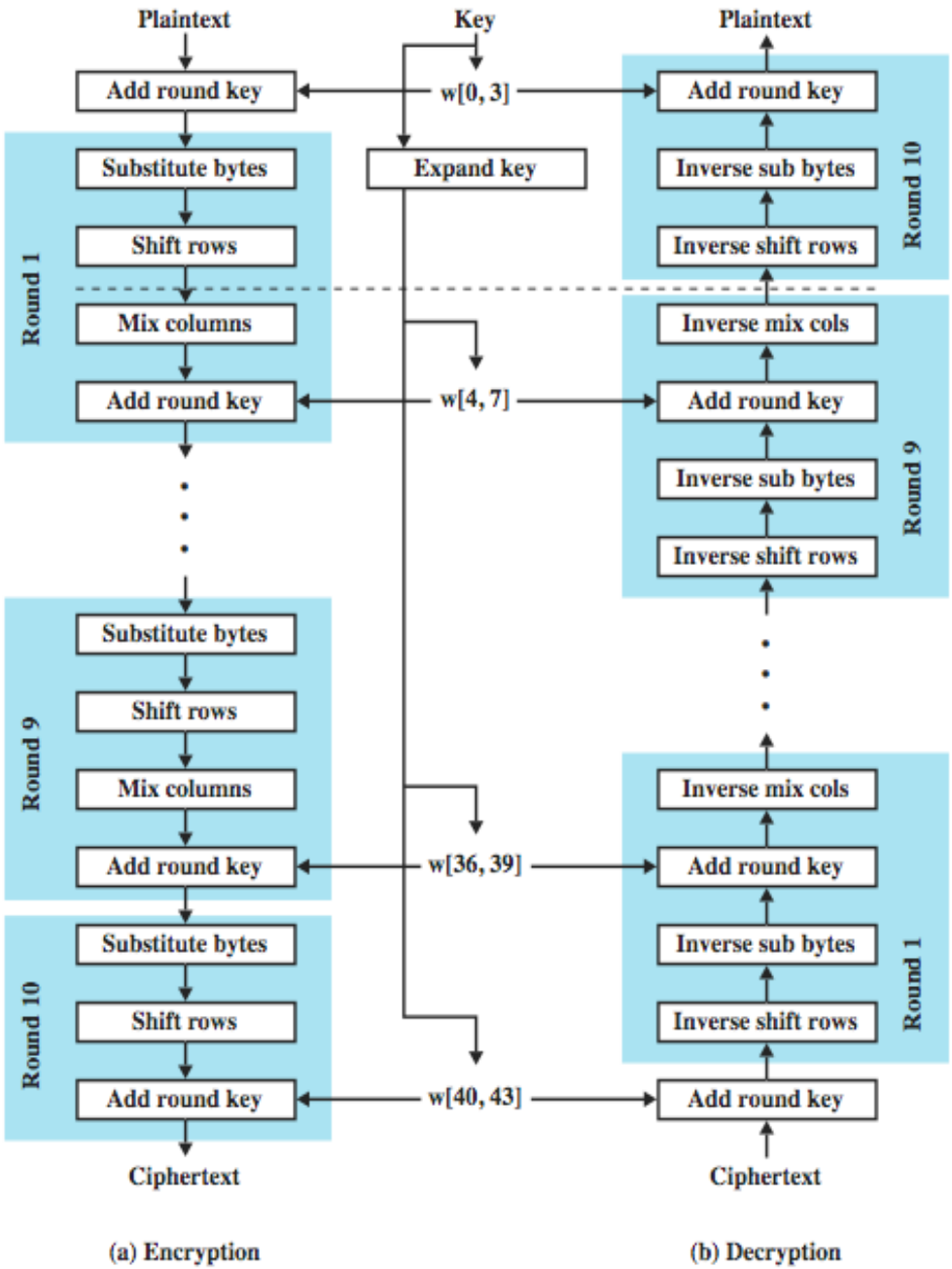
- Algoritmo de encriptación fuerte
  - conocido, que no se pueda desencriptar sin clave
  - incluso si tenemos disponibles muchos mensajes en plano y cifrados
- Emisor y receptor deben obtener la clave de forma segura
- Una vez que la clave es conocida, toda la comunicación usando esa clave es inteligible
- Ataques:
  - criptonálisis,
    - a partir del algoritmo y mensajes en plano y cifrados
  - fuerza bruta,
    - probando todas las combinaciones de clave (se vuelve irrealizable a medida que el tamaño de clave aumenta)

Cifrado por bloque

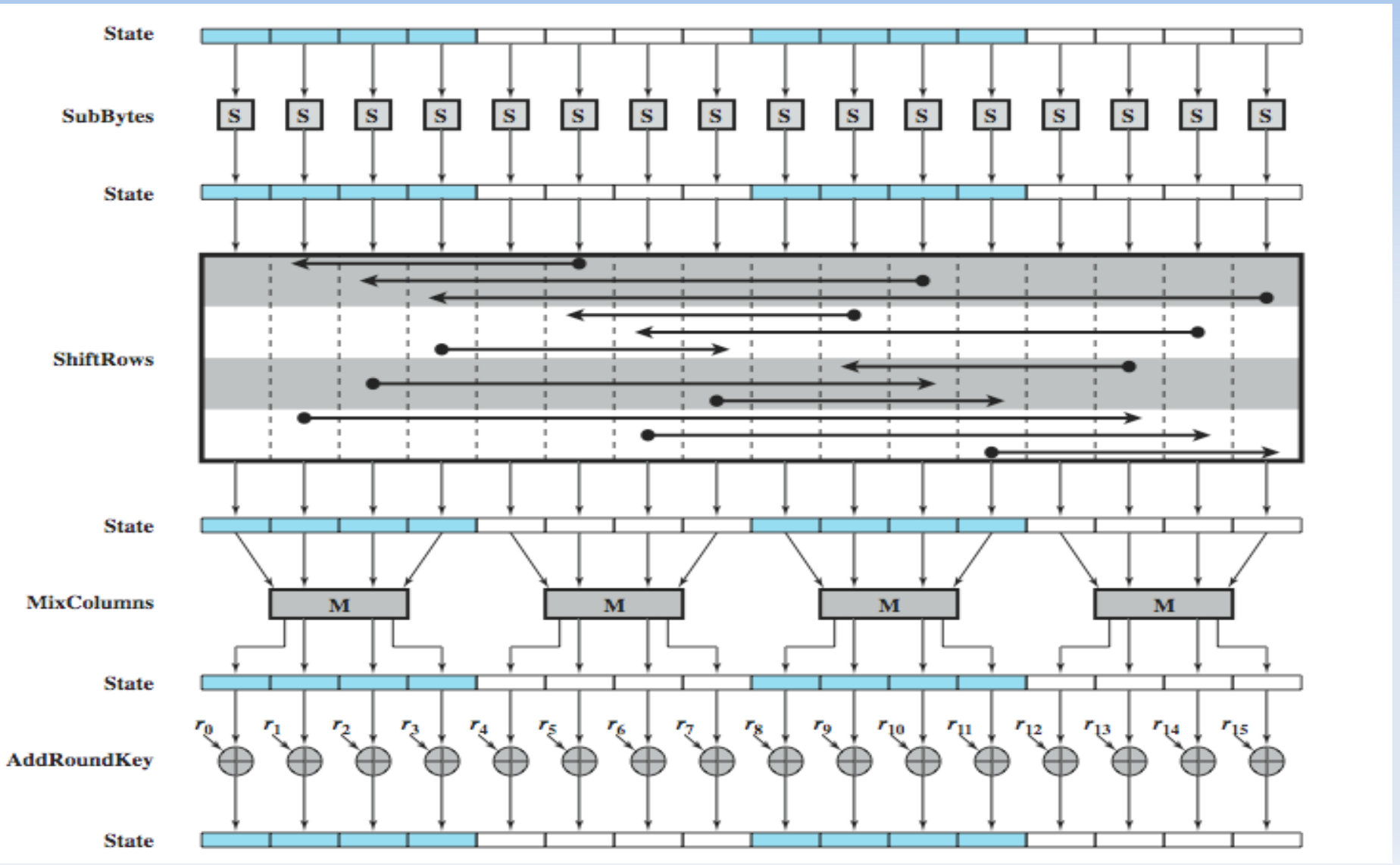
- La mayoría de los algoritmos usados comúnmente
- Procesa texto plano en bloques de tamaño fijo dando un bloque de texto cifrado de igual tamaño
- Algoritmos de cifrado por bloque muy usados son:
  - Data Encryption Standard (**DES**) (1977)
    - usa bloques de texto plano de 64 bits
    - y llave de 56 bits, rota por la EFF en 1998
    - en 1999, se aplica 3 veces DES con 3 llaves => **Triple DES**
  - Advanced Encryption Standard (**AES**) (2001)
    - concurso convocado por NIST y ganado por RIJNDAEL
    - usa bloques de 128 bits y llaves de 128/192/256 bits
- Otros algoritmos son: IDEA, Twofish, Blowfish, MARS, Serpent, ...

AES - cifrado

- los bloques de texto junto con la clave se someten a sucesivas rondas (>10) de transformación:
- sustituciones según una tabla (*S-box*)
- permutación fila a fila
- transposición de bytes en función de los otros
- *xor* del resultado con la clave de cada ronda, calculada a partir de la clave compartida



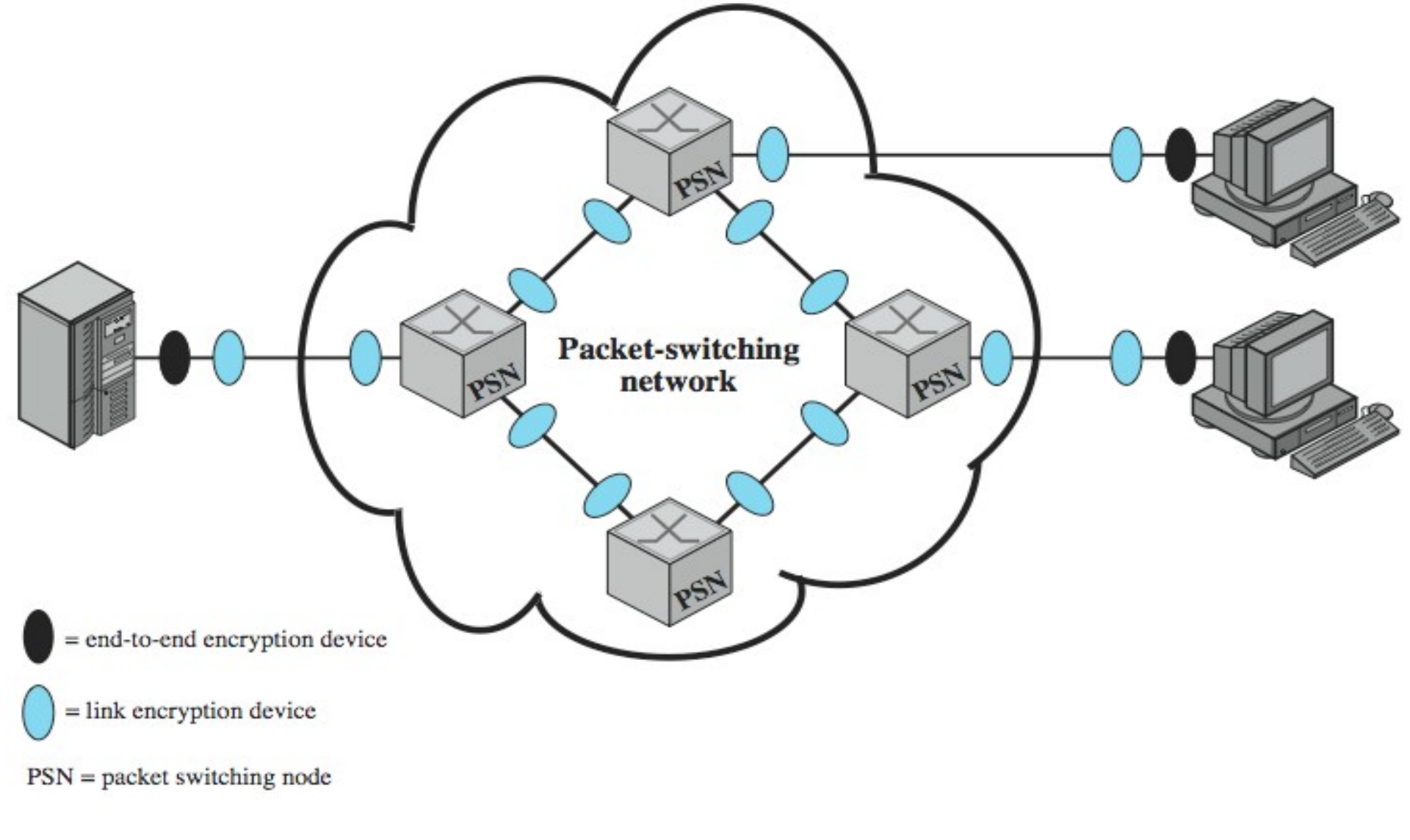
AES - ronda de encriptación



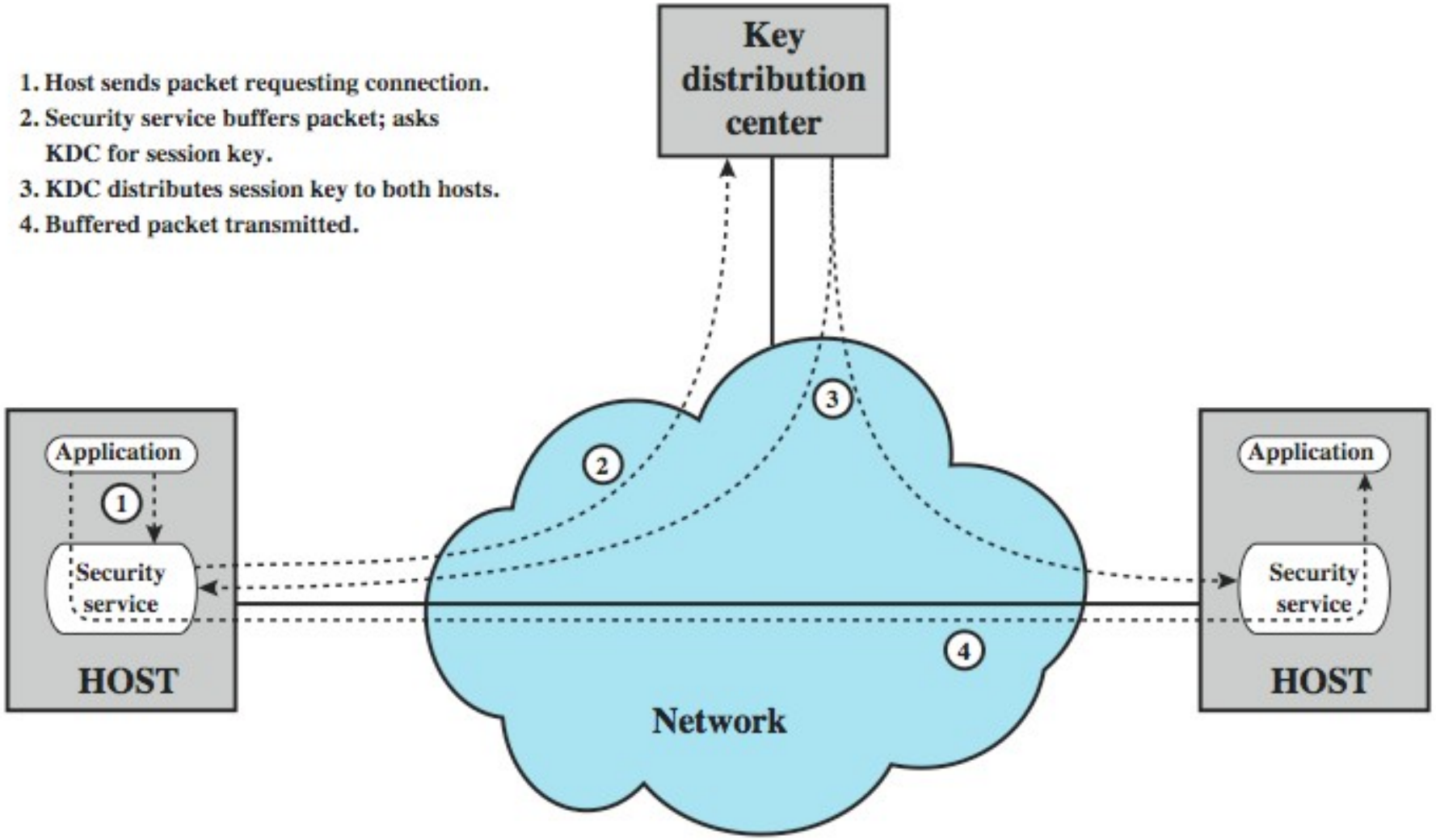
Lugar donde cifrar

- Encriptación en cada enlace vulnerable:
  - se asegura todo el tráfico en todas las capas
  - pero no dentro de cada conmutador, con los que además hay que compartir la clave
- Encriptación en los sistemas finales:
  - sólo se pueden asegurar la capa de aplicación y transporte
  - lo más seguro es emplear ambas
- Problemas de la distribución de claves compartidas:
  - la fortaleza del sistema depende de cómo se entregan las claves
  - además se necesita una clave por conexión y renovarlas
  - se puede usar un servidor de claves, pero continúan los problemas anteriores.

Lugar del cifrado - diagrama



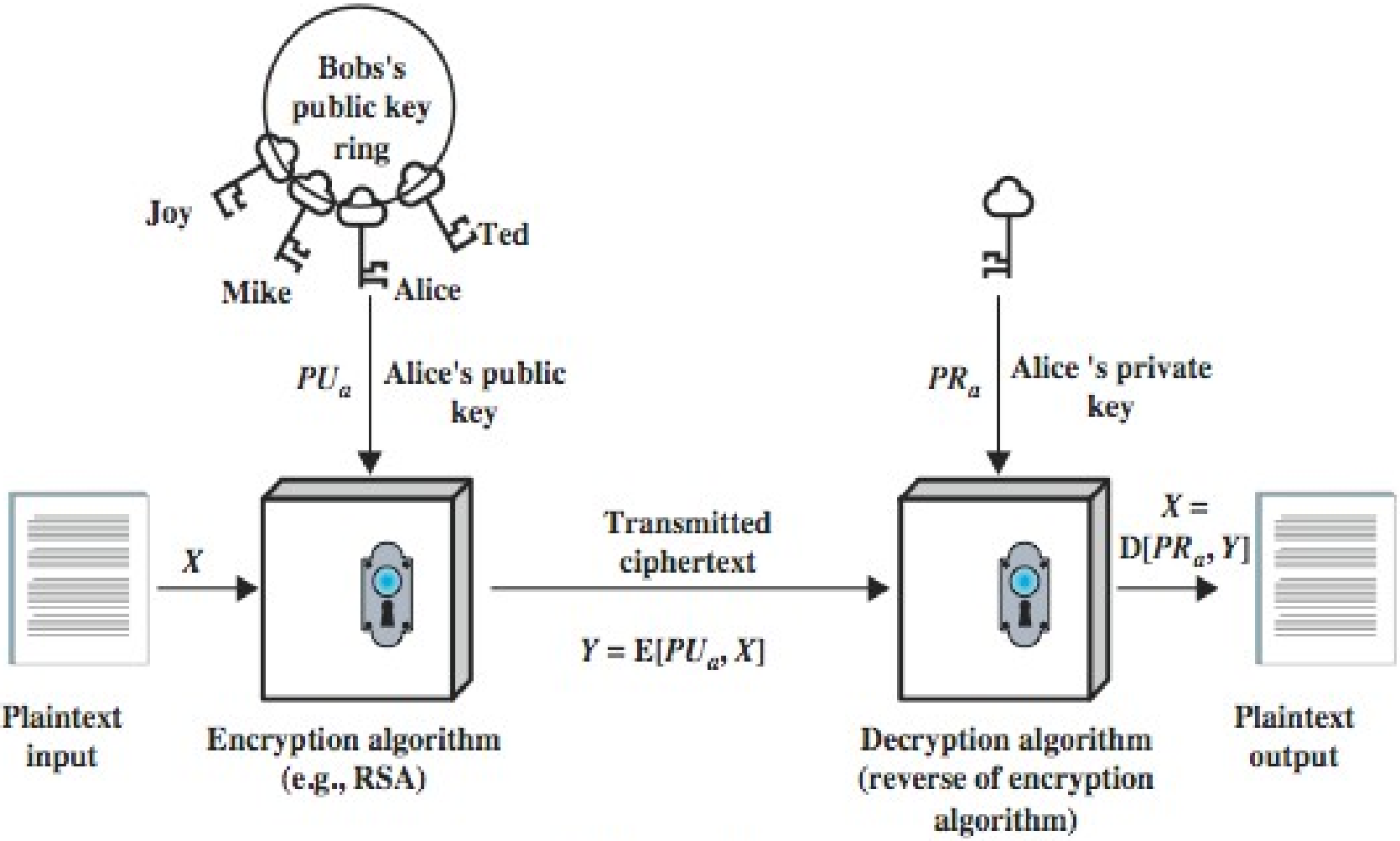
Distribución de las claves



Criptografía asimétrica

- Un usuario tiene **dos claves** distintas y no deducibles:
  - una **pública**, que usará cualquiera que quiera enviarle un mensaje cifrado (o autentificar un mensaje recibido)
  - una **privada**, que le permite descifrar sólo a él un mensaje recibido (o enviar uno firmado)
- Se basan en *funciones-trampa* de un solo sentido, basadas en problemas matemáticos:
  - factorización de enteros grandes: **RSA** (el + usado)
  - logaritmos discretos: **DSA**, **ElGamal**, **D-H**
  - curvas elípticas: **ECC**
- Estos algoritmos suelen ser mucho más lentos que los de clave compartida

Criptografía de clave pública



RSA - algoritmo

- a números más grandes, más difícil romperlo
- la long. del mensaje está limitado por n (=p*q)
- la longitud media usada hoy es de 1024 bits
- ya se ha roto una clave de 663 bits
- desafío para romper RSA2048b [edni] ($200K)
- Dr Scolnik dice que está en camino de lograrlo
- recomiendan usar ECC

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	KU = $\{e, n\}$
Private key	KR = $\{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

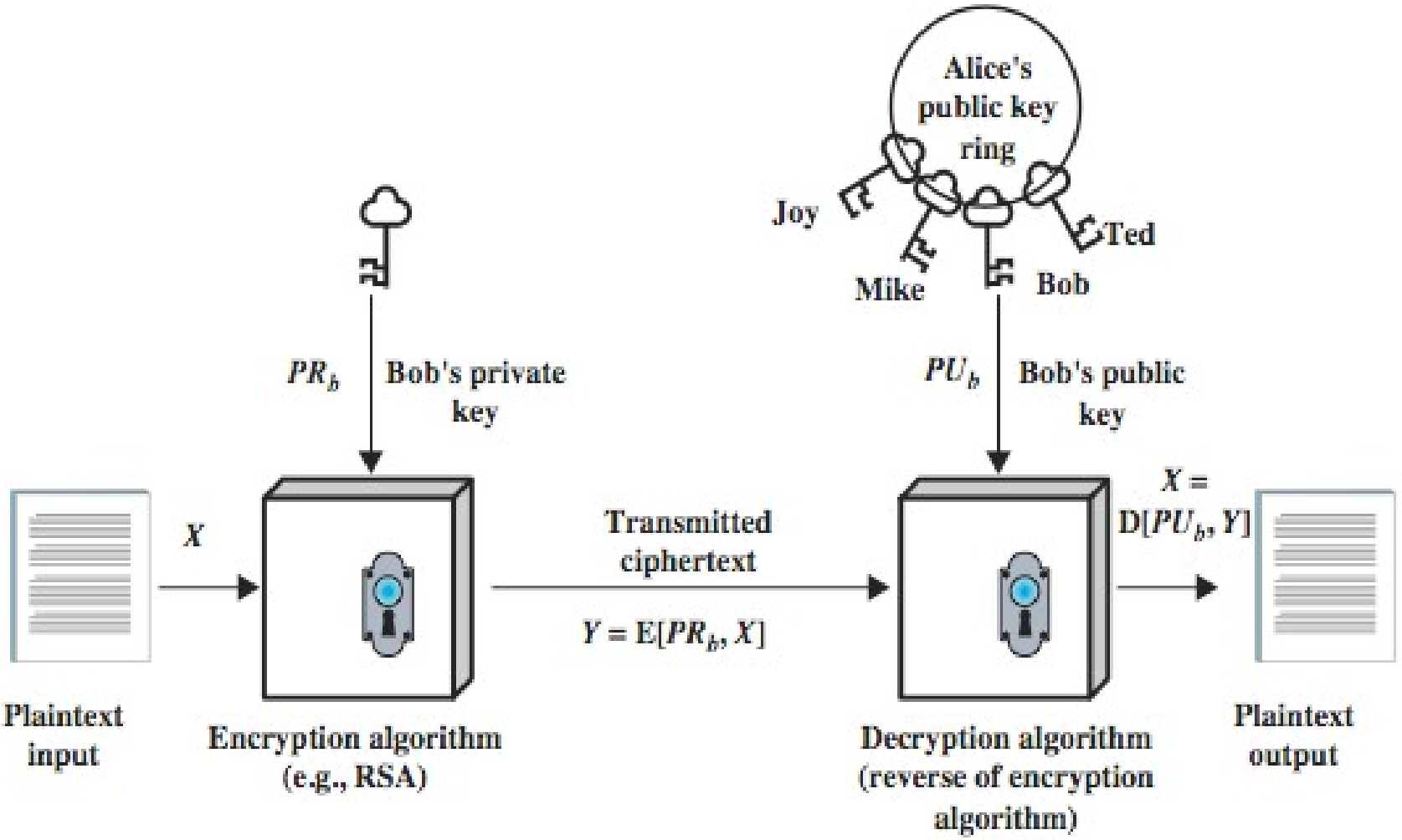
RSA - ejemplo

- p = 17, q = 11
- n = p * q = 187
- φ(n) = (p -1) * (q -1) = 160
- e = mcd(φ(n),n) = 7, 1<e<n; primo relativo de φ(n)
- de mod φ(n) = 1 => d = 23; 23 * 7 = 161 [mod 160 = 1]
- KU = {7,187}, clave pública
- KR = {23,187}, clave privada
- Para cifrar un texto plano de entrada M = 88
  - $C = 88^{7} \bmod 187 = 11$
- Para descifrarlo
  - $M = 11^{23} \bmod 187 = 88$

Autentificación de mensajes

- **No repudio** del mensaje: prueba de que un usuario ha enviado un archivo y que éste no ha sido modificado (integridad).
- Uso *inverso* del sistema de clave pública:
  - el emisor usa la clave **privada** para cifrar/firmar el archivo y el receptor la **pública** para descifrarlo/autentificarlo
- Sin embargo, este sistema es muy lento, por lo que se prefiere firmar un resumen del archivo (*Message Digest*), que se calcula con una función *hash* (troceo):
  - tb. permite probar que la integridad del archivo
  - se suelen usar las funciones:
    - **MD5**, calcula un número de 128 bits (roto)
    - **SHA-1**, calcula un nº de 160 bits (comprometido)
    - **SHA-2**, calcula un nº de 224/256/384/512 bits (el mejor)

Firmas digitales



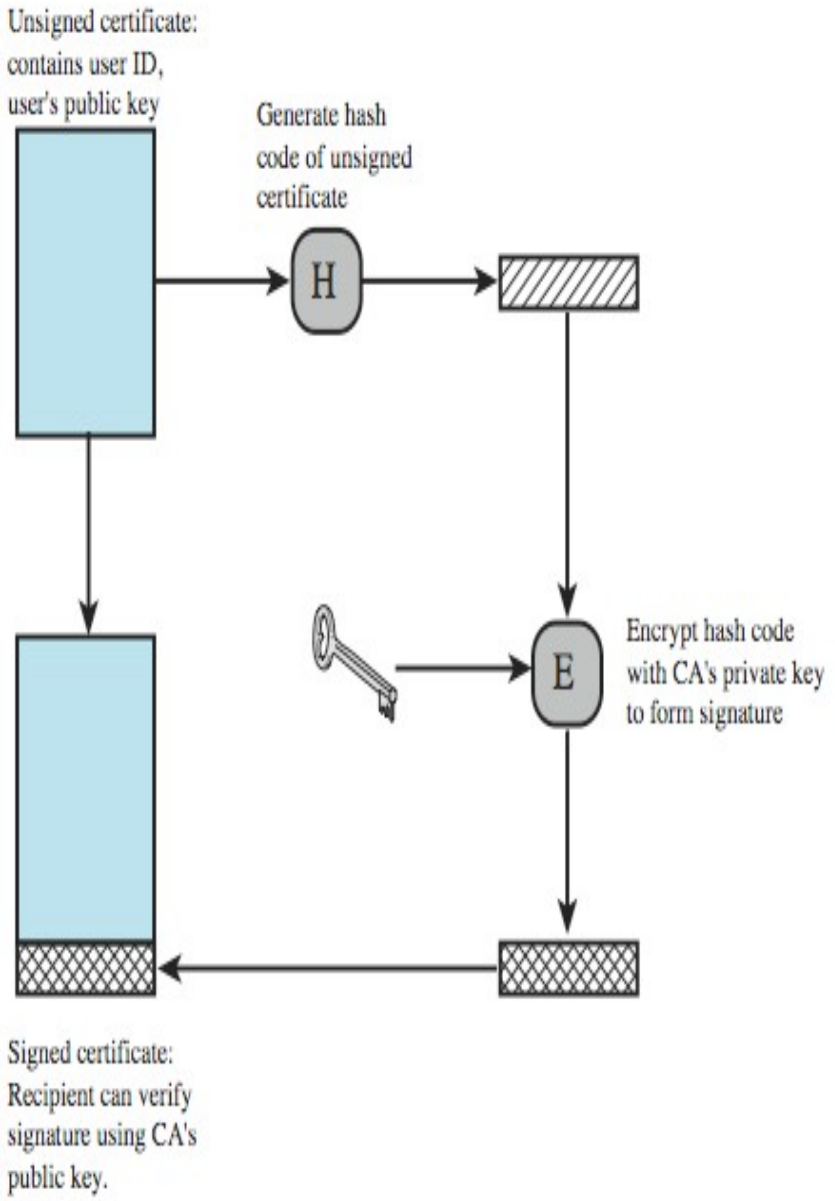
(b) Authentication

Autenticación de usuarios

- Prueba de que un usuario es quien pretende ser
- Se realiza mediante sistemas de
  - **clave pública,** asumiendo que todos los posibles usuarios conocen la clave pública del servidor:
    - el cliente crea un mensaje con su nombre y clave pública y la marca de tiempo, lo cifra y lo envía
    - el servidor descifra el mensaje, comprueba los datos y le envía un mensaje cifrado con el nombre, la misma marca de tiempo que recibió y la marca de tiempo actual
    - el cliente descifra el mensaje y comprueba las marcas y envía un mensaje cifrado con nombre y marca del servidor
  - **clave privada,** requiere que una 3ª parte confiable actúe como servidor de distribución de claves:
    - **Kerberos**

Certificados digitales

- Se basa en la existencia de Autoridades de Certificación (**CA**), que garantizan la vinculación entre la identidad de un sujeto o entidad y su clave pública mediante una ***P**ublic **K**ey **I**nfrastructure* (PKI)
- Los usarios pueden registrarse en los CA, que les expide un **certificado**, que contiene la clave pública, nombre, nº serie, periodo de validez y método empleado..
- Un CA en España: **FNMT**



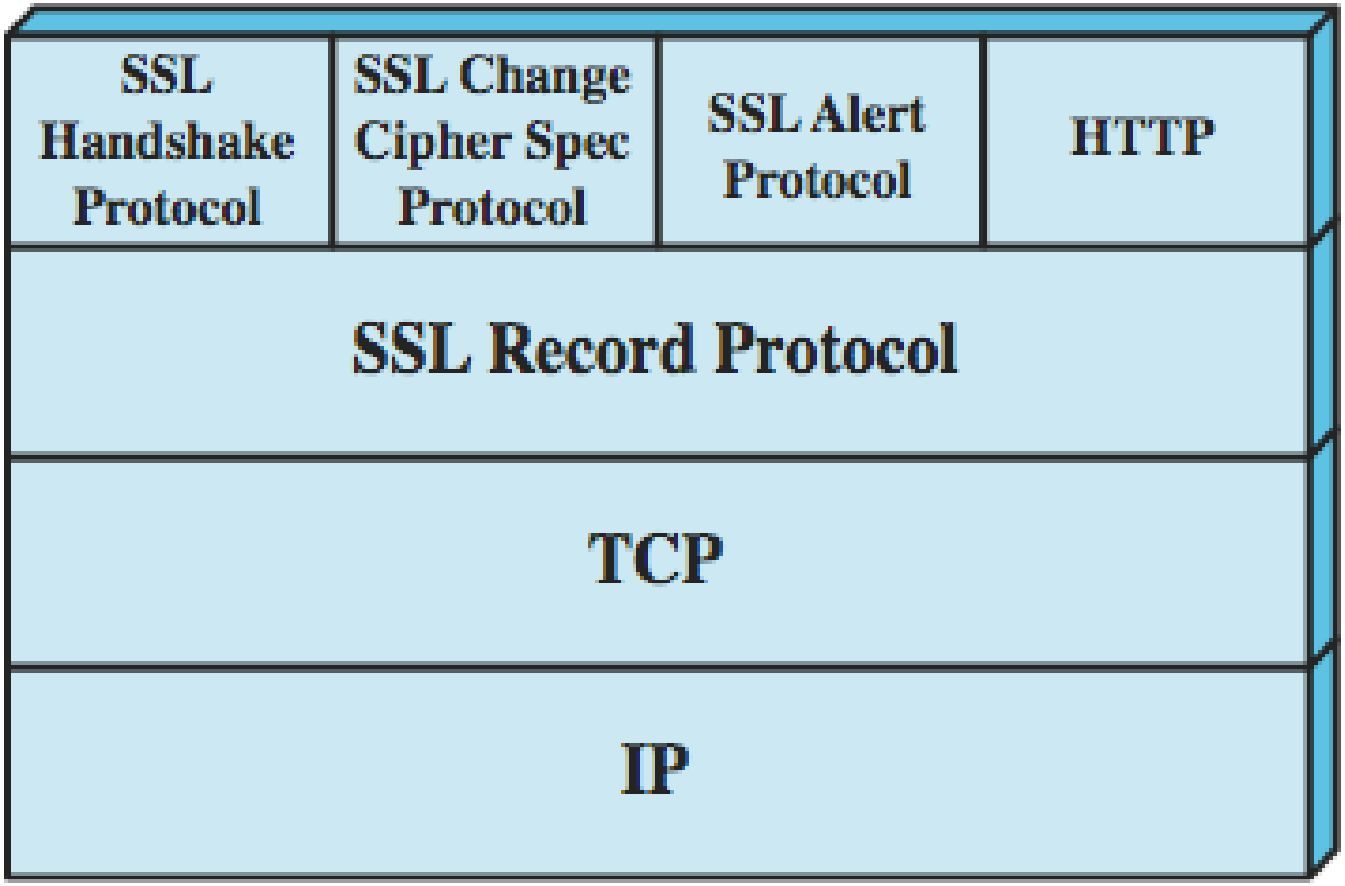
Privacidad del correo

- Seguridad en(/sobre) la capa de aplicación
- El esquema más usado es **PGP** (*Pretty Good Privacy*), que proporciona privacidad, autenticación, integridad y no repudio
- El proceso combina MD5, RSA, el algoritmo de compresión LZ, IDEA y Base64
- **OpenPGP** es el estándar IETF y a partir de éste el **GPG** (GnuPG)
- Las aplicaciones se extienden más allá del correo electrónico a todo tipos de mensajes y archivos
- **Herramientas de criptografía** usuales :
  - ***openssl, gpg, ssh-keygen*** (para consola)
  - *seahorse* (Appl. > Acces. > Contraseñas y claves ...)

Protocolo SSL/TLS

- Seguridad en(/sobre) la capa de transporte
- Protocolo de Netscape **SSL** (*Secure Socket Layer*) se convirtió en el estándar IETF **TLS** (*Transport Layer Security*)
- Se realiza la autenticación del servidor por el cliente (y viceversa, si es necesario) utilizando una autoridad de certificación y el establecimiento de un algoritmo de cifrado simétrico y una clave por sesión
- Primero se utilizó para asegurar las transacciones web, pero hoy en día usa en muchos otros tipos de aplicaciones (puede usarlo cualquiera que se base en TCP)

SSL - arquitectura



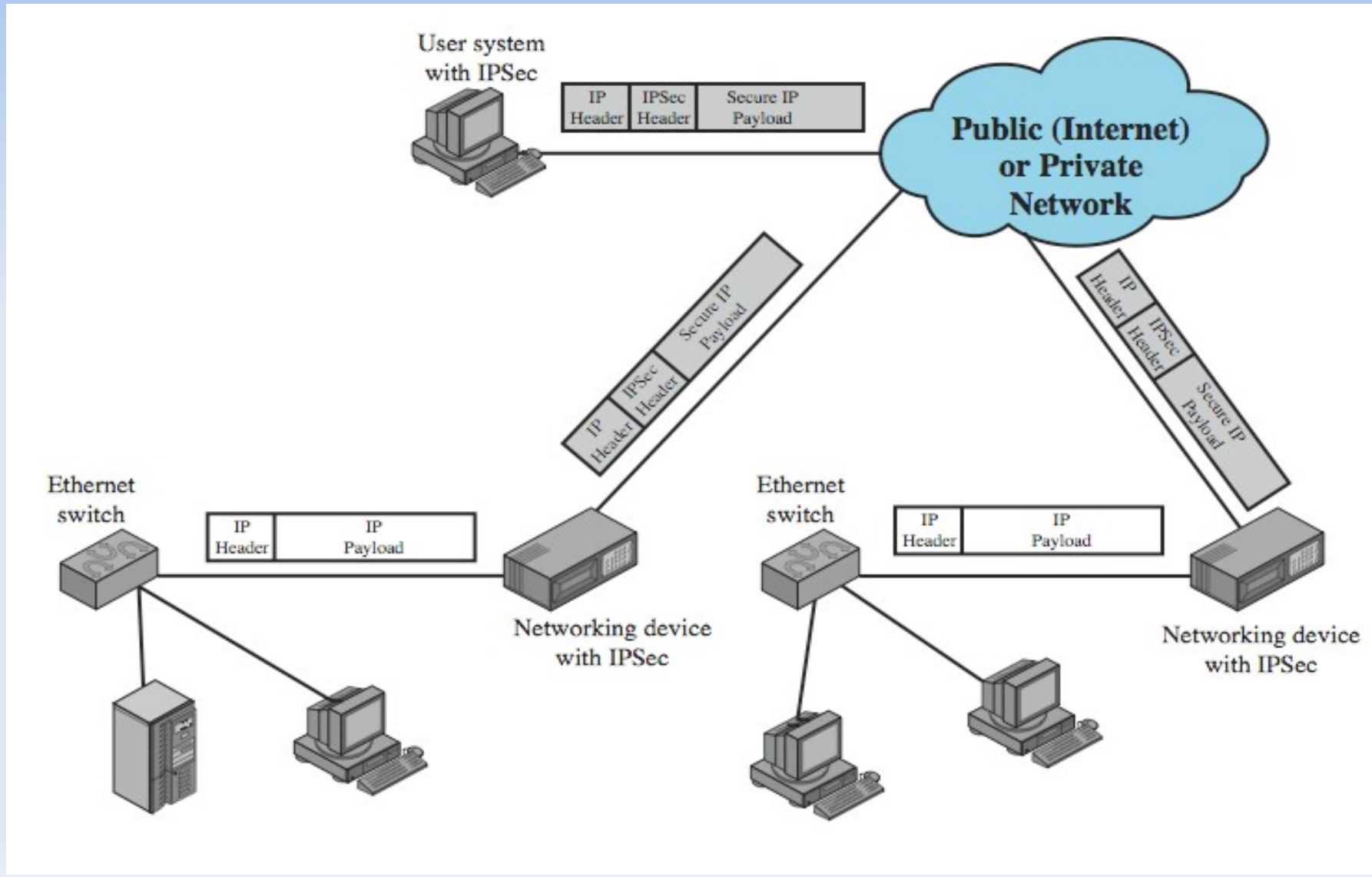
IPv4/6 seguros

- Seguridad en(/sobre) la capa de red
- El protocolo de la IETF **IPSec** lleva asociados dos:
  - el de cabecera de autenticación **AH**, que ofrece autenticación del origen e integridad de datos
  - el de encapsulado de seguridad de capa **ESP**, que ofrece además cifrado
- Antes del intercambio de datos, se establece una asociación de seguridad (SA) en cada sentido, definido por el protocolo (AH/ESP), la IP y un identificador de conexión (SPI), y que se puede implementar de dos formas:
  - modo transporte, cabecera IPSec entre las cab. IP y TCP
  - modo túnel, datagrama original encapsulado en uno nuevo (p.e. cuando hay que atravesar un cortafuegos)

Redes Privadas Virtuales (VPN)

- Conjunto de *hosts* interconectados usando una red insegura
  - p.e. conectando las LANs de una organización a través de Internet
- Se usa encriptación y protocolos especiales para proporcionar seguridad:
  - para evitar robo de información y el acceso a usuarios no autorizados
- Las soluciones propietarias son problemáticas, por esto se desarrolló el estándar IPSec (RFC 1636):
  - es una extensión al protocolo IP que añade cifrado fuerte para permitir servicios de autenticación y, de esta manera, asegurar las comunicaciones a través de dicho protocolo. Inicialmente fue desarrollado para usarse con el nuevo estándar IPv6, aunque posteriormente se adaptó a IPv4

Esquema IPSec



Wifi – acceso protegido

- Seguridad en la capa de enlace
- En la IEEE 802.11 se usa el protocolo de privacidad equivalente a cable **WEP**
- WEP ofrece autenticación y cifrado de datos sobre el enlace entre el AP y el resto de nodos empleando un algoritmo de clave compartida y simétrica.
- Utiliza un cifrado de flujo llamado RC4, que genera un flujo de claves de 64 bits, a partir de la clave secreta de 40b y un vector de inicialización (IV) de 24b.
  - cada clave se usa para cifrar un bloque de 64b de la carga de datos de una trama
- Este esquema ya ha sido roto hace tiempo

802.11X/i – control de acceso

- IEEE **802.11i** corrige el cifrado WEP con 2 protocolos:
  - **TKIP** (*Temporal Key Integrity P.*) compatible con WEP amplia el tamaño de clave y las cambia dinámicamente
  - **CB-MAC** basado en AES (no comp. con hardware WEP)
- IEEE **802.1X** permite **autenticar** a usuarios (no *hosts*)
  - se basa en EAP (RFC3748), un encapsulado usable sobre PPP, 802.3, 802.11, ... y puede usar diversos métodos de autenticación TLS (usa certificados), TunnelTLS o PEAP (usan LDAP, Kerberos...), vía servidores de acceso (RADIUS)
- Mientras que se terminaba la industria implementó **WPA** (*Wi-Fi Protected Access*), incluyendo TKIP y admite autenticación ***P**re**S**hared **K**ey* (**PSK** o personal) y 802.11X (EAP o empresarial)
- **WPA2** está basado en la versión final 802.11i (2004)

Filtrado de paquetes

- Software que permite filtrar el flujo de paquetes por la pila de protocolos y que puede servir de:
  - cortafuegos
    - local, con acceso a los PIDs (API de aplicación)
    - router
  - nat
  - manipulación
- Linux: **Netfilter** (núcleos 2.6.X)
- Windows: **WFirewall** (XP2/2003) y **WFP** (Vista/2008)
  - interesante lectura p.1 y p. 2
- MacOSX: **Ipfirewall** (10.4) + **ALF** (10.5) [comentario]
- Filtrado de contenidos (aplicación): IDS, antivirus, ...

Netfilter - iptables

- **iptables** es una herramienta para manejar *Netfilter*:
  - iptables [opc] [-t tab] [com] [cad] [reg] -j DESTINO
- Hay 4 tablas (**filter**, **nat**, **mangle y raw**) y cada tabla contiene varias cadenas predefinidas (+ de usuario)
  - filter: INPUT, FORWARD y OUTPUT
  - nat: PREROUTING, OUTPUT y POSTROUTING
- Destinos: ACCEPT, DROP, REJECT, LOG, RETURN
- Las reglas se componen de parámetros: -p -s -d -i -o
- Extensiones de coincidencia (-p/-m)
- Ver *man iptables*
- Otras herramientas para manejar *Netfilter*: *ebtables*, *arptables*, *ip6tables*