

Criptografía y Seguridad de Datos

Introducción a la criptografía: confidencialidad de los mensajes

Carlos Figueira.

Universidad Simón Bolívar

Basado en láminas del Profesor

Henric Johnson (<http://www.its.bth.se/staff/hjo/>)

henric.johnson@bth.se

Contenido

- Principios de cifrado convencional
- Algoritmos de cifrado convencional
- Modos de operación de cifrado por bloque
- Ubicación de dispositivos de cifrado
- Distribución de claves

Principios de cifrado convencional

- Un esquema de cifrado tiene 5 ingredientes:
 - Texto “en claro”
 - Algoritmo de cifrado
 - Clave secreta
 - Texto cifrado
 - Algoritmo de descifrado
- Seguridad depende de secreto de la clave, no del algoritmo

Principios de cifrado convencional

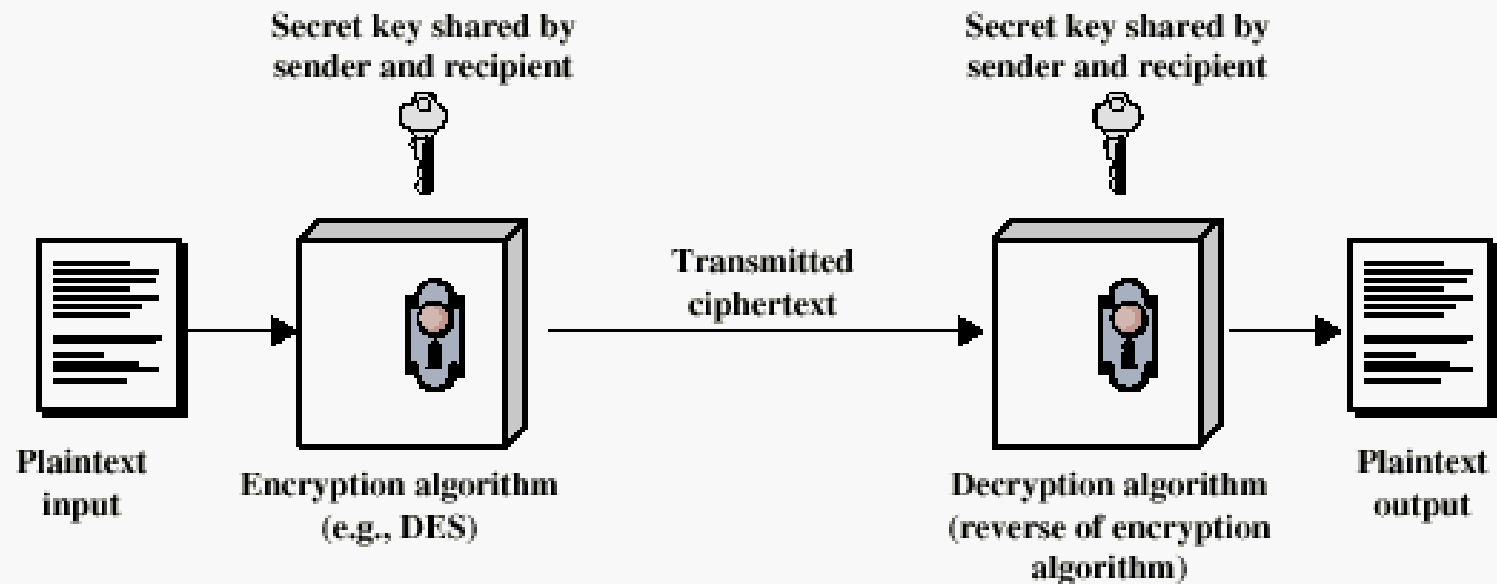


Figure 2.1 Simplified Model of Conventional Encryption

Algoritmos

- Clasificados de acuerdo a tres dimensiones independientes:
 - Tipo de operaciones usadas para transformar el texto claro a cifrado
 - El número de claves usadas:
 - simétrica (clave única)
 - asimétrica (dos claves, o cifrado de clave pública)
 - La forma en que se procesa el texto claro (por bloque o de flujo)

Algoritmos (cont.)

- Elementos del texto claro: bit, letra, grupo de bits o letras
- Operaciones para cifrar:
 - Sustitución
 - Transposición (reordenamiento)
- Múltiples etapas

Criptoanálisis

- Proceso para intentar descubrir un texto claro o una clave de cifrado
- Ejemplos de estrategias de ataque:
 - Sólo texto cifrado (y algoritmo)
 - Texto claro conocido (texto cifrado-claro, algo.)
- Cifrado es computacionalmente seguro si costo de romperlo $>$ valor de info. o tiempo $>$ vida útil de info.

Tiempo promedio requerido para búsqueda exhaustiva de clave

Key Size (bits)	Number of Alternative Keys	Time required at 10^6 Decryption/ μ s
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Estructura de cifrado de Feistel

- Casi todos los algoritmos de cifrado convencional, incluyendo DES, tienen estructura descrita por Horst Feistel (IBM, 1973)
- La realización de una Red de Feistel depende de la selección de 6 parámetros y características de diseño

Estructura cifrado Feistel

- **Tamaño del bloque:** a mayor tamaño, mayor seguridad
- **Tamaño de la clave:** mayor => mayor seguridad
- **Número de etapas o rondas:** múltiples vueltas dan mayor seguridad
- **Algoritmo de generación de sub-claves:** mayor complejidad dificulta criptoanálisis.
- **Velocidad de cifrado/decifrado en software**

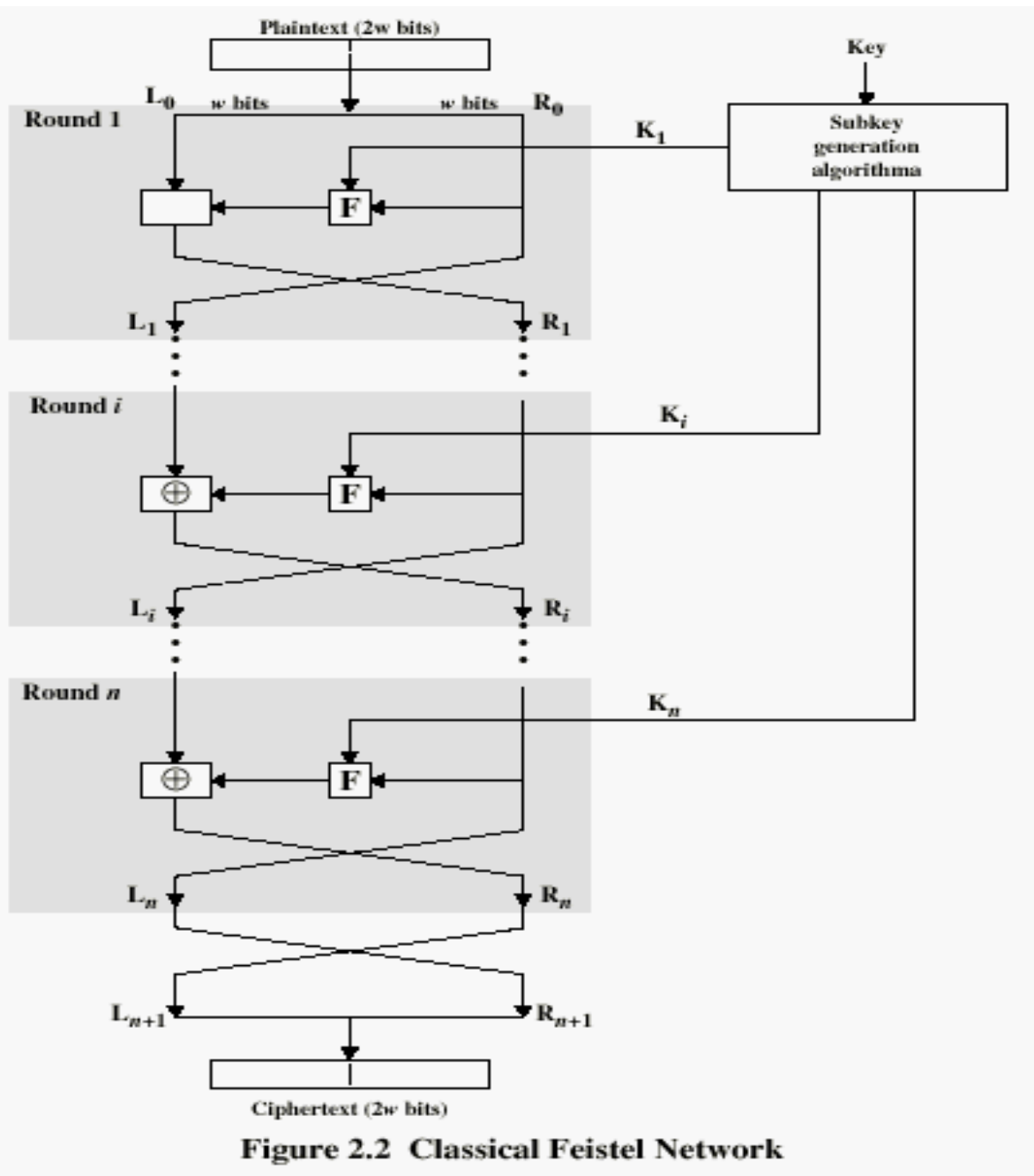


Figure 2.2 Classical Feistel Network

Algoritmos de cifrado convencional

- Data Encryption Standard (DES)
 - Esquema más usado y difundido
 - Se conoce como el Data Encryption Algorithm (DEA)
 - DES es un cifrador de bloque
 - El texto claro se procesa en bloques de 64-bit
 - La clave es de 56-bits

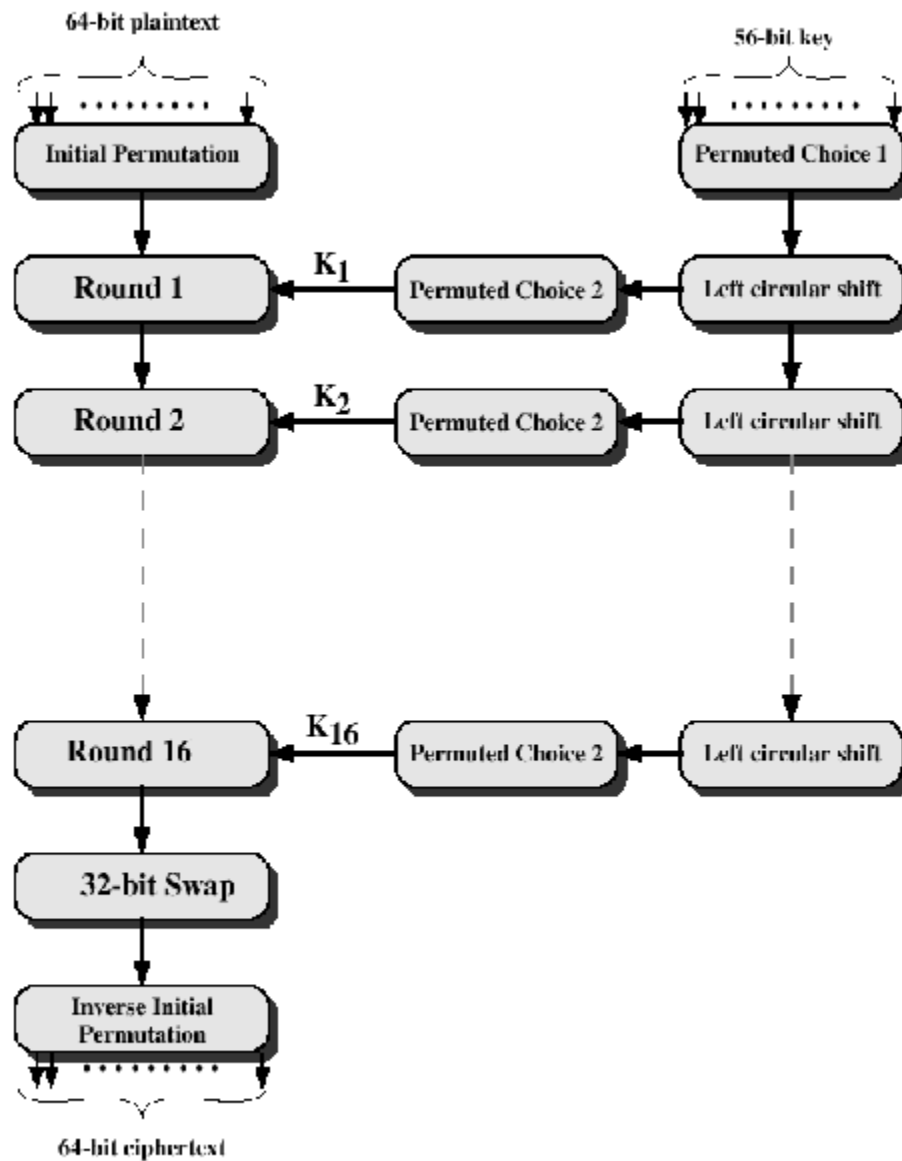


Figure 2.3 General Depiction of DES Encryption Algorithm

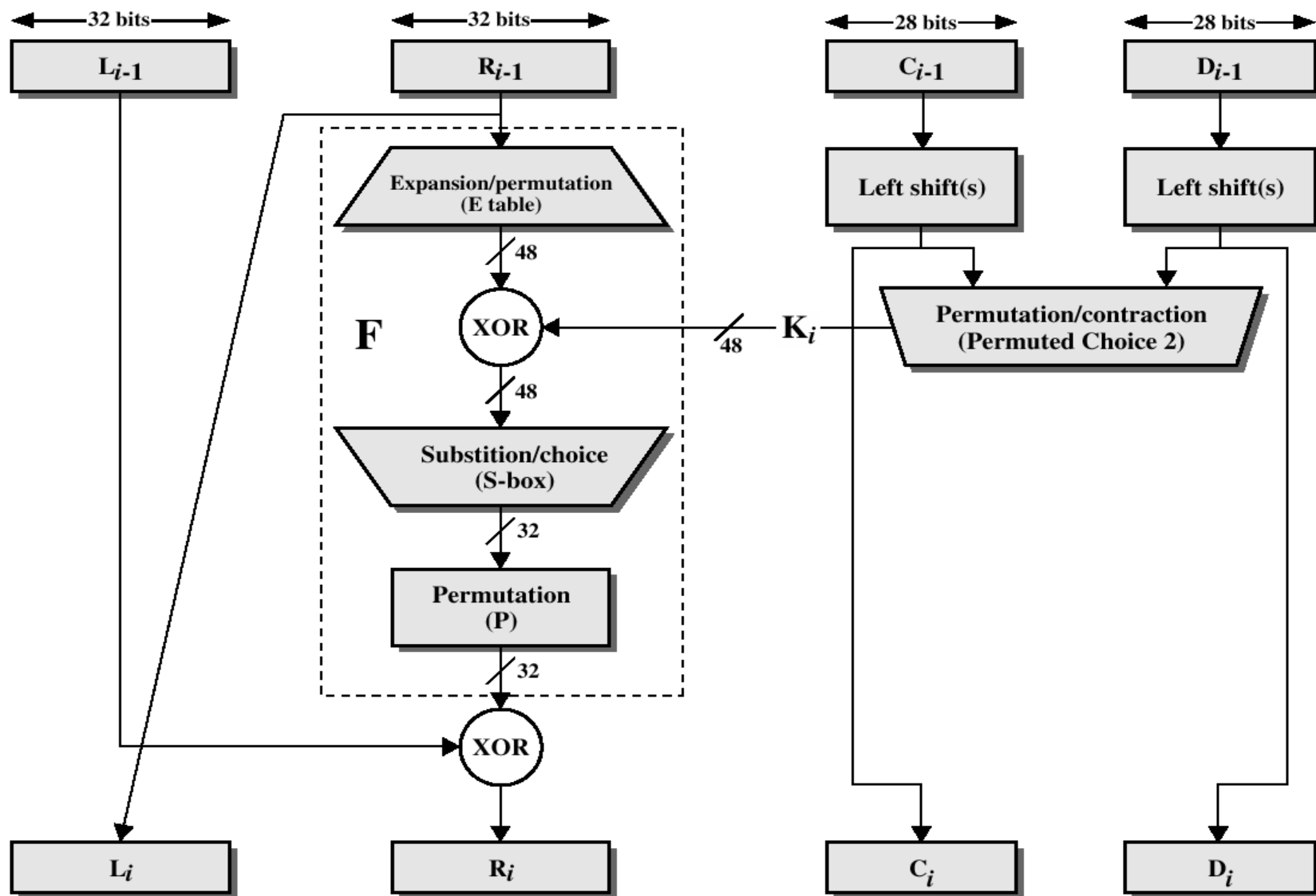


Figure 2.4 Single Round of DES Algorithm

DES

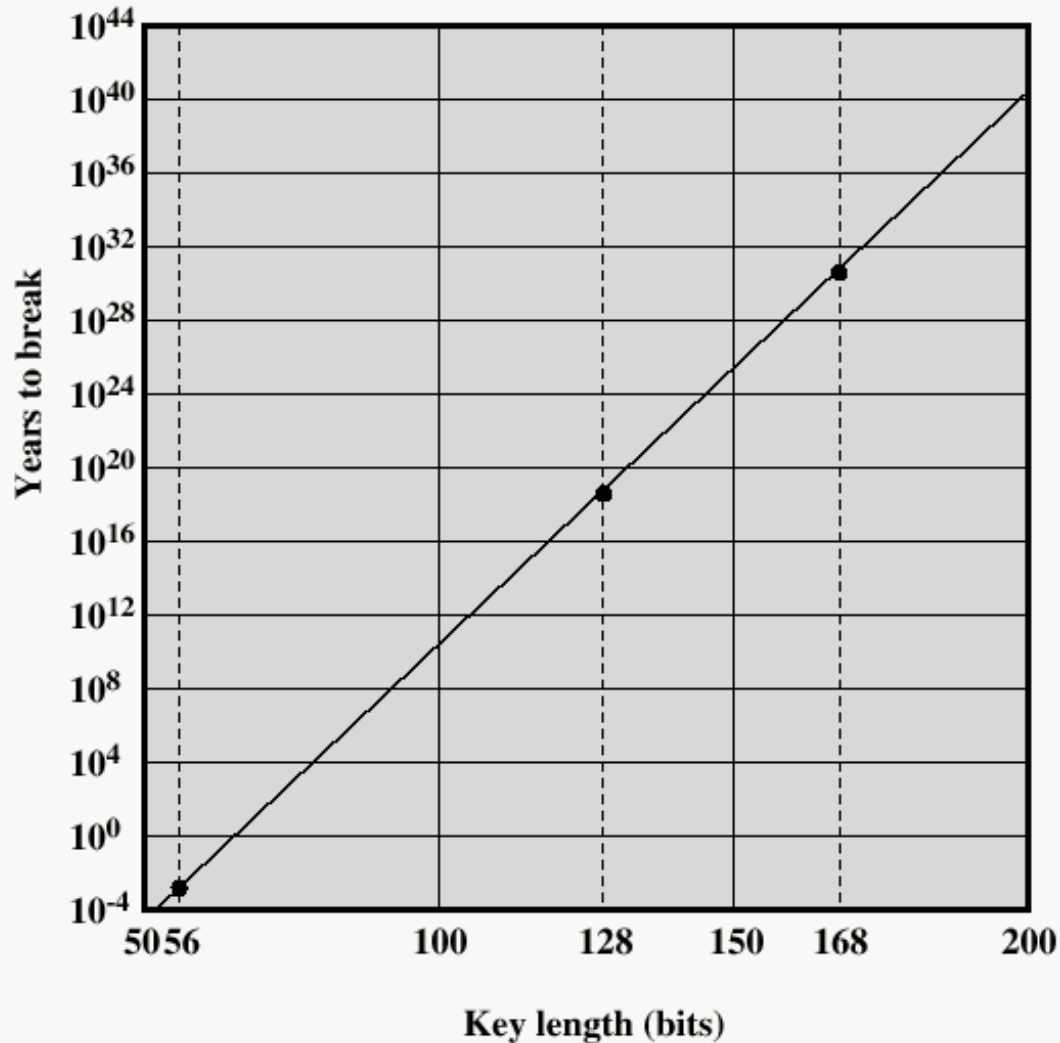
- **Procesamiento en cada iteración:**

- $L_i = R_{i-1}$
- $R_i = L_{i-1} \otimes F(R_{i-1}, K_i)$

- **Inquietudes acerca de:**

- El algoritmo y el tamaño de la clave (56-bits)

Tiempo romper cifrado (10^6 decifrados/ μ s)



Extendiendo DES

- Excelentes propiedades:
 - Muy eficiente (implementaciones en Hw), robusto, probado, etc.
 - Pero ... clave muy pequeña
- Propuestas: cifrar varias veces!
 - 2DES: No mejora mucho
 - 3DES

Doble DES: $E_{K_2}(E_{K_1}(P))$

- Costo doble, clave efectiva de 57 bits (no 112)
- Ataque de encuentro en el medio.
 - Se tiene uno o mas pares (texto plano, texto cifrado). Ya que $X = E_{K_1}(P) = D_{K_2}(C)$, ataca cifrando P con todas las claves y guarda, luego descifra C con todas las claves, ordenas y busca dos iguales. Ese par de claves son candidatas; aplica en otros pares para ver si funciona

Triple DEA (3DES)

$$C = E_{K3}[D_{K2}[E_{K1}[P]]]$$

- 3 claves, 3 ejecuc. de DES (cifra-descifra-cifra)
- Longitud de clave efectiva de 168 bits
- La D en el medio permite compatibilidad con DES (misma clave), e implementaciones en Hw

Triple DEA

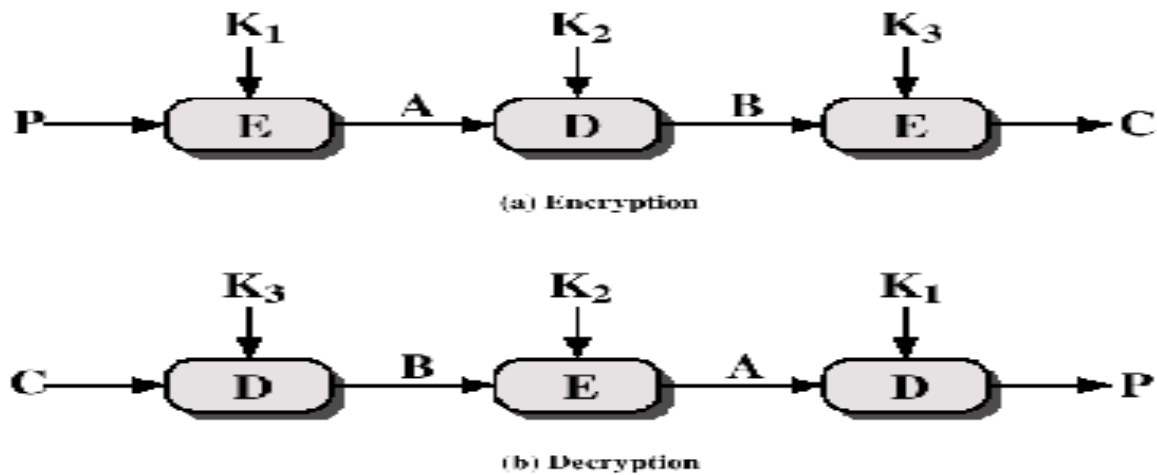


Figure 2.6 Triple DEA

Triple DEA

- La D de EDE evita ataques
- Si se usa una misma clave es equivalente (y compatible con DES)
- Variantes: con dos claves ($K1=K3$)
- Está siendo reemplazado por AES, hasta 6 veces más rápido en software.
- Sólo en transacciones electrónicas y TC
- Ataques conocidos muy costosos (memoria-tiempo)

Otros cifradores de bloque simétricos

- **AES: Advanced Encryption Standard**
 - 3DES es lento. NIST realizó concurso en 1997 para sustituto de DES
 - Ganador: Rijndael (de Daemen y Rijmen)
 - Eficiente (en hard. y soft.), flexible, claves 128, 192, 256 bits

AES

- Bloque de 128 bits. No es Feistel. Se procesa todo el bloque en cada etapa
- Clave -> vector de 44 palabras de 32 bits. En c/ ronda usa 4 palabras, 128 bytes de clave
- 4 fases: sust. de bytes, desplaza filas, mezcla columnas, suma de clave etapa
- 1 fase de suma de clave de etapa, luego 9 etapas de 4 fases, y 1 final de 3 fases

Otros cifradores de bloque simét. (cont.)

- International Data Encryption Algorithm (IDEA) (suizos 91)
 - Clave 128-bit (uno de los primeros)
 - Usado en PGP, muy probado
 - Función de etapa: XOR, suma binaria 16 bits, mult. binaria de enteros
 - Seis claves generadas por *rotates* para las 8 etapas

Otros cifradores de bloque simét. (cont.)

- Blowfish (Schneier 93)
 - Fácil de implementar
 - Rápido
 - Corre en menos de 5K de memoria
 - Función de etapa: cajas S (generadas), XOR y sumas binarias
 - Costosa generación de subclaves, no es adecuado para aplicaciones que necesiten cambio frecuente de clave

Otros cifradores de bloque simét. (cont.)

• **RC5 (Rivest 94), RFC 2040**

- Adecuado para hardw. y soft.
- Rápido, simple
- Adaptable a procesadores de diferente tamaño de palabra
- Número variable de rondas
- Longitud de clave variable
- Bajo consumo de memoria
- Alta seguridad
- Rotaciones dependientes de los datos

Modos de operación de cifrado en bloques

- Conversión a flujo: usando registros de desplazamiento
- Método original ECB (*Electronic Code Book*): bloques iguales de entrada (claro) producen bloques iguales de salida (cifrado)
- Esto puede dar al criptoanalista mucho material para analizar (p.e. búsqueda de patrones de 64 bits conociendo parte de la entrada)
- Solución Enlazado de bloques cifrados (*Cipher Block Chaining Mode CBC*)

CBC

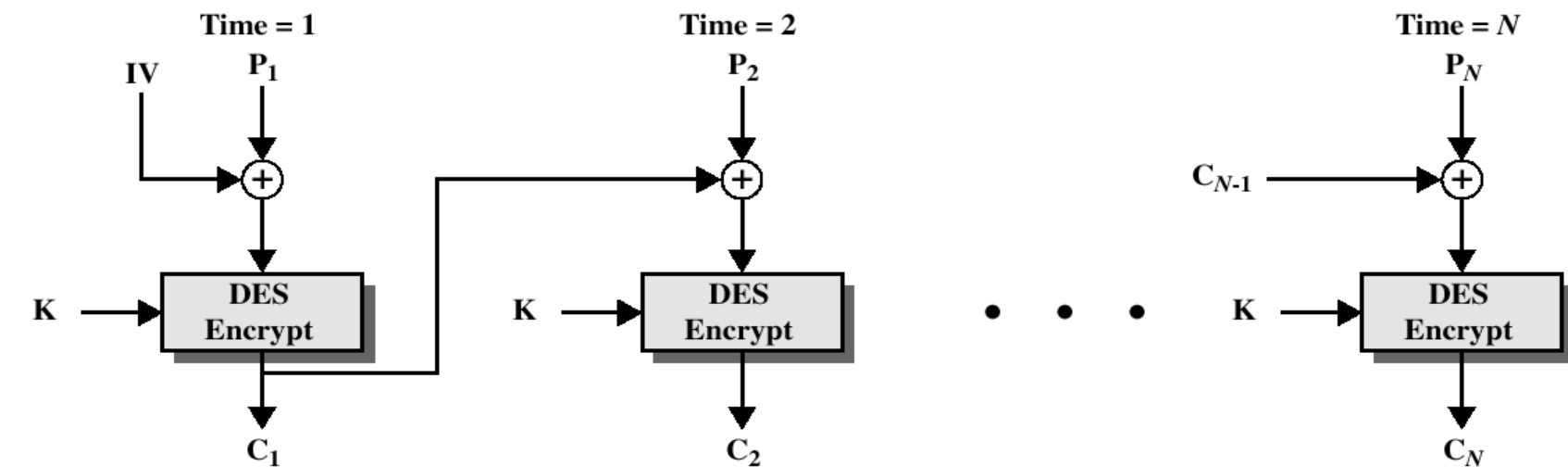
- La entrada del algoritmo de cifrado es el XOR del bloque claro actual y el cifrado anterior
- Para el primer bloque usa un Vector de Inicialización

$$C_i = E_k[C_{i-1} \oplus P_i]$$

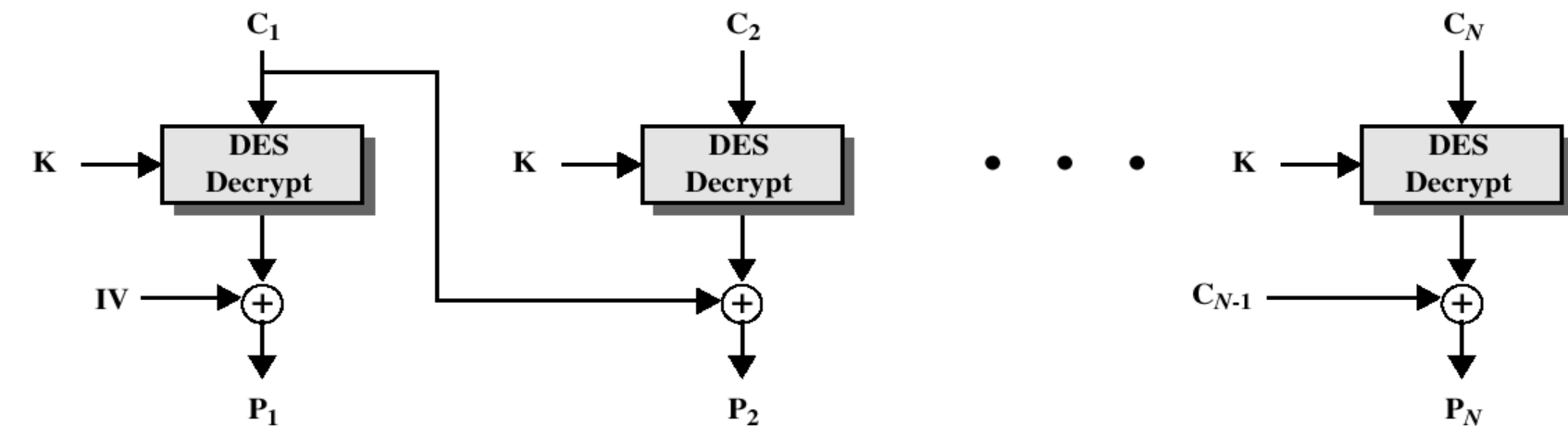
$$D_K[C_i] = D_K[E_K(C_{i-1} \oplus P_i)]$$

$$D_K[C_i] = (C_{i-1} \oplus P_i)$$

$$C_{i-1} \oplus D_K[C_i] = C_{i-1} \oplus C_{i-1} \oplus P_i = P_i$$



(a) Encryption



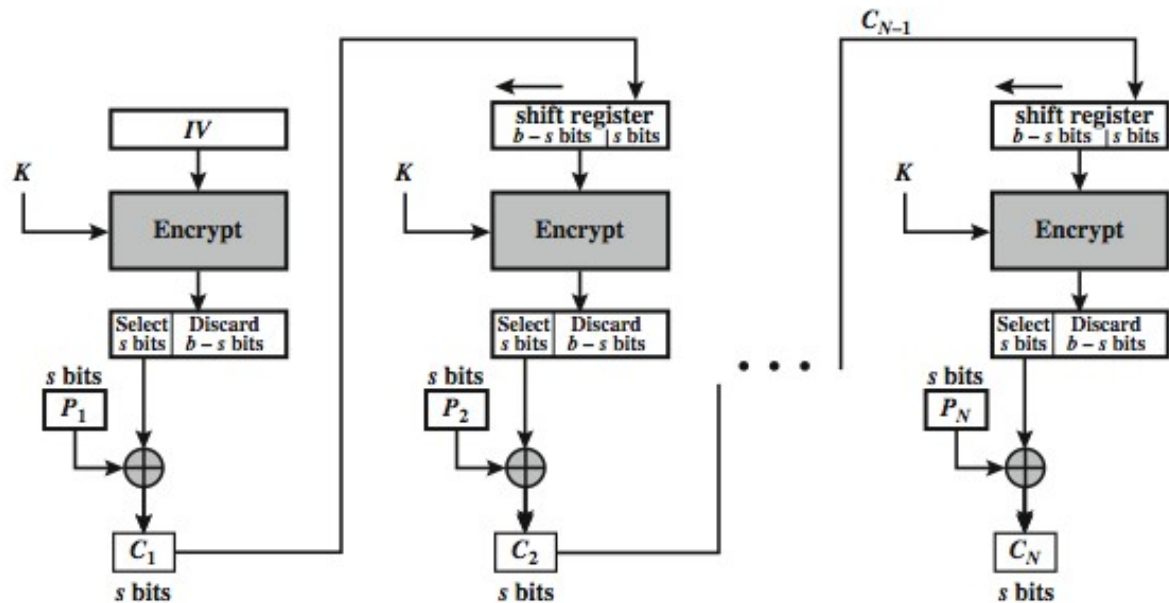
(b) Decryption

Figure 2.7 Cipher Block Chaining (CBC) Mode

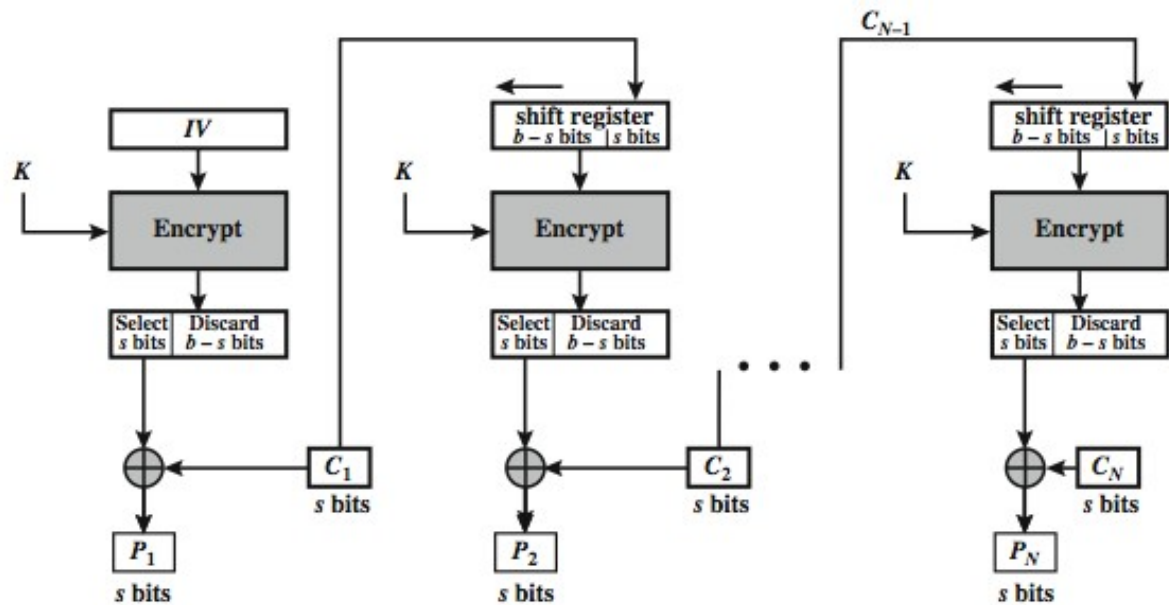
CFB: Cifrado con realimentación

- Mensaje considerado flujo de bits
- Se agrega a la salida del cifrado de bloque; el resultado se usa en etapa siguiente
- estándar permite realimentación de 1, 8, 64 or 128 etc bits (CFB-1, CFB-8, CFB-64, CFB-128 etc)

CFB-s (s-bits)



(a) Encryption



(b) Decryption

¿Dónde cifrar/descifrar?

- Dos entes emisor/receptor interconectados a través de uno o más proveedores de acceso
- ¿Cifra el proveedor o cifra el usuario?

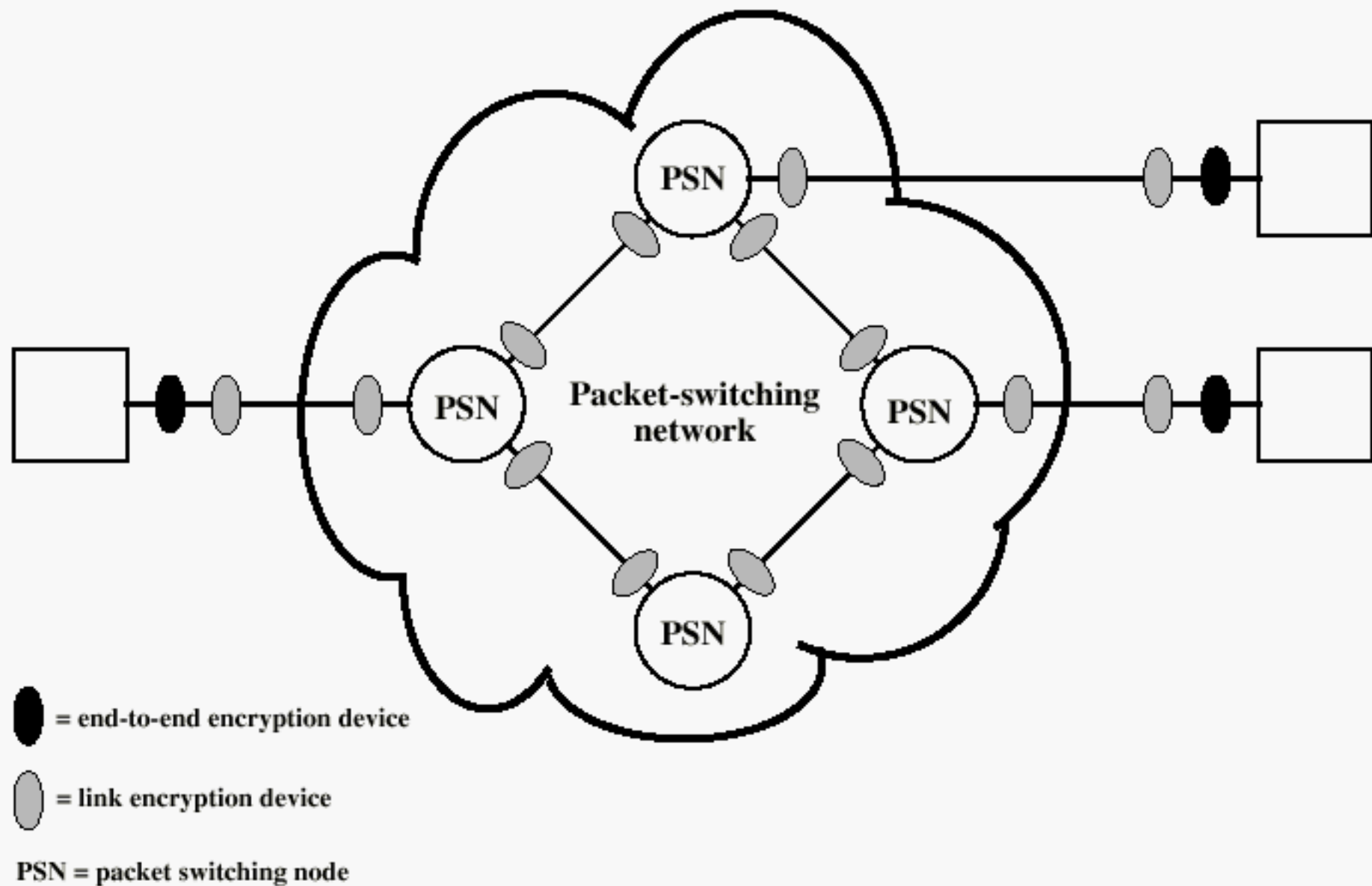


Figure 2.9 Encryption Across a Packet-Switching Network

¿Dónde cifrar/descifrar?

- **Enlaces:**

- ¡Cifrado/descifrado en cada enlace!
- Seguridad alta (incluso cabeceras)
- ¿Quién controla?

- **Punto a punto:**

- Emisor cifra, receptor descifra
- Cuerpo mensaje cifrado, cabeceras en claro

- ***Lo mejor: ¡Ambos!***

Distribución de claves

1. A selecciona clave y se la da (físicamente) a B
2. Un tercero la selecciona y las entrega (físicamente) a A y B
3. Si ya comparten una clave, uno selecciona nueva clave y la envía al otro usando clave vieja
4. Si ambos tienen conexión cifrada con un tercero, éste puede seleccionarla y enviársela

Distribución de claves (cont.)

- **El caso 4 puede utilizarse para generar claves de sesión**
- **Clave de sesión:** Se genera para una sesión (intercambio de información) y luego se descarta
- **Clave permanente:** Usada entre entidades para distribuir claves de sesión

1. Host sends packet requesting connection
2. Front end buffers packet; asks KDC for session key
3. KDC distributes session key to both front ends
4. Buffered packet transmitted

FEP = front end processor
KDC = key distribution center

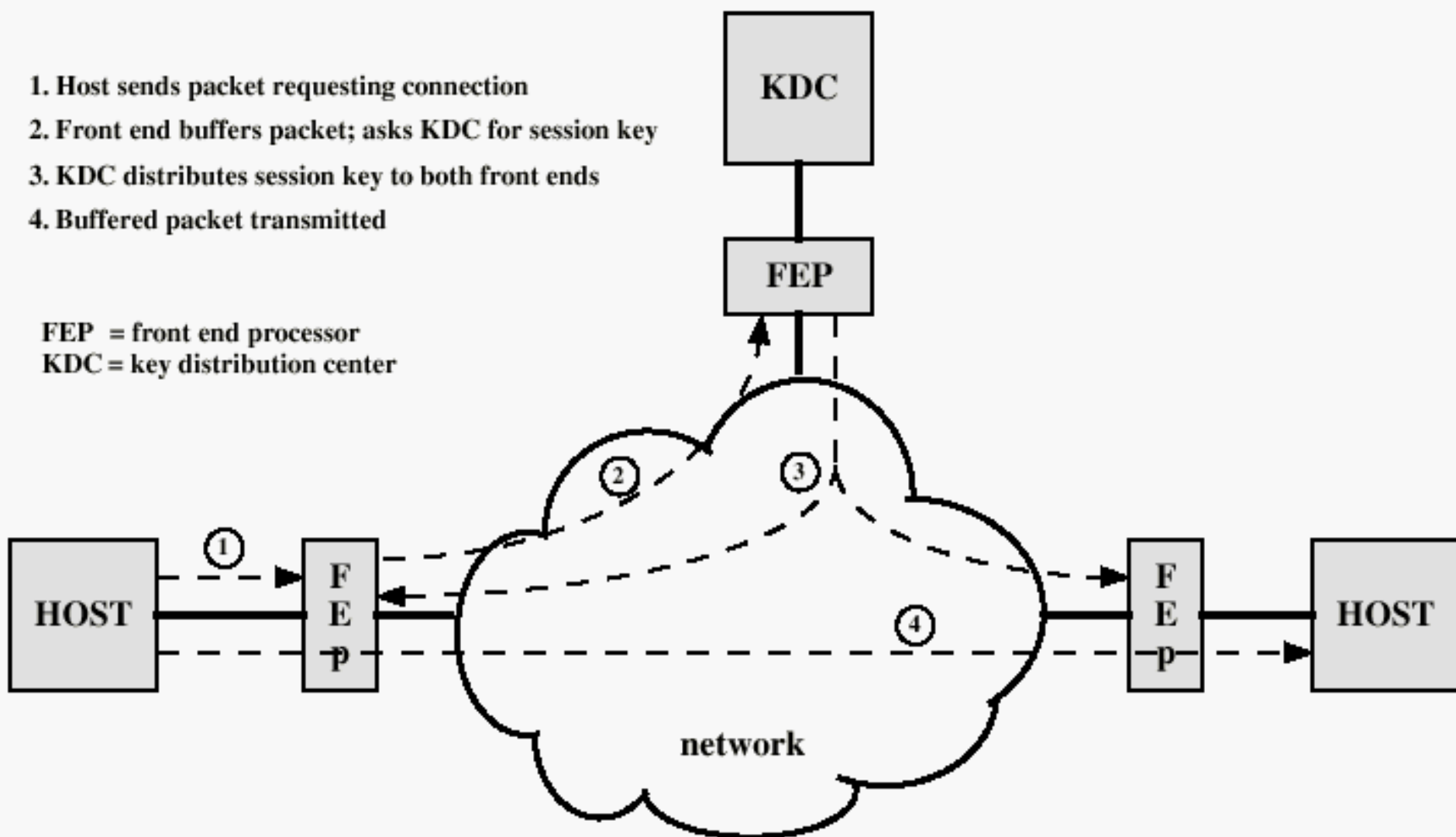


Figure 2.10 Automatic Key Distribution for Connection-Oriented Protocol