UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

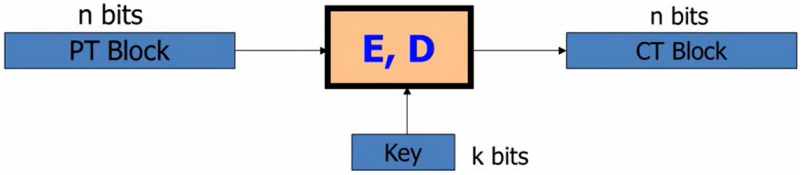
Cifrados de Bloque

Alan Reyes-Figueroa

Criptografía y Cifrado de Información (Aula 08) 17.agosto.2021

Cifrados de Bloque

Los **cifrados de bloque** (*block cipher*) son esquemas más elaborados que los cifrados de flujo. Son importantes, ya que son uno de los "caballos de batalla" de la criptografía actual.



Esquema general de un cifrado de bloque.

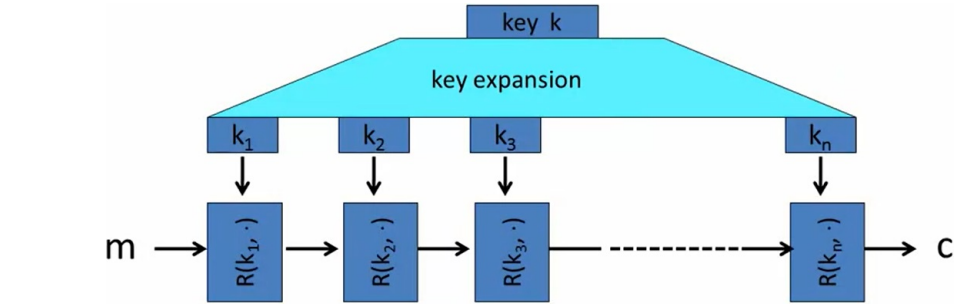
La característica principal es que si E o D reciben como entrada una cadena de n bits, ambos producen una salida exactamente de la misma longitud n bits.

Ejemplos:

- **DES** (3DES): $n = 64$ bits, $k = 168$ bits.
- **AES**: $n = 128$ bits, $k = 128, 192$ ó 256 bits.

Cifrados de Bloque

Los cifrados de bloque típicamente se implementan mediante un esquema iterado, donde se repite un mismo paso o **ronda**.



Esquema de rondas en un cifrado de bloque.

En cada ronda se aplica una función R, llamada **función de ronda** (*round function*):

$$\mathbf{m}_i = R(\mathbf{k}_i, \mathbf{m}_{i-1}), \qquad i = 1, 2, \ldots, n.$$

Cifrados de Bloque

Esquema general:

- Lo primero que ocurre es que la clave $\mathbf{k}$ se "expande" hasta obtener n claves $\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_n$ (*round keys*), todas independientes. Típicamente $\mathbf{k}$ se expande y luego se fragmenta en n claves de menor tamaño.
- Luego, estas claves se unas en la secuencia de n pasos iterados, donde se ejecuta la función de ronda $\mathbf{m}_i = R(k_i, \mathbf{m}_{i-1})$.
- Así, en cada paso $i = 1, 2, \ldots, n$, el mensaje original $\mathbf{m} = \mathbf{m}_0$ se encripta cada vez, utilizando llaves diferentes k_i, hasta completar n pasos de encripción.

Ejemplos:

- DES: $n = 16$ rondas,
- 3DES: $n = 48$ rondas,
- AES-128: $n = 10$ rondas.

Cifrados de Bloque

Comentarios sobre desempeño:

Como es de esperarse, los cifrados de bloque son más lentos que los cifrados de flujo que hemos visto.
Por ejemplo

Cifrado	Tamaño bloque/tamaño clave	Velocidad
RC4		126 MB/sec
Salsa20/12		643 MB/sec
Sosemanuk		727 MB/sec
3DES	64/168	13 MB/sec
AES-128	128/128	109 MB/sec

Velocidades de algunos cifrados comunes (AMD 2.2GHz / Linux)

Aunque son considerablemente más lentos, con los cifrados de bloque es posible hacer más operaciones que ocn los cifrados de flujo, y portanto, mejoran en cuanto a seguridad.

Modleo de Funcionamiento

Consideremos un espacio de entrada $\mathcal{X}$, y un espacio de salida $\mathcal{Y}$ (por ejemplo $\mathcal{X} = \mathcal{Y} = \{0,1\}^n$). Sea $\mathcal{K}$ el espacio de claves, e.g. $\mathcal{K} = \{0,1\}^k$

Definición

*Una **función pseudo-aleatoria** (**PRF**) definida sobre $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, es una función de la forma*

$$F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y},$$

tal que existe un algoritmo eficiente para evaluar $F(\mathbf{k}, \mathbf{x})$.

Definición

*Una **permutación pseudo-aleatoria** (**PRP**) sobre $(\mathcal{K}, \mathcal{X})$, es una función de la forma*

$$E : \mathcal{K} \times \mathcal{X} \to \mathcal{X},$$

tal que

1. *existe un algoritmo determinista eficiente para evaluar $E(\mathbf{k}, \mathbf{x})$.*
2. *la función $E(\mathbf{k}, \cdot) : \mathcal{X} \to \mathcal{X}$ es inyectiva (y portanto uno a uno e invertible).*
3. *existe una forma eficiente de calcular la inversa $D(\mathbf{k}, \cdot) = E^{-1}(\mathbf{k}, \cdot)$.*

Modleo de Funcionamiento

Ejemplos (de PRPs):

- 3DES: $E : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$, con $\mathcal{K} = \{0,1\}^{168}$, $\mathcal{X} = \{0,1\}^{64}$.
- AES-128: $E : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$, con $\mathcal{K} = \mathcal{X} = \{0,1\}^{128}$.

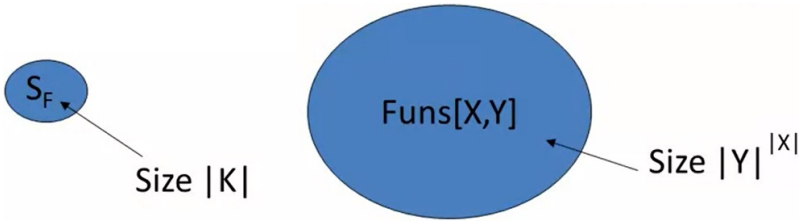
En realidad una PRP es un caso particular de una PRF, con mucha más estructura, específicamente:

- $\mathcal{X} = \mathcal{Y}$,
- $F(\mathbf{k}, \cdot) : \mathcal{X} \to \mathcal{X}$, es una permutación,
- F es eficientemente invertible, toda vez se establece una clave $\mathbf{k}$.

PRFs Seguras

Sea $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ una función pseudo-aleatoria. Consideramos dos conjuntos:

- el conjunto de todas las funciones de $\mathcal{X}$ a $\mathcal{Y}$, denotado $\mathcal{Y}^{\mathcal{X}}$ ó *Funs*$[\mathcal{X}, \mathcal{Y}]$,
- el conjunto de funciones $S_F = \{F(\mathbf{k}, \cdot) : \ \mathbf{k} \in \mathcal{K}\} \subseteq$ *Funs*$[\mathcal{X}, \mathcal{Y}]$.



Intuición: *F* es una PRF **segura** si una función aleatoria en S_F es indistinguible de una función aleatoria en *Funs*$[\mathcal{X}, \mathcal{Y}]$.
(Similar al concepto de PRG seguros): No es posible distinguir entre funciones arbitrarias de $\mathcal{X}$ a $\mathcal{Y}$, y funciones psedo-aleatorias.

PRFs Seguras

Aplicación: Generadores pseudo-aleatorios (PRG seguros).

Sea $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ una PRF segura. Construimos un generador pseudo-aleatori como sigue:

1. Tomamos un espacio de salida de tamaño nt, digamos $\mathcal{C} = \{0,1\}^{nt}$.
2. Dada una clave $\mathbf{k} \in \mathcal{K}$, para cada $i = 1, 2, \ldots, t$, calculamos la función
$$F(\mathbf{k}, i), \qquad i = 1, 2, \ldots, t.$$
3. Definimos el generador pseudo-aleatorio $G : \mathcal{K} \to \{0,1\}^{nt}$ como la concatenación de las funciones anteriores
$$G(\mathbf{k}) = F(\mathbf{k}, 1) + F(\mathbf{k}, 2) + \ldots + F(\mathbf{k}, t).$$

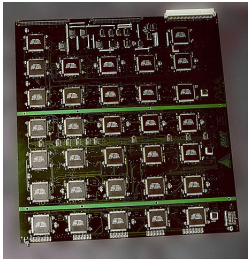
Este generador G es seguro.

Bonus! G es paralelizable (cada función $F(\mathbf{k}, i)$ se puede procesar en un núcleo diferente de forma simultánea) $\Rightarrow$ más rápido.
(En contraste, las PRG que vimos antes son secuenciales, no se pueden paralelizar).

DES

DES (*Data Encryption Standard*).

- Inicios 1970s: Horst Feistel, líder del grupo de encripción en IBM, desarrolla el cifrado Lucifer. Un cifrado de bloques con $k = 128$ bits, $n = 18$ bits.
- 1973: El NBS (*National Bureau of Standards*) lanza a concurso propuestas para cifrados de bloque. IBM somete una variante de Lucifer.
- 1976: Luego de un proceso de modificación y estandarización sobre Lucifer, el NBS establece a DES como estándar federal. Usa $k = 56$ bits, $n = 64$ bits, 16 rondas de bloque.
- 1997: La *Electronic Frontier Foundation* quiebra DES por búsqueda exhaustiva en 56 horas. Se usó un circuito integrado Board-300. En 1999, en sólo 22 horas.



Board300

- 2000: El NIST (*National Institute of Standards*) reemplaza el estándar DES por AES.

DES

Usos: DES es ampliamente utilizado

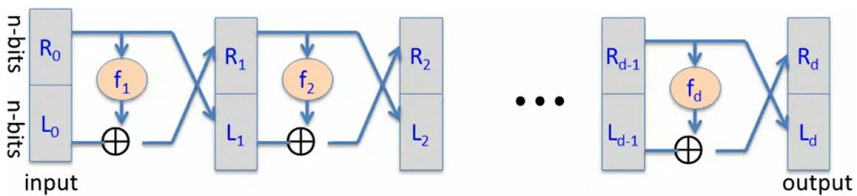
- Banca (*Electronic CLearingHouse*),
- Comercio exterior,
- E-Commerce, Transacciones web,
- Método de cifrado principal en la web.

Hoy, todos estos usan AES en lugar de DES.

Redes de Feistel

DES núcleo principal: Redes de FEISTEL .

Idea muy simple para construir el cifrado de bloques a partir de funciones arbitrarias, $f_1, \ldots, f_d$.
Dadas funciones $f_1, f_2, \ldots, f_d : \{0,1\}^n \to \{0,1\}^n$, queremos construir una función invertible $F : \{0,1\}^{2n} \to \{0,1\}^{2n}$.



Red de Feistel

Observen que hay dos bloques de n bits: L_0 y R_0. En otras palabras, la entrada es en realidad de $2n$ bits.

Redes de Feistel

La entrada R_0 básicamente se copia en la salida L_1, sin ningún cambio en absoluto.
La entrada L_0 se modifican un poco. Básicamente, la entrada R_0 alimenta a la función f_1 y el resultado luego se hace XOR con L_0.

$$\begin{aligned} L_1 &= R_0, \\ R_1 &= f_1(R_0) \oplus L_0. \end{aligned}$$

Esto es la primera ronda de la red de Feistel. Todo esto se repite d veces:

$$\begin{aligned} L_i &= R_{i-1}, \\ R_i &= f_i(R_{i-1}) \oplus L_{i-1}, \qquad i = 1, 2, \ldots, d. \end{aligned}$$

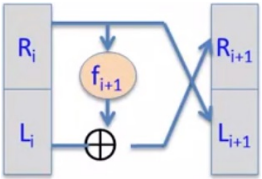
Lo interesante es que este mecanismo es que siempre produce una función F invertible.

Teorema

Para cualesquiera funciones $f_1, f_2, \ldots, f_n : \{0,1\}^n \to \{0,1\}^n$, la red de Feistel $F : \{0,1\}^{2n} \to \{0,1\}^{2n}$ es invertible.

Redes de Feistel

Prueba: Para cada bloque $i = 1, 2, \ldots, d$, tenemos

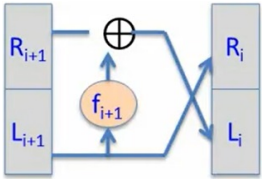


Bloque de Encripción

$$\begin{aligned} L_{i+1} &= R_i, \\ R_{i+1} &= f_{i+1}(R_i) \oplus L_i. \end{aligned}$$

Invirtiendo estas ecuaciones,

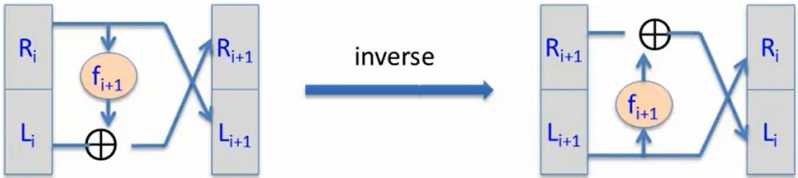
$$\begin{aligned} R_i &= L_{i+1}, \\ L_i &= f_{i+1}(R_i) \oplus R_{i+1} = f_{i+1}(L_{i+1}) \oplus R_{i+1}. \end{aligned}$$



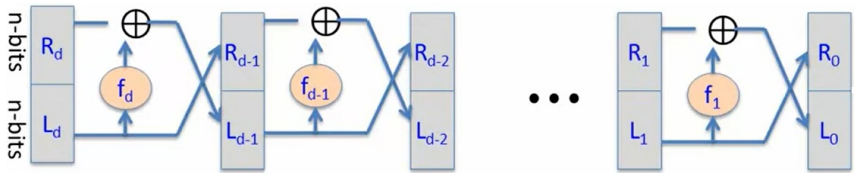
Bloque de Decripción

Redes de Feistel

Esto es el inverso de una ronda de la red Feistel.

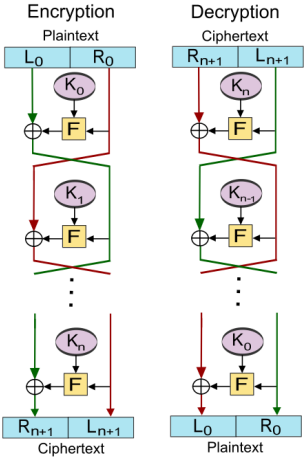


Para ver que toda la red es invertible, basta concatenaer los inversos de cada ronda:



Así, la red inversa es una especie de imagen especular de la red de Feistel inicial.

Redes de Feistel



El inversor es básicamente el mismo circuito:

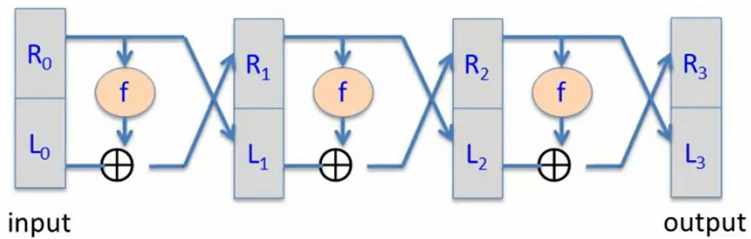
- La única diferencia es que las funciones se toman en orden reverso: $f_d, f_{d-1}, \ldots, f_2, f_1$.
- Desde el punto de vista práctico, esto es atractivo, ya que los constructores de hardware sólo tienen que implementar el algoritmo una vez (reduce costos y espacio).
- Con la misma simplementación se obtiene el encriptor y el decriptor.
- Las redes de Feistel constituyen un método general para producir funciones invertible. Se utilizan en muchos métodos de cifrado.

Teorema (Luby-Rackoff, 1985)

Si $f : \{0,1\}^n \to \{0,1\}^n$ *es una PRF segura, entonces el esquema de Feistel de 3 rondas (3-round Feistel),*

$$F : \mathcal{K}^3 \times \{0,1\}^{2n} \to \{0,1\}^{2n},$$

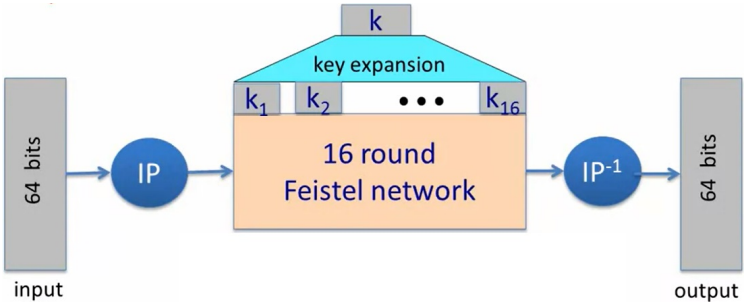
es una PRP segura.



Esquema 3-round Feistel network.

DES

DES: Es básicamente una red de Feistel de 16 rondas.
Tenemos $f_1, f_2, \ldots, f_{16} : \{0,1\}^{32} \to \{0,1\}^{32}$, donde $f_i(\mathbf{x}) = F(\mathbf{k}_i, \mathbf{x})$.

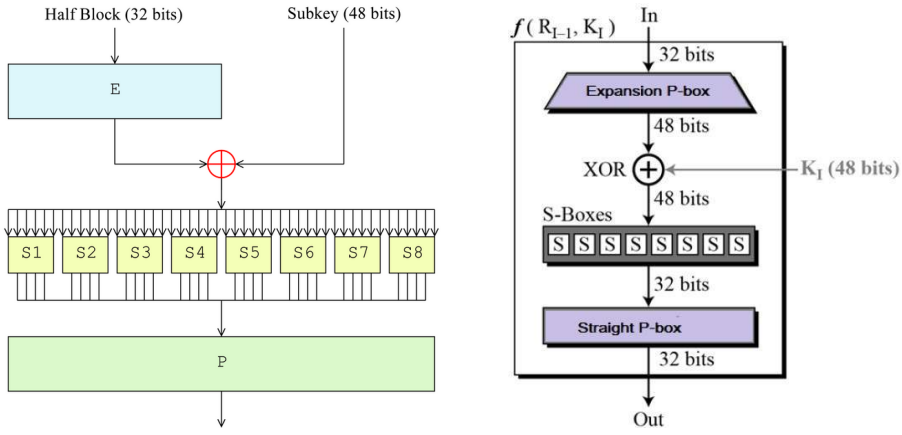


Esquema de cifrado DES.

La clave **k** consta de 56 bits, y pasa por un proceso de expansión, donde se construyen 16 claves de ronda k_i, $i = 1, 2, \ldots, 16$, cada una de longitud 48 bits.

F-function

La función $F(\mathbf{k}_i, \mathbf{x})$:



DES *F*-function.

S-boxes

Lo único que falta especificar son las S-boxes. Éstas son funciones $S : \{0,1\}^6 \to \{0,1\}^4$ que mapean 6 bits en 4 bits.

Se implementan mediante una tabla (de 16×4), como se ilustra en la siguiente figura. Por ejemplo $S_5 : \{0,1\}^6 \to \{0,1\}^4$

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

S-box S_5 en el esquema DES.

S-boxes

Ejemplo: Una mala construcción de una *S*-box.

$$S_i(x_1, x_2, \ldots, x_6) = \left(x_2 \oplus x_3,\ \ x_1 \oplus x_4 \oplus x_5,\ \ x_1 \oplus x_6,\ \ x_2 \oplus x_3 \oplus x_6 \right)$$

$$S_i(\mathbf{x}) = A_i \cdot \mathbf{x} \pmod 2$$

$$\begin{bmatrix} 011000 \\ 100110 \\ 100001 \\ 011001 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_2 \oplus x_3 \\ x_1 \oplus x_4 \oplus x_5 \\ x_1 \oplus x_6 \\ x_2 \oplus x_3 \oplus x_6 \end{bmatrix}$$

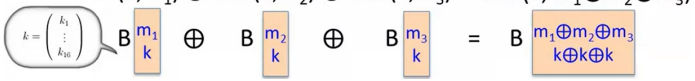
En este caso, si $S_i(\mathbf{x}) = A_i\mathbf{x}$ es lineal, obtendríamos un DES lineal:

$$DES(k,m) = \underset{64}{\overset{832}{B}} \cdot \begin{bmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{bmatrix} = c \pmod 2$$

S-boxes

Lo anterior conduce a una relación entre las entradas de DES:

$$\mathrm{DES}(k,m_1) \oplus \mathrm{DES}(k,m_2) \oplus \mathrm{DES}(k,m_3) = \mathrm{DES}(k, m_1 \oplus m_2 \oplus m_3)$$



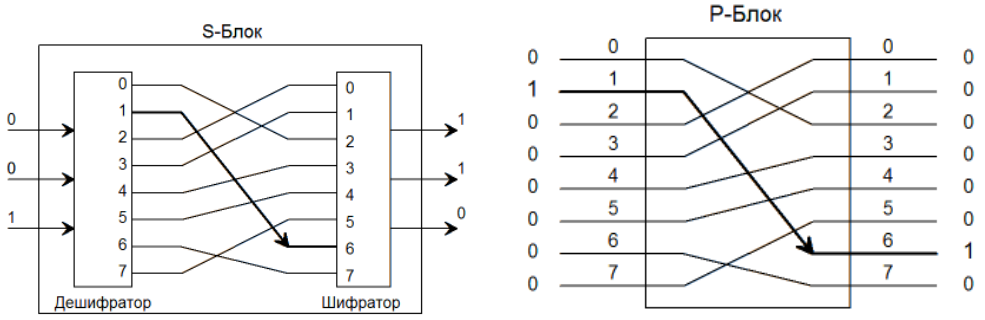
Esta es una relación que no es natural en la funciones aleatorias, de modo que no es deseable para DES.

Importante: Si las *S*-boxes son lineales, entonces se obtiene un DES completamente inseguro.

Teorema (Boneh-Shoup, 1989)

Elegir las S-boxed y las P-boxes de forma aleatoria, conduce a un cifrado DES completamente inseguro (la clave se recupera a partir de $\approx 2^{24}$ *salidas).*

S-boxes



(a) Feistel *S*-box de 3 bits; (b) Feistel *P*-box de 8 bits.