MA-53G Intr. a la Criptografía **Semestre Primavera, 1996**

Cátedra No. 5 : 20 de Agosto

Transcriptor: Mladen W. Nadinic

1 Pendientes

Partimos probando que si en el *one-time pad*, las claves se usan dos veces, entonces el criptosistema deja de ser *perfectamente secreto*.

En este caso $\mathcal{P} = \mathcal{C} = (\mathcal{Z}_2)^{2n}$ y $\mathcal{K} = (\mathcal{Z}_2)^n$

$$\begin{aligned} e_K((u,v)) &= (u+k, v+k)\,, \\ d_K((u,v)) &= (u'+k, v'+k)\,. \end{aligned}$$

donde k es la clave, $u, v, u', v' \in (\mathcal{Z}_2)^n$, y $x + y = (x_i +_2 y_i)_{i=1}^n$. Veamos, si es que[1] $p_{\mathcal{P}}((u,v)) = p_{\mathcal{P}}((u,v)/(u',v'))$, $\forall\ u, v, u', v' \in (\mathcal{Z}_2)^n$.

Supongamos que $u' = v'$ y $u \neq v$, y que $p_{\mathcal{P}}$ asigna la misma probabilidad a todos los elementos en $\mathcal{P}$, entonces

$$\begin{aligned} p_{\mathcal{P}}((u,v)) &= \frac{1}{|\mathcal{P}|} = \frac{1}{2^{2n}}\,, \\ p_{\mathcal{P}}((u,v)/(u',v')) &= p_{\mathcal{K}}(u' = u+k, v' = v+k) = 0\,. \end{aligned}$$

Puesto que estas dos últimas expresiones son distintas, concluimos que usar la clave dos veces en el criptosistema de *one-time pad* hace que este criptosistema deje de ser perfectamente seguro.

2 Criptosistema Producto

Shannon [1949] introdujo la idea de iterar un criptosistema con la esperanza que esto condujese a criptosistemas más dificiles de quebrar. Esta idea a afectado el diseño de criptosistemas en uso actualmente, como el DES. En téminos más formales, se tiene:

[1] En la clase anterior se definio de esta manera a un criptosistema perfectamente secreto.

Definición 1 *Criptosistema Producto*

Sean $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *y* $S' = (\mathcal{P}', \mathcal{C}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$*, Definimos el criptosistema producto, denotado por* $S' \times \ S''$ *tal que:* $\mathcal{P} = \mathcal{P}'$ $\mathcal{C} = \mathcal{C}''$, $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$, $e_K(x) = e_K(e_{K'}(x))$, $d_K(y) = d_{K'}(d_{K''}(y))$, $K = (K',\ K'')$.

Si $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ y además $\mathcal{P} = \mathcal{C}$. Escribimos S^2 como el producto del criptosistema S, por si mismo, es decir, $S \times \ S$. Y en general denotamos el producto *n-ésimo* del criptosistema S por $S^n = \ S \times \ S \times \ldots \times \ S$; observar que el producto es asociativo.

Ahora introducimos el concepto de *criptosistemas equivalentes*.

Definición 2 *Decimos que los criptosistemas* $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *y* $S' = (\mathcal{P}', \mathcal{C}', \mathcal{K}', \mathcal{E}', \mathcal{D}')$*, son equivalentes bajo las distribuciones a priori* $p_{\mathcal{P}}$*,* $p_{\mathcal{P}'}$ *y las distribuciones de claves* $p_{\mathcal{K}}, p_{\mathcal{K}'}$ *si:* $a\colon\ Sop(p_{\mathcal{P}}) = \mathcal{P} \to \ Sop(p_{\mathcal{P}'}) = \mathcal{P}'$, $b\colon\ Sop(p_{\mathcal{C}}) = \mathcal{C} \to \ Sop(p_{\mathcal{C}'}) = \mathcal{C}'$, $g\colon\ Sop(p_{\mathcal{K}}) = \mathcal{K} \to \ Sop(p_{\mathcal{K}'}) = \mathcal{K}$ *son biyectivas y ,*

$$
\begin{aligned}
p_{\mathcal{P}}(x) &= p_{\mathcal{P}'}(a(x)) \ , \forall\, x \in \mathcal{P}\,, \\
p_{\mathcal{K}}(k) &= p_{\mathcal{K}'}(g(K)) \ , \forall\, k \in \mathcal{K}\,, \\
e_{g(k)}(a(x)) &= b(e_K(x))\, \forall\, x \in \ Sop(p_{\mathcal{P}}), \forall\, k \in \ Sop(p_{\mathcal{K}})\,,
\end{aligned}
$$

Decimos que $S = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *es idempotente bajo la distribución a priori* $p_{\mathcal{P}}$ *y la distribución de claves* $p_{\mathcal{K}}$*, si* S^2 *es equivalente a* S *bajo las distribuciones* $p_{\mathcal{P}}$ $p_{\mathcal{P}}$ *y distribución de claves* $p_{\mathcal{K} \times \mathcal{K}}, p_{\mathcal{K}}$.

Ejercicio 1 *El criptosistema afín es idempotente.*

3 DES (Data Encryption Standard)

3.1 Un poco de Historia

En 1972 la NBS (National Bureau of Standards), hoy en día la NIST (National Institute of Standards and Technology), inicio un programa para proteger computadores y comunicaciones. En 1973 la NBS publicó una solicitud pública para proposiciones conducentes a establecer algoritmos criptografícos. Entre los criterios de diseño estaban:

- el algoritmo debe proveer un alto grado de seguridad
- el algortimo debe estar completamente especificado y ser fácil de entender
- la seguridad del algoritmo debe residir en la clave; la seguridad no debe depender de mantener el algoritmo secreto
- el algoritmo debe estar disponible para todos los usuarios
- el algoritmo debe poder implementarse económicamente a nivel de hardware
- el algoritmo debe ser eficiente
- el algoritmo debe ser exportable
- el algoritmo debe poder validarse.

IBM desarrollo DES (Data Encryption Standard) como una modificación de un sistema conocido como LUCIFER. La NBS pidió ayuda a la NSA (National Security Agency) para evaluar el algoritmo. DES se publicó en 1975, y se adoptó como standard después de un intenso debate en 1976. El debate se centraba en las modificaciones sugeridas por NSA. Estas consistieron basicamente en :

- Reducción del tamaño de la clave de 128 bits a 56 bits.
- Modificación de ciertas partes del algoritmo que en la época vieron misteriosas pero que se aclararon a comienzos de 1990.

La NBS (NIST) re-evalua el DES aproximadamente cada cinco años. La próxima evaluación es en 1998, donde se espera sea descartado.

3.2 Descripción de DES

Una completa descripción del DES fue dada por la *Federal Information Processing Standards Publication 46* el 15 de enero de 1977. Basicamente el DES es un criptosistema en que $\mathcal{P} = (\mathcal{Z}_2)^{64}$ es el conjunto de textos planos y $\mathcal{K} = (\mathcal{Z}_2)^{56}$, es el conjunto de claves.

El algoritmo consta de tres etapas(figuras 1 y 2):

- Etapa Inicial IP: $(\mathcal{Z}_2)^{64} \rightarrow (\mathcal{Z}_2)^{64}$, es una permutación fija. Escribimos $x_0 = IP(x) = L_0 R_0$, donde L_0 y R_0 son el primer y el segundo bloque, cada uno de 32 bits de largo.

- Etapa Intermedia Se realiza 16 veces el siguiente computo, para obtener L_i y R_i, $1 < i < 16$:

$$\begin{aligned} L_i &= R_{i-1}\,, \\ R_i &= L_{i-1} +_2 f(k_i, R_{i-1})\,. \end{aligned}$$

  Más adelante se describe como obtener las claves k_i de 48 bits, apartir de una clave k de 64 bits.

  Por ahora nos centramos en la función f que es donde descansa la fortaleza del criptosistema.

  La función f tiene dos argumentos k_i y R_{i-1}, como se puede apreciar en la figura 3. El primero de estos argumentos es sometido a una expansión E, que le agrega 16 bits más. Este último resultado se opera con una operación xor, con el valor de la clave k_{i-1}. El resultado de esta operación es dividido en 8 bloques de 6 bits cada uno. Cada bloque es pasado por una *caja-S*, que entrega como resultado un bloque de 4 bits de tamano. Finalmente los últimos 32 obtenidos se permutan.[2]

  Como se puede apreciar la piedra angular de la función f son las famosas *cajas-S*, que son una de las tantas fuentes de controversias de la DES.

- Etapa Final Notar que el orden de L_{16} y R_{16} estan invertidos. Y además se aplica una permutación inversa a la permutación inicial, es decir, $y = IP^{-1}(R_{16}L_{16})$.

4 Controversia en torno a el DES

Como ya se senalo el DES ha estado, desde su origen, envuelto en muchas controversias [Diffie y Hellman, 1977; Kolata, 1977].[3]

[2]Ejemplos del operarar de las cajas-S se puede encontrar en D. R. Stinson , "Cryptography: Theory and cryptography". Capítulo 3.

[3]Véase como referencia: "Redes de Ordenadores", Andrew Tanenbaum, 2 edición.

Varios científicos de computación han proclamado que 56 bits representa un tamaño muy pequeno para una clave, haciendo que sea fácilmente atacada aplicando fuerza bruta. De hecho en 1993 Wiener detallo una máquina que por un millón de dolares (poco en el contexto actual de una gran compañía) permite descubrir la clave de encripción en un promedido de 3 horas. Otra cosa que ha molestado a numerosos científicos es el rechazo de IBM para hacer públicas las razones por las cuales se escogieron para el cifrador unas *cajas-S* específicas. Lo único que IBM ha comentado a respecto es que la NSA le solicitó que mantuviera en secreto los principios de diseno. Sin el conocimiento de los principios de diseño resulta díficil excluir la posibilidad de que exista algún truco por medio del cual el cifrado puede llegar a romperse fácilmente. También hubo un incidente, que algunos investigadores interpretaron como un intento realizado por un empleado del gobierno para impedir la publicación de investigaciones academicas enfocadas a desarrollo de cifradores más poderosos [Shapley y Kolata, 1977].[4]

5 Como mejorar el DES

DES, al igual que otros cifradores, se le puede aplicar ciertas técnicas para hacer más díficil su violación. Entre estas destacamos.

5.1 Cifrador Nulo

Esta técnica suguiere incluir caracteres aleatorios en el texto plano, de acuerdo con una regla bien definida (por ejemplo, todos los n-ésimos caracteres son reales y el resto sólo *ruido*. En términos formales se tiene un criptosistema S como una *tupla* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, R, \mathcal{D})$, donde lo nuevo es la *regla* R para agregar basura al texto plano. Podemos definir R como: $R:\ \mathcal{P} \rightarrow\ \mathcal{P}_R$ donde $\mathcal{P}_R$ es el conjunto de todos los textos planos a los cuales se le ha agregado basura en ciertas posiciones. Obviamente debe existir R^{-1}, para poder, después de desencriptar el texto cifrado, elimminar la basura. Este principio, que se conoce como *cifrador nulo*, tiene la desventaja de desperdiciar *ancho de banda* en el sistema. Pero su fortaleza radica en que es muy difícil de descifrar, porque la posición de los caracteres reales y de los mensajes se conserva en secreto, y se cambia cuando se modifica la clave. En líneas privadas alquiladas, vale la pena pensar en la importacia de transmitir basura, cuando de otra manera la línea quedaría inactiva.

[4]Véase como referencia: "Redes de Ordenadores", Andrew Tanenbaum, 2 edición.

5.2 Cifrador de Flujo.

[5] Otra manera de lograr que el criptoanálisis del DES sea más difícil, es la de hacerla funcionar como si fuera un *cifrador de flujo*, como se muestra en la figura 4, en lugar de que opere como un *cifrador de bloque*, como se le describio en las secciones anteriores. Cuando se le utiliza como un cifrador de flujo, tanto el trasmisor como el receptor operar sus circuitos integrados DES en modo de cifrado. Cada uno de los circuitos integrados DES cuenta con registro de entrada de 64 bits, el cual opera como un registro de desplazamiento, y un registro de salida de 64 bits, que no lo hace así. En el momento que llega un car/'acter de texto plano, se le realiza un *xor* con los 8 bits del registro de salida, O1. (Los correspondientes a O2 hasta O8, nunca se utilizan). De esta manera en carácter así creado se transmite al receptor y se introduce en el registro de desplazamiento de entrada, sacando a I8 del registro. Después, el circuito integrado (DES) es activado, y la salida se calcula para la nueva entrada. Por otra parte, en el extremo receptor, al carácter que llega se le aplica primero un *xor* con O1 (produciendo texto plano) , y después se introduce en I1. Si el transmisor y el receptor arrancan con registros idénticos , éstos permanecerán iguales para siempre. La propiedad de los cifradores de flujo que los hace muy valiosos, es que O1 depende de la historia completa del texto plano, por lo que la repetición de un patrón en el texto plano no generará un patrón repetido en el texto cifrado. Los cifradores de flujo, también, son muy apropiados para utilizarse en terminales, porque no tienen que colectar ocho caracteres antes de emitir un texto cifrado. Cada golpe de tecla ganarará inmediatamente una salida, a medida que se teclea.

[5] También conocido como cifrador por chorro.

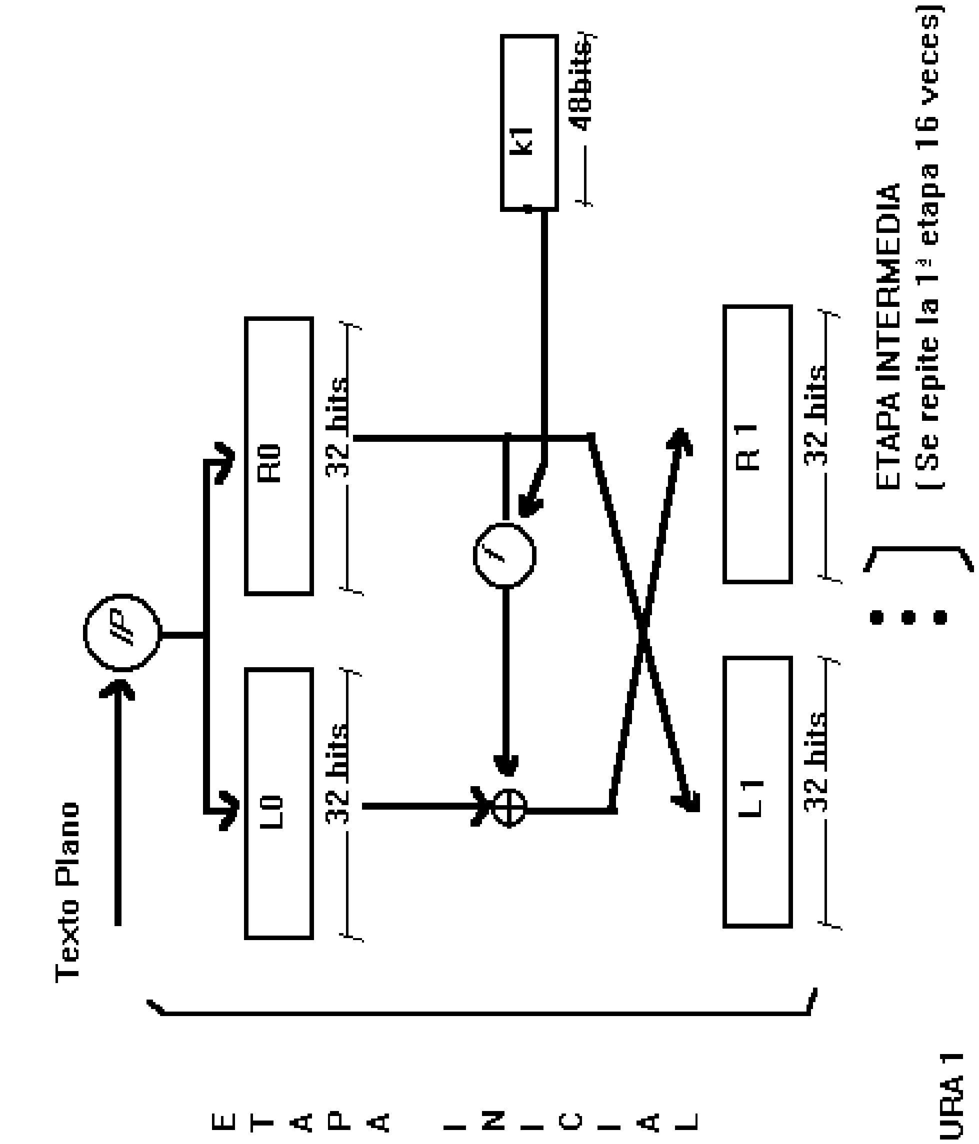

Texto Plano
IP
L0
R0
32 bits
32 bits
k1
48bits
f
L 1
R 1
32 bits
32 bits
ETAPA INTERMEDIA
(Se repite la 1ª etapa 16 veces)
E T A P A I N I C I A L

FIGURA 1

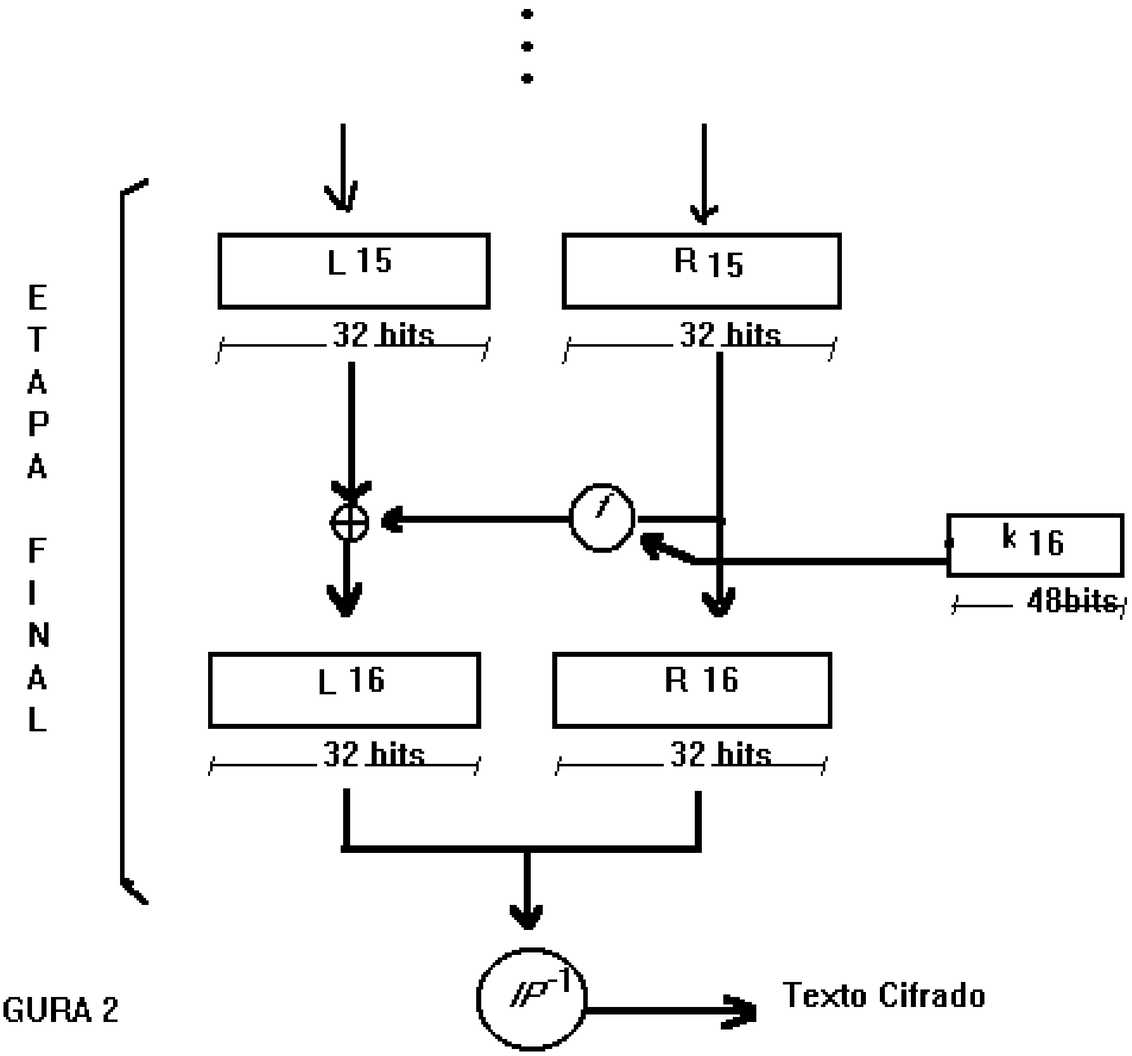
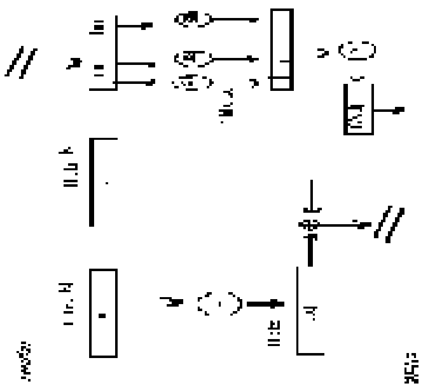

ETAPA FINAL
L 15
32 bits
R 15
32 bits
f
k 16
48bits
L 16
32 bits
R 16
32 bits
IP-1
Texto Cifrado

FIGURA 2



.

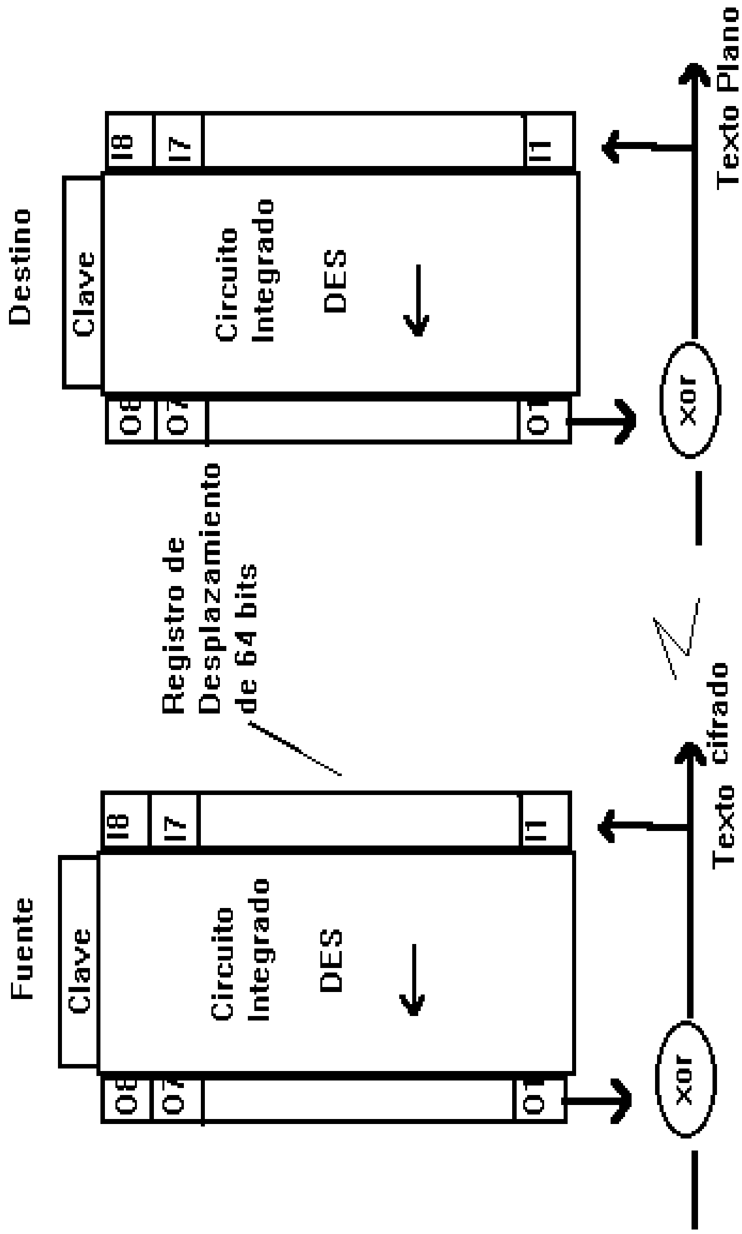

Fuente
Clave
Circuito Integrado
DES
O8
O7
O1
I8
I7
I1
xor
Texto cifrado
Registro de Desplazamiento de 64 bits
Destino
Clave
Circuito Integrado
DES
O8
O7
O1
I8
I7
I1
xor
Texto Plano

FIGURA 4