MA-53G Intr. a la Criptografía	**Semestre Otoño, 1998**

Cátedra No. 9 : 13 de Abril

Transcriptor: Gonzalo Navarro[1]

1 Compromiso Tiempo-Memoria (cont.)

Dado un texto plano X conocido y un texto $Y = e(X)$ encriptado, la alternativa de fuerza bruta consiste en probar todas las claves K hasta hallar la que cumpla $Y = e_K(X)$. Esto no necesita espacio extra pero requiere tiempo $\widetilde{O}(|\mathcal{K}|)$ [2]. Por ejemplo en DES requeriremos 2^{56} encripciones, lo que en 1992 con un chip especializado de US$ 300 demoraría unos 136 años (claro que se puede correr en varios chips en paralelo, por ejemplo demoraría un mes teniendo 1600 chips, a un costo de casi un millón de dólares).

Por otro lado, dado un X *fijo*, podríamos encriptar X con todas las claves K posibles y almacenar los pares $(e_K(X), K)$ previamente (indexados por la primera componente). Una vez hecho este preprocesamiento podríamos fácilmente romper cualquier criptosistema dado mediante el cálculo de $Y = e(X)$ y luego la búsqueda de Y en la tabla, de lo que se obtendría K. Esta búsqueda toma poco tiempo porque la tabla está indexada por esa componente, de modo que necesitamos un preprocesamiento de tiempo $\widetilde{O}(|\mathcal{K}|)$, espacio $\widetilde{O}(|\mathcal{K}|)$ y luego tiempo $O(\log |\mathcal{K}|)$ por cada clave. Este espacio equivale aproximadamente a 2^{60} bytes, o 2^{30} Gb.

Obviamente ninguna de las dos ideas es factible. La pregunta es si será posible una estrategia intermedia, que use una cantidad razonable de tiempo y de memoria. Esta es la base de la idea que se detalla a continuación.

Supongamos que elegimos una *función de reducción* R

$$R \; : \; \mathcal{C} \;\longrightarrow\; \mathcal{K}$$

que convierte textos cifrados en claves de alguna manera (ej. en DES podría eliminar 8 de los 64 bits para que quedaran 56). Dada R se define g:

$$g \; : \; K \;\rightarrow\; K \qquad\qquad g(k) = R(e_k(x)) \; ,$$

donde x es un texto plano *fijo*.

El objetivo de g es simplemente tener una forma determinística pero que se comporte como aleatoria para generar claves a partir de una clave inicial. Para obtener esto se usa $e_k(x)$, que si el criptosistema es bueno debería comportarse así (en forma determinística y aparentemente aleatoria). El objetivo de R es permitir la composicionalidad de g, dado que la salida de $e()$ puede no ser del mismo tipo que la entrada.

[1] Estos apuntes están basados en la transcripción de Gregorio González (27/8/96)

[2] Recordar que $\widetilde{O}(X)$ significa $O(X \times polylog(X))$.

Teniendo esto, elegimos m claves *iniciales* $K_1, ..., K_m$ al azar (o aparentemente al azar[3]). Una vez obtenido K_i, denominamos $X(i,0) = K_i$, y calculamos $X(i,j) = g(X(i,j-1))$, para j desde 1 hasta t. Esto produce una matriz de claves, a saber

$$\begin{array}{ccccccccc}
K_1 = X(1,0) & \longrightarrow & X(1,1) & \longrightarrow & X(1,2) & \longrightarrow & .... & \longrightarrow & X(1,t) \\
K_2 = X(2,0) & \longrightarrow & X(2,1) & \longrightarrow & X(2,2) & \longrightarrow & .... & \longrightarrow & X(2,t) \\
... & ... & ... & ... & ... & ... & ... & ... & ... \\
K_m = X(m,0) & \longrightarrow & X(m,1) & \longrightarrow & X(m,2) & \longrightarrow & .... & \longrightarrow & X(m,t)
\end{array}$$

Es importante entender, sin embargo, que salvo la primera y la última columna, nada de esta matriz se almacena en memoria. Los resultados de las columnas intermedias se usan sólo para llegar al $X(i,t)$.

Una vez obtenida la última columna a partir de la primera, se genera una tabla que almacene los pares $(X(i,t), X(i,0))$, indexada por la primera componente. Con esto concluye el preprocesamiento.

Supongamos ahora que nos presentan un texto cifrado y, que corresponde a encriptar nuestro x con alguna clave desconocida (o sea $y = e_K(x)$). Reducimos el texto cifrado y a una clave (notar que esta reducción no tiene más significado que, por ejemplo, sacarle algunos bits para que quede del largo adecuado, es un mapeo casi aleatorio). Es decir, calculamos $R(y)$. La apuesta es que, con un poco de suerte, existe un $X(i,j)$ en la matriz que cumple $R(y) = X(i,j)$ (para algún $j > 0$, i). Si esto ocurre, lo que tenemos es que

$$R(y) = X(i,j) = g(X(i,j-1)) = R(e_{X(i,j-1)}(x))$$

y por lo tanto es bastante posible que $y = e_k(x)$, donde $k = X(i,j-1)$. Es decir, $X(i,j-1)$ tiene buenas posibilidades de ser la clave.

Cómo buscamos en esta matriz, si no la hemos almacenado? Tenemos que se cumple que si $R(y) = X(i,j)$, entonces $g^{t-j}(R(y)) = X(i,t)$. Es decir, si iteramos g suficientes veces (máximo t) sobre $R(y)$ llegaremos a $X(i,t)$. Estos $X(i,t)$ no sólo están almacenados sino que tenemos la tabla indexada por ellos. Nótese que la inversa no es cierta, es decir que podemos llegar a un $X(i,t)$ por iteración y sin embargo nuestra $R(y)$ puede no estar realmente en la fila (dado que g no es inyectiva). Sin embargo, hay una buena posibilidad. De modo que el algoritmo es

1. Dado $y \in \mathcal{C}$
2. $y_0 \leftarrow R(y)$
3. for $j = 1..t$, calcular $y_j = g(y_{j-1})$
4. for $j = 1..t$,
5.   buscar y_j en la primera coordenada de la tabla
6.   si se halló, reportar la posible clave y_{t-j}

En general se recomienda construír s tablas distintas (con distintas funciones R). En este caso, la memoria requerida es $\widetilde{O}(sm)$ (dado que cada tabla tiene dos columnas, su espacio es $\widetilde{O}(m)$).

[3]No usar $g()$ para simular este azar, pues entonces $X(i+1,0) = X(i,1)$.

El tiempo requerido es el de iterar t veces sobre una búsqueda indexada, que toma $\widetilde{O}(t)$, lo que para las s tablas totaliza $\widetilde{O}(ts)$. Para el precálculo hay que generar todas las tablas (aunque no se almacenen), de modo que el costo es $\widetilde{O}(smt)$.

La consideración final es la probabilidad de hallar la clave. En la práctica el método no es más que una prueba sobre $s \times m \times t$ claves casi aleatorias. Por lo tanto, la probabilidad de que ninguna de ellas sea la que buscamos es

$$\left(1 - \frac{1}{|\mathcal{K}|}\right)^{smt}.$$

Se recomienda usar este método con $t = m = s = |\mathcal{K}|^{1/3}$, con lo cual la probabilidad de no encontrar la clave es

$$\left(1 - \frac{1}{|\mathcal{K}|}\right)^{|\mathcal{K}|} = e^{-1}\ (1 + O(1/|\mathcal{K}))$$

dado que $e^{-1} \approx 37\%$ no es despreciable, parece más recomendable hacer crecer s. Dada una probabilidad de error tolerada ϵ, vamos a elegir $s = a|\mathcal{K}|^{1/3}$, $m = t = |\mathcal{K}|^{1/3}$ para obtener esa probabilidad. En este caso la probabilidad de error es e^{-a}, que igualado a ϵ nos da $a = \ln(1/\epsilon)$. Esta "constante" afecta tanto al tiempo como al espacio del algoritmo. Se podría afectar sólo al tiempo o al espacio si se agrandaran m o t en vez de s.

En el caso $a = 1$, con DES la cantidad de memoria requerida es cercana a 2^{11} Gb, y el tiempo sería 2 horas y cuarto con la máquina referida antes. Si en cambio usáramos $s = 1,\ t = |\mathcal{K}|^{2/3},\ m = |\mathcal{K}|^{1/3}$ tendríamos la misma probabilidad de hallar la clave (63%) pero ahora el espacio sería de sólo 4 Mb, y el tiempo sería el mismo de antes.

Este cálculo apunta a que no es buena idea hacer crecer s, y $s = 1$ parecería óptimo. Es posible que la razón para elegirlo grande tenga que ver con la poca aleatoriedad real de las tablas, donde ciertas claves no pueden aparecer de ninguna forma.

2 Criptoanálisis Diferenciable

2.1 El Método de Criptoanálisis Diferenciable

El método de Criptoanálisis Diferenciable fue presentado por Eli Biham y Adi Shamir en 1990.[4] Este método permite realizar un ataque de texto plano elegido que resulta (en teoría al menos) más eficiente que un ataque de fuerza bruta contra el sistema DES.

El criptoanálisis diferenciable se basa en el análisis de la propagación de diferencias entre dos textos planos a través de las funciones del criptosistema cuando se usa la misma clave. Básicamente, se toma un par de textos planos, posiblemente al azar, pero con una cierta "distancia" elegida. Según la clave usada, la diferencia en la entrada llevará con mayor o menor probabilidad a ciertas diferencias en la salida (texto cifrado). El criptoanálisis diferenciable aprovecha el hecho que estas

[4] E. Biham y A.Shamir, 1.- *"Differential Cryptanalysis of DES-like Cryptosystems"* , Advances in Cryptology–CRYPTO '90 Proceedings, Springer Verlag, 1991, pp. 2-21; 2.- *"Differential Cryptanalysis of DES-like Cryptosystems"* , Journal of Cryptology, v.4, n. 1, 1991, pp 3-72; 3.- *"Differential Cryptanalysis of FEAL and N-Hash"* , Advances in Cryptology–EUROCRYPT '91 Proceedings, Springer-Verlag, 1991, pp. 1-16.

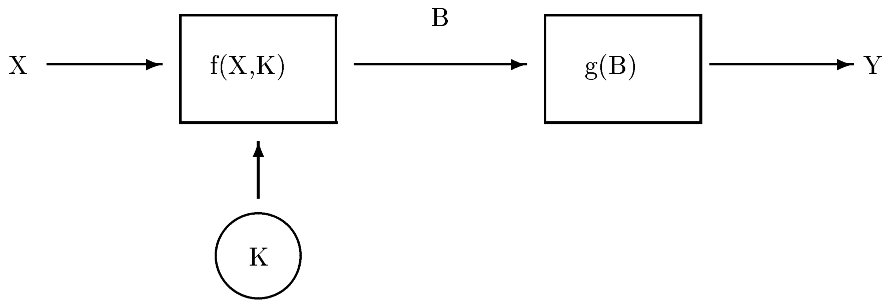

Figura 1: Estructura de un criptosistema atacable mediante criptoanálisis diferenciable.

distribuciones en la diferencia de salida no son uniformes. Al obtener la diferencia entre los textos cifrados, se puede descartar aquellas claves con las cuales esta diferencia tiene probabilidad cero. A medida que se analizan más pares de textos, se irán descartando claves hasta que unas pocas claves aparecerán como las únicas posibles. Se iterará entonces probando todas las claves de ese pequeño conjunto (en un caso ideal podríamos obtener una única clave posible, que sería necesariamente la que buscamos).

Otra posible alternativa, si no hay claves con probabilidad cero, usar las probabilidades para probar las claves en cierto orden, de la más a la menos probable.

El método sirve para atacar un criptosistema o una parte de él que tenga la estructura que se muestra en la Figura 1.

Hemos dicho que usamos una "distancia" elegida entre textos planos y cifrados. Esta distancia (que representaremos con el operador de diferencia, "$-$") se define según el criptosistema a atacar. Para el criptosistema DES la diferencia adecuada es el XOR, denotado $\oplus$. Recordemos que $(\mathcal{Z}_2)^n$ con este operador es un grupo abeliano, es decir tenemos conmutatividad, asociatividad, neutro (0) e inverso ($X^{-1} = X$). Dado este valor del inverso, el operador es nilpotente ($X \oplus X =$ neutro).

La condición que debe cumplir la operación diferencia (y que se usa para elegirla) es la siguiente:

$$\triangle B = B - B' = f(X,k) - f(X',k) = \triangle X.$$

es decir, que al aplicar la función f sobre X, X', su diferencia no se altere (ver Figura 2).

Dado que el único punto donde la clave interviene es en la función f, esto implica inmediatamente que podemos montar un ataque sobre g. Para este ataque podemos elegir la diferencia ($\triangle B = \triangle X$), pero no podemos elegir el "texto". Esto se debe a que la entrada de g es B, que no podemos controlar.

Llamaremos "texto intermedio" a B. Se define entonces:

$$IN(\delta, \tau) = \{B : g(B) - g(B - \delta) = \tau\}.$$

como el conjunto de todos los textos intermedios que convierten una diferencia δ en la entrada en una diferencia τ en la salida (diremos que estos textos intermedios "llevan δ a τ"). En el caso de

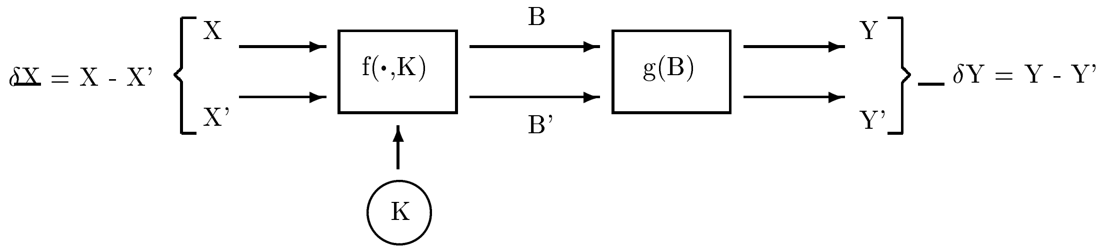

Figura 2: Propagación de las diferencias a través del criptosistema.

DES:

$$IN(\delta,\tau) = \{B : g(B) \oplus g(B \oplus \delta) = \tau\},$$

Y construímos:

$$K(X,\delta,\tau) = \{k : f(X,k) \in IN(\delta,\tau)\}.$$

como el conjunto de todas las claves que aplicadas sobre X pueden generar un texto intermedio que lleve de δ a τ.

Notemos que si encriptamos $X,\ X' = X - \delta$ y obtenemos $Y,\ Y'$ tal que $\triangle Y = \tau$ entonces necesariamente $k \in K(X,\delta,\tau)$. La recíproca también es cierta.

La idea es fijar δ y generar pares X, X', pasarlos por el criptosistema y determinar el τ resultante. Sólo las claves que pertenecen a $K(X,\delta,\tau)$ pueden ser la correcta. Si hacemos esto varias veces achicaremos el conjunto. Este conjunto representará aquellas claves para las cuales las diferencias obtenidas tienen probabilidad mayor que cero.

Lo que se desea es que restringiendo sucesivamente las claves posibles mediante los $K(X,\delta,\tau)$ terminemos con un conjunto manejable de candidatos que se puedan verificar exhaustivamente. El principal problema es que debemos tener una forma eficiente de obtener $K(X,\delta,\tau)$ sin recorrer todo el espacio, pues sino el costo de calcular $K(X,\delta,\tau)$ se acerca al de la búsqueda en todo el espacio de claves.

El ataque procede de la siguiente forma:

1. $S \leftarrow \mathcal{K}$
2. Mientras $|S| >$ tamaño razonable
3. Elegir $X, X' = X - \delta$
4. Determinar Y, Y' y $\tau = \triangle Y$
5. $S \leftarrow S \cap K(X,\delta,\tau)$
6. Probar exhaustivamente las claves de S

2.2 Problemas del Criptoanálisis Diferenciable

A pesar de su aparente simplicidad, este ataque puede resultar impráctico en varios casos. Primero que nada, hay una probabilidad bastante baja de tener éxito hasta que se pasa de cierto umbral, es decir, mientras no se verifiquen suficientes pares de textos es imposible distinguir la clave correcta de las demás que no se han descartado. Además, este ataque es costoso desde el punto de vista del almacenamiento de datos, pues requiere almacenar un subconjunto arbitrario de las $|\mathcal{K}|$ claves posibles (ej. de 2^{56} claves en el caso de DES). Si el subconjunto es S, se necesitan a lo menos $\log_2 \binom{|\mathcal{K}|}{|S|}$ bits para almacenarlo, lo cual en el peor caso ($|S| = |\mathcal{K}|/2$) implica $|\mathcal{K}|$ bits.

Otro problema es el número de textos planos necesario para descartar casi todas las claves. Se puede probar que un criptosistema es inmune al criptoanálisis diferenciable demostrando que requiere analizar más textos planos que los que es posible generar con el tamaño de bloque utilizado. O al menos que se deben analizar tantos textos planos que el ataque no es mucho más eficiente que la búsqueda exhaustiva.

Finalmente, es imperativo tener un buen algoritmo para calcular los conjuntos $K(X, \delta, \tau)$ sin proceder por búsqueda sobre todas las claves, pues sino el enfoque pierde la gracia. Una vez que S es relativamente pequeño es fácil calcular su intersección con el nuevo $K(X, \delta, \tau)$ mediante recorrer S. Pero el comienzo es difícil y la solución parece depender de los detalles específicos del criptosistema que se ataca.

2.3 Un Ejemplo: Criptoanálisis de 3-DES

Veremos ahora un ejemplo de criptoanálisis diferencial aplicado a DES con tres iteraciones sólamente. Este criptosistema calcula a partir de una clave K de 56 bits, tres subclaves K_1, K_2 y K_3. Cada una de ellas tiene 48 bits elegidos de K, y se conoce cuáles son estos bits. De modo que si una sóla de las claves K_i fuera conocida, tendríamos automáticamente 48 bits de K, y los restantes 8 se podrían obtener por búsqueda exhaustiva sobre las $2^8 = 256$ posibilidades.

Ignoraremos las permutaciones de entrada y salida de DES, pues son fáciles de deshacer (ya que son de dominio público, o sea conocidas). Vamos a tratar de descubrir la tercera subclave K_3. Recordemos que:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i) \quad i = 1, 2, 3. \end{aligned}$$

Donde la entrada es L_0R_0 y la salida es L_3R_3. Luego:

$$\begin{aligned} R_3 &= L_2 \oplus f(R_2, K_3) \\ &= R_1 \oplus f(R_2, K_3) \\ &= L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3). \end{aligned}$$

Sea entonces $L_3'R_3'$ otra salida, correspondiente a la entrada $L_0'R_0'$. Sigue que:

$$\triangle R_3 = R_3 \oplus R_3'$$

$$\begin{aligned} &= L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3) \oplus L'_0 \oplus f(R'_0, K_1) \oplus f(R'_2, K_3) \\ &= \triangle L_0 \oplus f(R_0, K_1) \oplus f(R'_0, K_1) \oplus f(R_2, K_3) \oplus f(R'_2, K_3) \end{aligned}$$

El ataque es de texto plano elegido, así que en particular podemos tomar las entradas de forma que $R_0 = R'_0$. Luego, al ser $\oplus$ nilpotente:

$$\triangle R_3 \quad = \quad \triangle L_0 \oplus f(R_2, K_3) \oplus f(R'_2, K_3),$$

luego

$$\triangle R_3 \oplus \triangle L_0 \quad = \quad f(R_2, K_3) \oplus f(R'_2, K_3)$$

(lo último se obtiene operando con $\triangle L_0 = (\triangle L_0)^{-1}$ de ambos lados). Recordemos que en DES, la estructura de f es la que se muestra en la Figura 3, donde P es nuevamente una permutación (recordar que el operador de permutación conmuta con $\oplus$). Dado que la $f(A, J)$ de la figura es $f(R_2, K_3)$ (pues nos referimos al último paso), se tiene que

$$\triangle R_3 \oplus \triangle L_0 = f(R_2, K_3) \oplus f(R'_2, K_3) = P(C) \oplus P(C') = P(C \oplus C')$$

de donde se deduce que:

$$\triangle C \quad = \quad P^{-1}(\triangle R_3 \oplus \triangle L_0).$$

Luego $\triangle C_i,\ i = 1, \ldots, 8$ son conocidos (dado que $\triangle C$ es la concatenación de los $\triangle C_i$). Veamos que $\triangle B_i,\ i = 1, \ldots, 8$ también son conocidos (mediante mostrar que $\triangle B$ es conocido). En la iteración final de la Figura 3, se tiene $A = R_2$ y $J = K_3$, de modo que

$$E(R_2) \oplus K_3 = B \quad y \quad E(R'_2) \oplus K_3 = B'$$

luego

$$\triangle B = E(R_2) \oplus E(R'_2).$$

Es decir, $\triangle B$ no depende de la clave.

Como $L_3 = R_2,\ L'_3 = R'_2$, se tiene que:

$$\triangle B = E(L_3) \oplus E(L'_3)$$

Luego los $\triangle B_i,\ i = 1, \ldots, 8$ se pueden obtener a partir de la información que poseemos. Con esto podemos montar un ataque de criptoanálisis diferenciable contra cada una de las 8 cajas-S (ya que se comportan como se especificó en la Figura 1). Con ello obtenemos 6 bits de la clave K_3 por cada caja, en total 48 bits (no tenemos más que concatenar los 8 grupos de 6 bits obtenidos). Los restantes 8 para llegar a K se obtienen por fuerza bruta sin mayor problema, tal como se explicó.

Por último, es interesante notar que dentro de cada caja-S el conjunto de textos planos y claves posibles es completamente manejable (4 y 6 bits, respectivamente). No habría el menor problema en almacenar todas las $2^6 = 64$ claves distintas, ni en recorrer el conjunto exhaustivamente para calcular $K(X, \delta, \tau)$. Incluso si probáramos todos los textos posibles ($2^4 = 16$), no necesitaríamos más de $2^{10} = 1024$ iteraciones en total. Sumando sobre las 8 cajas y considerando las 2^8 alternativas a probar al final, tenemos un total de $2^{13} + 2^8 = 8448$ operaciones "rápidas" (encripciones a lo sumo), lo cual es un número despreciable.

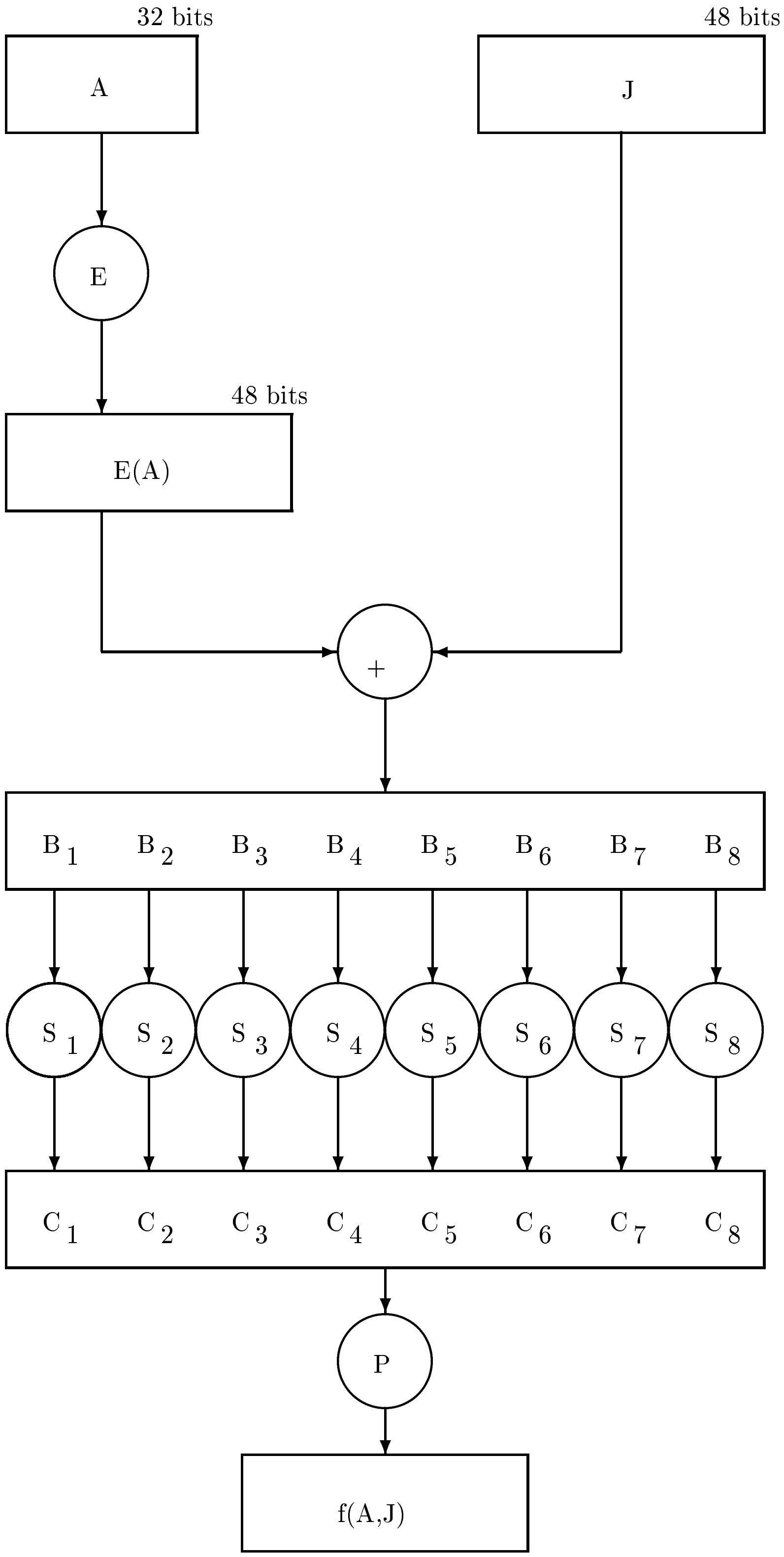

Figura 3: Estructura de f en el criptosistema DES.

2.4 Criptoanálisis Diferenciable y DES

El criptoanálisis diferenciable funciona contra DES en cualquiera de sus modos de operación, ECB, CBC, CFB y OFB con la misma complejidad computacional, es decir, ninguno de estos modos presenta ventajas respecto a los otros en cuanto a resistencia contra este ataque.

En la siguiente tabla presentamos un resumen de los mejores ataques diferenciales contra DES con distintos números de pasos [5] . La primera columna es el número de pasos de la implementación. Las siguientes dos son el número de textos elegidos o textos conocidos que deben ser examinados para efectuar el ataque (el criptoanálisis diferenciable puede adaptarse a una forma con texto plano conocido en vez de elegido de manera relativamente simple). La cuarta columna es el número de estos textos que fueron analizados realmente en estos ataques más exitosos.

Número de Pasos	Textos Elegidos	Textos Conocidos	Textos Analizados
8	2^{14}	2^{38}	4
9	2^{24}	2^{44}	2
10	2^{24}	2^{43}	2^{14}
11	2^{31}	2^{47}	2
12	2^{31}	2^{47}	2^{21}
13	2^{39}	2^{52}	2
14	2^{39}	2^{51}	2^{29}
15	2^{47}	2^{56}	2^{7}
16	2^{47}	2^{55}	2^{36}

La resistencia de DES se puede mejorar aumentando el número de pasos, como se puede apreciar en la tabla. Un ataque de criptoanálisis diferenciable de texto elegido contra un DES con 17 o 18 pasos toma aproximadamente el mismo tiempo que un ataque de fuerza bruta. [6] Con 19 pasos o más, el criptoanálisis diferenciable se hace imposible porque requiere más textos planos elegidos que los que se pueden generar con el tamaño de bloque que utiliza DES (más de 2^{64} textos, donde los textos son de 64 bits).

Este ataque depende mucho de la estructura de las cajas-S. Estas están optimizadas contra el criptoanálisis diferenciable, es decir, están construídas de manera de hacer que el ataque sea lo más difícil posible.

¿Por qué DES es tan resistente al criptoanálisis diferenciable? ¿Por qué están optimizadas las cajas-S para hacer que este ataque sea lo más difícil que sea posible? ¿Por qué tiene la cantidad justa de pasos para que sea ineficiente, pero ni uno más? La respuesta es porque los diseñadores conocían este ataque. Don Coppersmith, de la IBM, escribió: [7]

> El diseño aprovechaba ciertas técnicas de criptoanálisis, principalmente la técnica del "criptoanálisis diferenciable", que no eran conocidas en la literatura publicada. Tras discusiones con NSA, se decidió que entregar las consideraciones del diseño revelaría

[5]E. Biham y A. Shamir, *"Differential Cryptanalysis of the Data Encryption Standard"*, Springer-Verlag, 1993.

[6]B. Schneier, *"Applied Cryptography", Wiley, 1996, p. 289.*

[7]D. Coppersmith, 1.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, Technical Report RC 18613, IBM T.J. Watson Center, Dic. 1992; 2.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, IBM Journal of Research and Development, v. 38, n. 3, Mayo 1994, pp. 243-250

la técnica del criptoanálisis diferenciable, una técnica poderosa que puede ser utilizada contra varios criptosistemas. Esto a su vez debilitaría la ventaja competitiva que los Estados Unidos disfrutaba sobre el resto de los países en materia de criptografía.

Adi Shamir desafió a Coppersmith a decir que no había encontrado ningún ataque más fuerte contra DES desde entonces, pero Coppersmith no ha respondido. Sabiendo que la NSA conocía el criptoanálisis diferenciable más de 15 años antes que el mundo académico, existe cierta intriga en cuanto a qué métodos habrán desarrollado en esos 15 años de ventaja, durante los cuales ciertamente sus investigadores no deben haber estado descansando precisamente.

2.5 Ideas Alternativas

Existen otros métodos con ciertas similitudes al criptoanálisis diferenciable que se han utilizado también contra DES, con mayor o menor éxito. Entre ellas tenemos el Criptoanálisis de Clave Relacionada (Related Key Cryptanalysis) y el Criptoanálisis Lineal.

Criptoanálisis de Clave Relacionada

La clave de DES se rota cierta cantidad de bits en cada paso: 2 tras cada paso salvo los pasos 1,2,9 y 16, en los cuales rota uno sólo. ¿Por qué?

El criptoanálisis de clave relacionada es similar al criptoanálisis diferencial, pero examina las diferencias entre claves. El ataque en sí es muy distinto: El criptoanalista toma deliberadamente cierta relación entre dos claves, pero no conoce las claves mismas. Se encriptan datos con ambas claves. En la versión de texto plano conocido, se dispone del texto plano y el texto cifrado con ambas claves. En la versión con texto plano elegido, el criptoanalista elige el texto plano a encriptar con ambas claves (pero no las claves).

Un DES modificado, en el cual la clave se rota exactamente dos bits en cada paso, es menos seguro. Utilizando criptoanálisis de clave relacionada se puede quebrar esta variante usando 2^{17} textos elegidos de clave elegida o 2^{33} textos conocidos de clave elegida. [8].

Este ataque es bastante impráctico, pero es interesante por tres motivos:

1. es el primer ataque criptoanalítico contra el sistema de generación de subclaves de DES.
2. este ataque es independiente del número de pasos del criptosistema, es igualmente efectivo contra DES con 16, 32 o 1000 pasos.
3. DES es inmune a este ataque. La variación en el número de bits que se rotan frustra el ataque de clave relacionada.

[8] E. Biham, 1.- *"New Types of Cryptanalitic Attacks Using Related Keys"*, Technical Report #753, Computer Science Department, Technion–Israel Institute of Technology, Sep. 1992.; 2.- *"New Types of Cryptanalitic Attacks Using Related Keys"*, Journal of Cryptology, v. 7, n. 4, 1994, pp. 229-246.

Criptoanálisis Lineal

El Criptoanálisis Lineal es otro tipo de ataque, inventado por Mitsuru Matsui. [9] Este ataque usa aproximaciones lineales para describir la acción de un criptosistema de bloque (en este caso DES). Esto significa que si tomamos el "o exclusivo" (XOR) de algunos de los bits del texto plano, el XOR de algunos del texto cifrado, y tomamos el XOR de ambas cosas, obtendremos un bit que sería el XOR de algunos bits de la clave. Esto es una aproximación lineal, y coincidirá con la realidad con cierta probabilidad p. Si $p \neq \frac{1}{2}$, este sesgo puede ser aprovechado. Utilizando textos planos y sus textos cifrados se puede ir adivinando los valores de los bits de la clave. Mientras más datos se tiene, más confiable es la estimación. Mientras mayor es el sesgo (i.e. mientras mayor es $\mid p - \frac{1}{2} \mid$) mayor es la tasa de éxitos con la misma cantidad de datos conocidos.

Si las aproximaciones lineales fuesen insesgadas, entonces resultarían en 32 de 64 entradas posibles a cada caja-S. Este ataque está dirigido a dicha parte del algoritmo DES. Si hay una caja con sesgo suficientemente grande, el ataque puede funcionar. La caja-S más sesgada es S_5. De hecho, el segundo bit de entrada es igual al XOR de los cuatro de la salida sólo en 12 casos. Esto se traduce en una probabilidad de $\frac{3}{16}$, i.e. un sesgo de $\frac{5}{16}$, y es el sesgo más extremo de todas las cajas-S. (Según el libro de Schneier, Shamir notó este sesgo, pero no pudo encontrar una manera de explotarlo).

Contra un DES de 16 pasos, este ataque puede encontrar la clave con un promedio de 2^{43} textos conocidos. Una implementación en software de este ataque quebró una clave DES en 50 días usando 12 estaciones HP9735 [10] . Es el ataque más efectivo contra DES que se conoce a la fecha.

El criptoanálisis lineal depende fuertemente de la estructura de las cajas-S, y las cajas-S del DES no están optimizadas contra este ataque. De hecho, el ordenamiento de las cajas-S elegido para DES está entre el 9% y el 16% que ofrecen menor protección contra este ataque. [11] De acuerdo a Don Coppersmith [12], la resistencia contra el criptoanálisis lineal "no fué parte del criterio de diseño del DES". O bien no conocían el criptoanálisis lineal, o conocían otro ataque mucho más poderoso, haciendo que la resistencia contra éste tuviera prioridad sobre la resistencia contra el criptoanálisis lineal.

El criptoanálisis lineal es más reciente que el diferencial, y pueden haber mayores mejoras a su rendimiento en los años venideros. Algunas ideas se han propuesto, [13] pero no es claro que puedan ser utilizadas efectivamente contra un DES completo (con sus 16 pasos). Sin embargo, funcionan bastante bien contra variantes de menos pasos.

---

[9] M. Matsui, *"Linear Cryptanalysis Method for DES Cypher"*, Advances in Cryprology–EUROCRYPT '93 Proceedings, Springer-Verlag, 1994, pp. 386-397.

[10] M. Matsui, *"The First Experimental Cryptanalysis of the Data Encryption Standard"*, Advances in Cryptology–CRYPTO '94 Proceedings, Springer-Verlag, 1994, pp. 1-11.

[11] B. Schneier, *"Applied Cryptography"*, Segunda Edición, Wiley, 1996, p. 293.

[12] D. Coppersmith, 1.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, Technical Report RC 18613, IBM T.J. Watson Center, Dic. 1992; 2.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, IBM Journal of Research and Development, v. 38, n. 3, Mayo 1994, pp. 243-250

[13] Ver referencias [1270],[811] del libro de Schneier.

2.6 Direcciones a Explorar

Se han hecho algunos trabajos intentando extender el concepto de criptoanálisis diferenciable a derivadas de orden superior [14] Lars Knudsen usa "derivadas parciales" para atacar DES de 6 pasos; requiere 32 textos elegidos y 20.000 encripciones. [15] Todavía es muy pronto para saber si estas extensiones facilitarán quebrar un DES de 16 pasos. Otra vía de ataque es el Criptoanálisis Lineal-Diferenciable, que combina ambas técnicas antes presentadas. Susan Langford y Hellman tienen un ataque contra 8-DES que descubre 10 bits de la clave con una probabilidad de éxito de un 80% con 512 textos planos elegidos y un 95% de probabilidad de éxito con 768 textos planos elegidos, [16]. Tras el ataque se requiere una búsqueda de fuerza bruta de los bits restantes (2^{46} claves posibles). Mientras que este ataque consume un tiempo similar a ataques previos, requiere mucho menos texto plano. Lamentablemente parece que no se extiende con facilidad a DES con más pasos. Sin embargo, este ataque es nuevo y el trabajo sigue.

2.7 El Criterio de Diseño de las Cajas-S

Tras la salida a la luz pública del criptoanálisis diferenciable, IBM publicó el verdadero criterio de diseño de las cajas-S. [17] El criterio para las cajas S es:

- Cada caja-S tiene 6 bits de entrada y 4 de salida (pues era el mayor tamaño que se podía implementar en un sólo chip con la tecnología de 1974).
- Ningún bit de salida de una caja-S podría estar demasiado cerca de ser una función lineal de los bits de entrada.
- Si se fijan el bit de más a la izquierda y el de más a la derecha de la entrada de una caja-S, y se varía el resto, cada salida posible de 4 bits se obtiene exactamente una vez.
- Si dos entradas a una caja-S difieren exactamente en un bit, las salidas deben ser distintas en al menos dos bits.
- Si dos entradas a una caja-S difieren en sus primeros dos bits y son idénticas en los últimos dos, las dos salidas no pueden ser iguales.
- Para cualquier diferencia no nula de 6 bits entre las entradas, no más de 8 de los 32 pares de entradas que tienen esa misma diferencia pueden dar la misma diferencia en la salida.
- Un criterio similar al anterior, pero para el caso de tres cajas-S activas (i.e. para cualquier par de diferencias no nulas, no más de cierta cantidad de los tripletes de entradas puede dar el mismo par de diferencias de salida).

---

[14] Ver referencias [702,161,927,858,860] del libro de Schneier.

[15] L.R. Knudsen, *"Applications of Higher Order Differentials and Partial Differentials"*, K.U. Leuven Workshop on Cryptographic Algorithms, Springer-Verlag, 1995.

[16] S.K. Langford y M.E. Hellman, *"Cryptanalysis of DES"* presentado en la conferencia 1994 RSA Data Security, Redwood Shores, California, 12-14 de Enero 1994

[17] D. Coppersmith, 1.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, Technical Report RC 18613, IBM T.J. Watson Center, Dic. 1992; 2.- *"The Data Encryption Standard (DES) and Its Strength against Attacks"*, IBM Journal of Research and Development, v. 38, n. 3, Mayo 1994, pp. 243-250

Claramente el sexto criterio apunta a defender las cajas-S contra el criptoanálisis diferenciable, agrandando los conjuntos de claves posibles para cada par de diferencias de entrada y salida.

El artículo explica estos criterios y sus razones de ser. Generar cajas-S es sumamente fácil hoy en día, pero era una tarea complicada en aquella época. Se dice que tuvieron computadores corriendo durante meses para fabricar las famosas cajas-S.