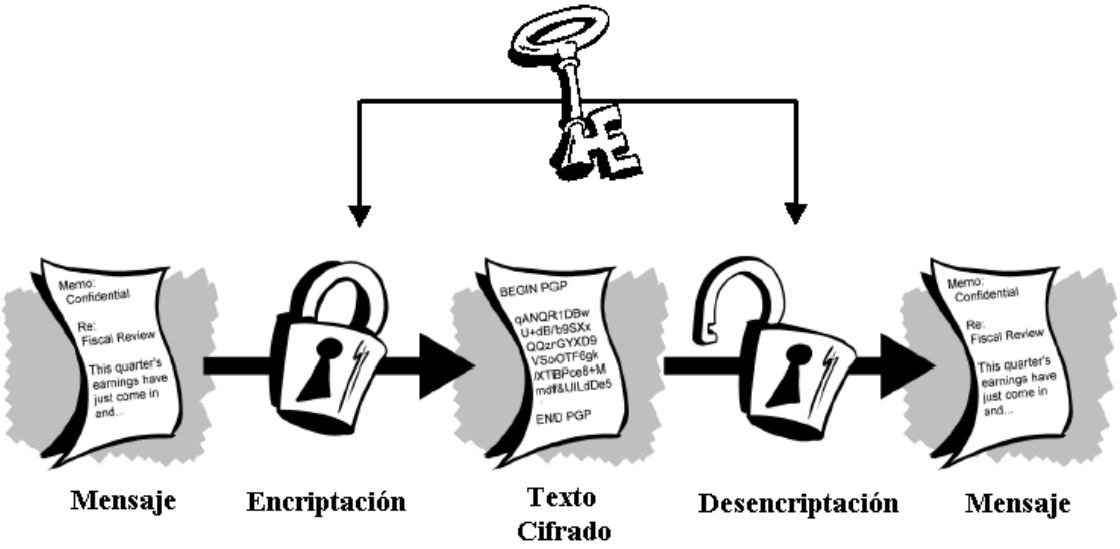
C A P Í T U L O III

CRIPTOGRAFÍA DE CLAVE PRIVADA



3.1 DEFINICIÓN

Como se mencionó en el capítulo anterior, *los cifrados simétricos (clave privada)* requieren que tanto el emisor como el receptor tengan la misma clave. Esta clave es usada por el emisor para cifrar los datos, y de nuevo por el receptor para descifrar los datos *(ver figura 3.1)*.

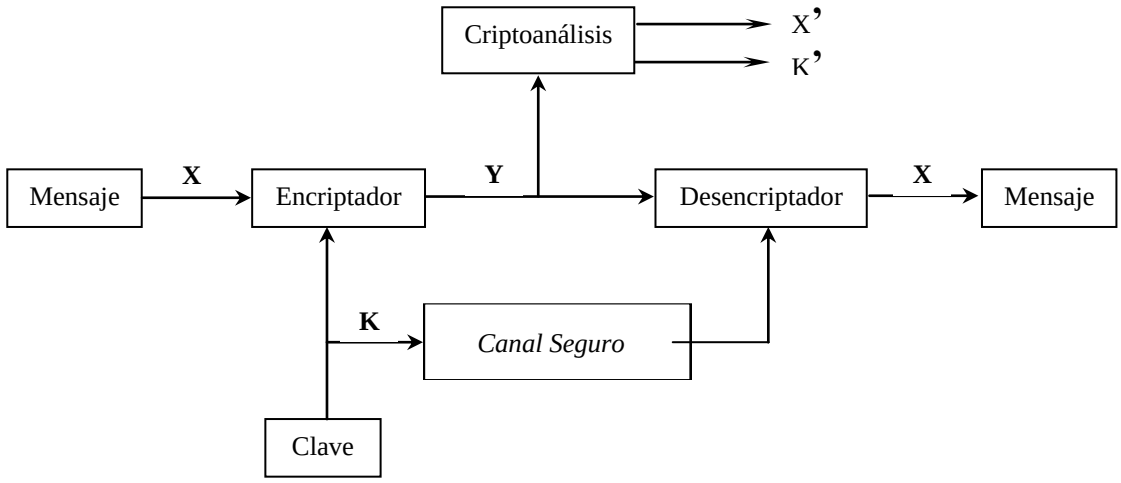


Figura 3.1 Modelo de Criptosistema Simétrico [LIB 002]

Con el mensaje *X* y la clave de encriptado *K* como entradas, el algoritmo de encriptado o encriptador, genera el texto cifrado *Y*, es decir, $Y=E_K(X)$. El receptor, en posesión de la clave *K*, es capaz de invertir la transformación: $X=D_K(Y)$. Un oponente, observando *Y* pero sin tener acceso a *K* o *X*, debe intentar recuperar *X* o *K* o ambos. Si el oponente está interesado sólo en este mensaje en particular, entonces el foco de esfuerzo está en recuperar *X* por medio de una estimación del texto nativo, *X'*. Si el oponente está interesado en leer mensajes futuros, se hace un intento de recuperar *K* por medio de una estimación de la clave, *K'*.

El modelo de cifrado simétrico se divide en 2 categorías: cifrado por sustitución y cifrado por transposición.

3.2 CIFRADO POR SUSTITUCIÓN

En el cifrado por sustitución, cada letra o grupo de letras se reemplaza por otra letra o grupo de letras para disfrazarla. Uno de los cifrados más viejos es el cifrado de César, expuesto en el capítulo anterior, y consiste en desplazar todas las letras del alfabeto en una cantidad conocida. Si el desplazamiento es, digamos cinco, eso significa que la letra A se transforma en la F (la letra que hay cinco posiciones a la derecha de la A en el alfabeto), la B en la G, la C en la H, y así sucesivamente. Es decir, convertimos cada letra de texto llano (primera fila) en una letra de texto cifrado (segunda fila) de acuerdo con la siguiente secuencia:

ABCDEFGHIJKLMNOPQRSTUVXYZ *(texto llano)*
FGHIJKLMNOPQRSTUVXYZABCDE *(texto cifrado)*

donde hemos supuesto que tras la Z viene la A. De este modo, JULIOCESAR se convierte en OAQNTHJYFX. Podemos denotar la clave como cinco (número de posiciones a desplazar); el algoritmo sería simplemente la sustitución de una letra por la que hay x posiciones a la derecha (x es la clave). Para descifrar, no hay más que tomar el texto cifrado y sustituir cada letra por la que hay cinco posiciones a su izquierda (la O se convierte en la J, la A en la U...). Si usamos dígitos binarios en lugar de letras, podremos utilizar el sistema de César para cifrar todo tipo de archivos informáticos (archivos de texto, audio, video, imagen e incluso archivos ejecutables).

El sistema Cesariano ya no se utiliza debido a la sencillez con la que se puede romper, o más correctamente, *criptoanalizar,* que es la ciencia de obtener el texto llano a partir del texto cifrado sin conocer la clave, o bien de obtener la propia clave a través de texto llano o cifrado.

El método de cifrado de César no es seguro porque resulta fácil presa del criptoanálisis. Para empezar, es susceptible de un *ataque de fuerza bruta*, consistente simplemente en probar todas las posibles claves. Igual que si probásemos todas las combinaciones de una caja fuerte. Utilizando el alfabeto apuntado anteriormente, solamente tenemos veinticinco posibles claves (naturalmente, desplazar una letra 26 posiciones da igual resultado que desplazarla uno solo), lo que nos permite probar todas las combinaciones de forma cómoda.

3.3 CIFRADO POR TRANSPOSICIÓN

Los cifrados por transposición reordenan las letras del texto normal pero no las disfrazan como en el cifrado por sustitución. A continuación se presenta el cifrado por transposición columnar, en donde la clave del cifrado es una palabra o frase que no contiene letras repetidas. El propósito de esta clave es numerar las columnas, estando la primera columna bajo la letra clave más cercana al inicio del alfabeto, y así sucesivamente. El texto normal se escribe horizontalmente en filas. El texto cifrado se lee por columnas, comenzando por la columna cuya letra clave es la más baja.

Por ejemplo, si nuestra clave es la palabra '*encripta'* y el texto normal es: '*por favor enviar documentación hasta el día lunes',* el texto cifrado sería algo como:

'rocanfrvmndapecieeaanaaconuólsvrtsldfiehíbodatur'

E	N	C	R	I	P	T	A	⟶ *clave*
3	5	2	7	4	6	8	1	⟶ *orden de la letra clave*
p	o	r	f	a	v	o	r	
e	n	v	i	a	r	d	o	
c	u	m	e	n	t	a	c	
i	ó	n	h	a	s	t	a	
e	l	d	í	a	l	u	n	
e	s	a	b	c	d	e	f	

3.4 DES (DATA ENCRYPTION STANDARD)

En Enero de 1977, el Departamento de Comercio y la Oficina Nacional de Estándares de Estados Unidos publicaron la norma **DES**, un esquema de cifrado de claves privadas. El algoritmo DES es un sistema monoalfabético que fue desarrollado en colaboración con IBM y se presentó al público con la intención de proporcionar un algoritmo de cifrado normalizado para redes de ordenadores. DES se basa en el desarrollo de un algoritmo de cifrado que modifica el texto con tantas combinaciones que el criptoanalista no podría deducir el texto original aunque dispusiera numerosas copias.

En la *figura 3.2* se muestra el esquema general del algoritmo DES. El algoritmo se parametriza con una clave de 56 bits (dadas en 8 bytes, en cada uno de los cuales 7 bits son de la llave y el octavo es de *paridad*). El texto normal se cifra en bloques de 64 bits, produciendo en la salida 64 bits de texto cifrado. El cifrado comienza con la función de *permutación inicial*, que es independiente de la clave, la salida está formada por los mismos bits cambiados de orden.

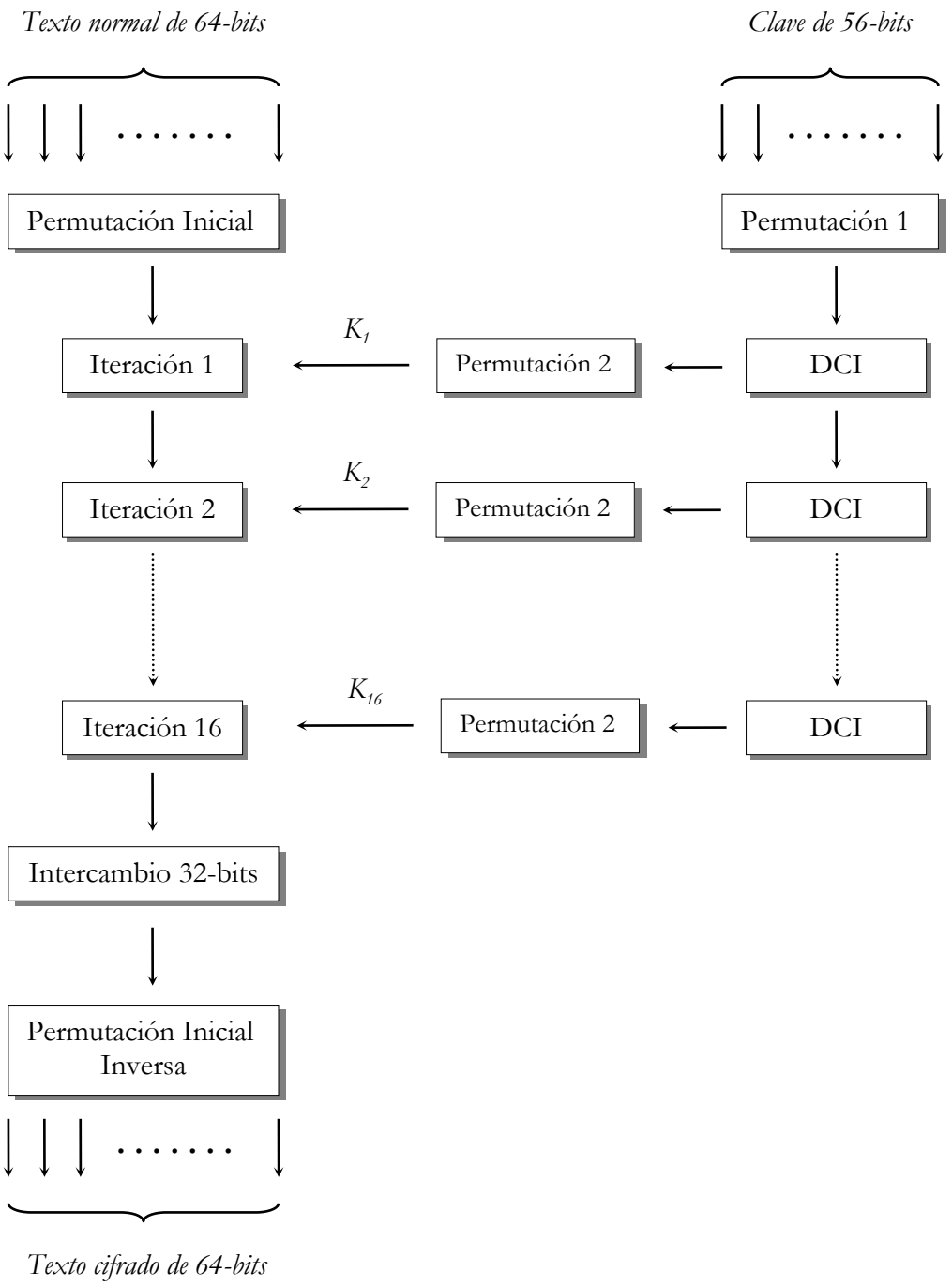


Figura 3.2 Esquema general del algoritmo DES [LIB 002]

Las 16 etapas siguientes denominadas *iteraciones* son funcionalmente idénticas, pero cada una se parametriza con una parte diferente de la clave. La penúltima función "*Intercambio 32-bits"* como su nombre lo indica, intercambia los 32 bits de la izquierda y los 32 bits de la derecha. La última función es el inverso exacto de la permutación inicial.

La función de permutación inicial con un bloque de entrada de 64 bits, se permuta de la siguiente manera:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Es decir, la entrada permutada tiene como primer bit el bit 58 del original, como segundo bit el bit 50 del original, y así sucesivamente, hasta llegar al último bit, que corresponderá al bit 7 del texto sin cifrar. La función de permutación inicial inversa se reordena de la siguiente manera:

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Es decir, la entrada permutada inversa tiene como primer bit el bit 40 de la salida de la función intercambio 32-bits, como segundo bit el bit 8 de la salida de la función intercambio 32-bits, y así sucesivamente, hasta llegar al último bit, que corresponderá al bit 25 de la salida de la función intercambio 32-bits.

Los 56 bits de la clave son usados de la siguiente manera: Inicialmente la clave se reordena por medio de una función de permutación. A continuación, para cada una de las iteraciones, se genera una subclave K_i por medio de la función DCI (desplazamiento circular a la izquierda) y otra función de permutación que es la misma para cada iteración. Note que se produce una subclave diferente para cada iteración debido al desplazamiento repetido de los bits de la clave suministrada.

En la *figura 3.3* se muestra el procesamiento para una iteración del algoritmo DES, el cual se puede resumir en las siguientes fórmulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \ominus f(R_{i-1}, K_i)$$

donde $\ominus$ denota la función XOR bit a bit.

La parte izquierda L_i de la salida de una iteración es igual a la parte derecha R_{i-1} de la entrada a esa iteración. La parte derecha de la salida R_i es la operación XOR de L_{i-1} y una función compleja de R_{i-1} y K_i.

La *"función compleja"* lleva a cabo cuatro pasos sobre la salida derecha, mediante una transposición basada en la operación OR-Exclusivo.

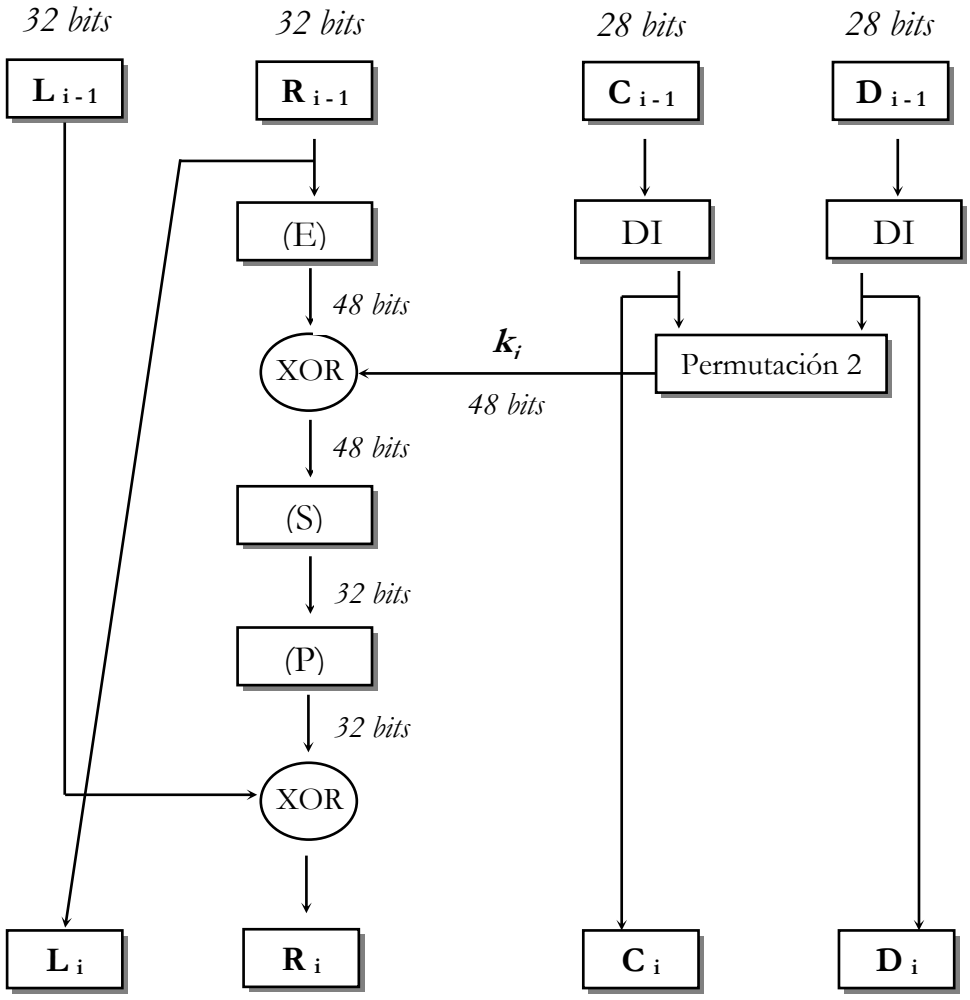


Figura 3.3 Iteración del Algoritmo DES [LIB 002]

1. La mitad derecha R_{i-1} de 32 bits se convierte, mediante una regla de expansión y permutación, en el número E, de 48 bits, los mismos que se reordenan de la siguiente manera:

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

2. *E* y *K* se combinan mediante un OR-Exclusivo.
3. Los 48 bits generados en la etapa 2 se dividen en ocho grupos de 6 bits que se introducen en las cajas "*S"*, cada una de las cuales produce 4 bits de salida, es decir, simplemente traducen cada combinación de entrada de 48 bits en un modelo particular de 32 bits.
4. Los 32 bits restantes se introducen en la caja de permutación *P* que consiste en el siguiente reordenamiento:

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

La clave de 56 bits se trata como dos cantidades de 28 bits, rotuladas como C_{i-1} y D_{i-1}. En cada iteración estas cantidades sufren de forma separada un desplazamiento circular a la izquierda (DI), los cuales sirven como entrada para la siguiente iteración y también como entrada a otra función de permutación que produce una salida de 48 bits que sirve como entrada a la función $f(R_{i-1}, K_i)$. Los desplazamientos a la izquierda son de dos bits, salvo para las rondas 1, 2, 9 y 16, en las que se desplaza sólo un bit.

El proceso de descifrado del DES es básicamente el mismo que el proceso de cifrado con una pequeña variación. La idea es, usar el texto cifrado como entrada al algoritmo DES, pero usar la clave en orden inverso, es decir, utilizar la subclave K_{16} en la primera iteración, la subclave K_{15} en la segunda iteración y así sucesivamente hasta utilizar la subclave K_1 en la última iteración.

Este algoritmo ha sido motivo de gran controversia, parte de ella se debe al secreto que rodeó a su desarrollo. IBM trabajó en colaboración con la Agencia Nacional de Seguridad de Estados Unidos y ambas guardaron el secreto de los aspectos del diseño del algoritmo. Muchas de las críticas obtenidas se encuentran en el hecho de usar sólo 56 bits de parte de la clave para conseguir el cifrado, esto ya es considerado por muchos insuficientes.

Evidentemente para romper una clave semejante sería necesaria una enorme cantidad de potencia de cálculo. Sin embargo, no es una tarea imposible. Los ordenadores de alta velocidad, mediante análisis estadísticos, no necesitan emplear todas las posibles combinaciones para romper la clave. A pesar de ello, el objetivo de DES no es proporcionar una seguridad absoluta, sino únicamente un nivel de seguridad razonable para las redes orientadas a aplicaciones comerciales.

Una variante al DES es usar el Triple DES. Consiste en aplicar varias veces el algoritmo DES con diferentes claves al mensaje original. Responde a la siguiente estructura:

$$C = E_{k1} (D_{k2} (E_{k1} (M)))$$

Es decir, codificamos con la subclave k_1, decodificamos con k_2 y volvemos a codificar con k_1. La clave resultante es la concatenación de k_1 y k_2, con una longitud de 112 bits.

La razón de que se usan sólo dos claves en lugar de tres es que una clave de 112 bits es suficiente para las aplicaciones comerciales por ahora. Subir a 168 bits simplemente agregaría una carga extra innecesaria al proceso de cifrado y descifrado. La razón para cifrar, descifrar y luego cifrar de nuevo es la compatibilidad en reversa con los sistemas DES de una sola clave [LIB 001].

3.5 IDEA (International Data Encryption Algorithm)

IDEA fue desarrollado por Xuejia Lai y James Massey en 1990. Es un criptosistema simétrico que encripta los datos en bloques de 64 bits usando una clave de 128 bits, lo que lo hará inmune durante décadas a los ataques de fuerza bruta, lotería china, etc. También se diseñó para resistir el criptoanálisis diferencial. Hasta la fecha IDEA se considera un criptosistema altamente seguro y no hay ninguna técnica o máquina conocida que se crea pueda violar el IDEA. Este algoritmo está libre de restricciones y permisos nacionales y es de libre distribución por Internet, lo que ha hecho que sea un algoritmo muy popular, sobre todo fuera de los EE.UU.

La estructura básica del algoritmo es similar al DES en cuanto se alteran bloques de texto normal de 64 bits en una secuencia de 8 iteraciones y una transformación parametrizadas con subclaves para producir bloques de salida de texto cifrado de 64 bits, como se muestra en la *figura 3.4*.

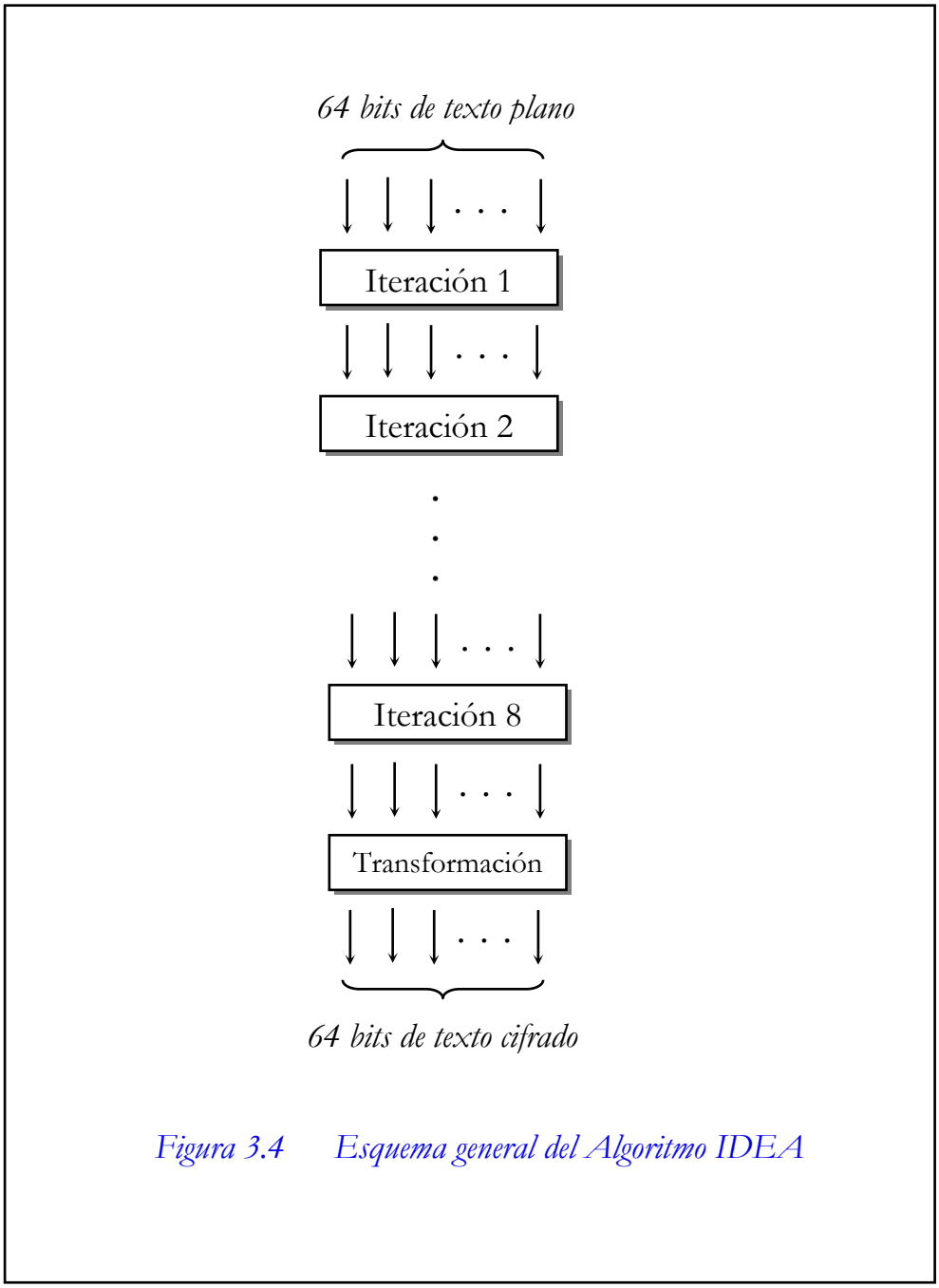


Figura 3.4 Esquema general del Algoritmo IDEA

Los detalles del algoritmo es como se muestra en la *figura 3.5*. Dividiremos el bloque de texto plano de 64 bits, en cuatro partes X_1, X_2, X_3 y X_4 de 16 bits cada una. Denominaremos Z_i a cada una de las 52 subclaves de 16 bits que

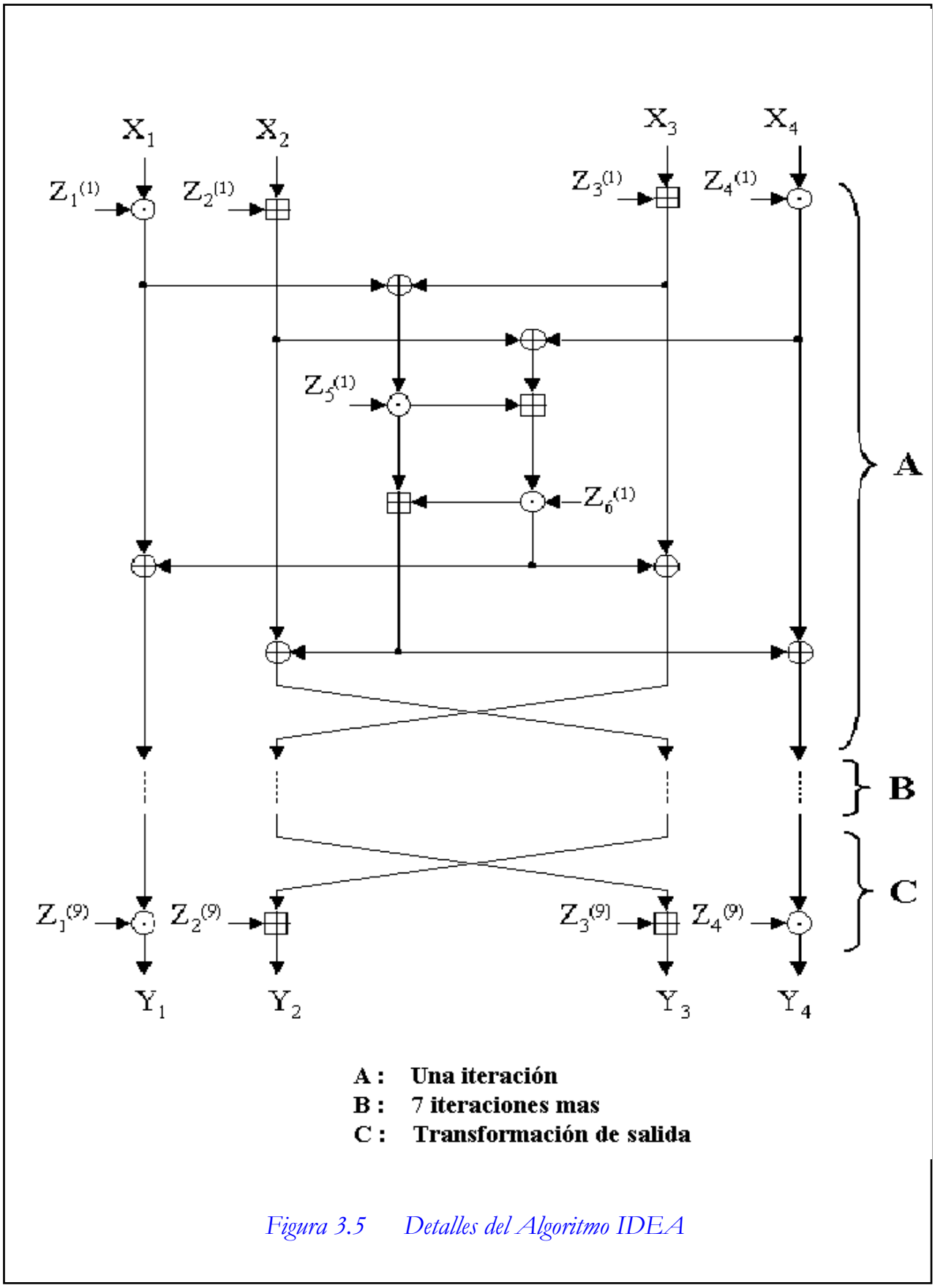
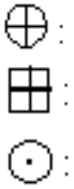


Figura 3.5 Detalles del Algoritmo IDEA

vamos a necesitar, 6 por cada una de las 8 iteraciones y 4 para la transformación final. Las operaciones elementales que usaremos en cada iteración son:



: Or Exclusivo sobre subbloques de 16 bits.

: Suma en módulo 2^{16} de enteros de 16 bits.

: Multiplicación en módulo $2^{16} + 1$ de enteros de 16 bits.

Las operaciones que llevaremos a cabo en cada iteración son las siguientes:

1. Multiplicar X_1 por Z_1.
2. Sumar X_2 con Z_2.
3. Sumar X_3 con Z_3.
4. Multiplicar X_4 por Z_4.
5. Hacer un XOR entre los resultados del paso 1 y el paso 3.
6. Hacer un XOR entre los resultados del paso 2 y el paso 4.
7. Multiplicar el resultado del paso 5 por Z_5.
8. Sumar los resultados de los pasos 6 y 7.
9. Multiplicar el resultado del paso 8 por Z_6.
10. Sumar los resultados de los pasos 7 y 9.
11. Hacer un XOR entre los resultados de los pasos 1 y 9.
12. Hacer un XOR entre los resultados de los pasos 3 y 9.
13. Hacer un XOR entre los resultados de los pasos 2 y 10.
14. Hacer un XOR entre los resultados de los pasos 4 y 10.

La salida de cada iteración serán los cuatro subbloques obtenidos en los pasos 11, 12, 13 y 14, que serán la entrada del siguiente ciclo, en el que emplearemos las siguientes seis subclaves, hasta un total de 48. Después de la octava

iteración se lleva a cabo una transformación en la que se realizan las siguientes operaciones:

1. Multiplicar X_1 por Z_{49}.
2. Sumar X_2 con Z_{50}.
3. Sumar X_3 con Z_{51}.
4. Multiplicar X_4 por Z_{52}.

Como se puede observar, a pesar de usarse una clave de 128 bits, se utilizan en total 52 subbloques de clave de 16 bits cada uno. Para obtenerlos se sigue el siguiente proceso:

1. La clave inicial de 128 bits se divide en los 8 primeros subbloques de 16 bits, los 6 subbloques de la primera iteración y los 2 primeros subbloques de la segunda iteración.
2. Se efectúa una rotación en 25 bits a la izquierda sobre la clave inicial de 128 bits para obtener una nueva clave que será a su vez dividida en los 8 siguientes subbloques de 16 bits, cuatro para la segunda iteración y cuatro para la tercera.

Se realizan sucesivamente estas rotaciones hasta tener los 52 subbloques requeridos.

COMPARACIÓN DES vs IDEA

DES	IDEA
✓ El algoritmo se parametriza con una clave de 56 bits. ✓ El texto normal se cifra en bloques de 64 bits. ✓ Produce una salida de 64 bits de texto cifrado. ✓ No proporciona una seguridad absoluta. ✓ IBM trabajó en colaboración con la Agencia Nacional de Seguridad de Estados Unidos y ambas guardaron el secreto de los aspectos del diseño del algoritmo.	✓ El algoritmo se parametriza con una clave de 128 bits. ✓ El texto normal se cifra en bloques de 64 bits. ✓ Produce una salida de 64 bits de texto cifrado. ✓ Se considera un criptosistema altamente seguro. ✓ Libre de restricciones y permisos nacionales y es de libre distribución por Internet.

3.6 CRIPTOANÁLISIS DIFERENCIAL Y LINEAL

El criptoanálisis consiste en comprometer la seguridad de un criptosistema. Esto se puede hacer descifrando un mensaje sin conocer la llave, o bien

obteniendo a partir de uno o más criptogramas[15] la clave que ha sido empleada en su codificación. No se considera criptoanálisis el descubrimiento de un algoritmo secreto de cifrado; por el contrario hemos de suponer que los algoritmos siempre son conocidos.

En general el criptoanálisis se suele llevar a cabo estudiando grandes cantidades de pares de mensaje-criptograma generados con la misma clave. El mecanismo que se emplee para obtenerlos es indiferente, y puede ser resultado de escuchar un canal de comunicaciones. Obviamente, cuanto mayor sea la cantidad de pares, más probabilidades de éxito tendrá el criptoanálisis.

Cuando el sistema es débil, pueden ser suficientes unos cuantos mensajes para obtener información que permita deducir la clave empleada. También podemos tratar de criptoanalizar un sistema aplicando el algoritmo de descifrado, con todas y cada una de las claves, a un mensaje codificado que poseemos y comprobar cuales de las salidas que se obtienen tienen sentido como posible texto plano. Este método y todos los que buscan exhaustivamente por el espacio de claves *K*, se denominan ataques por la fuerza bruta, y en muchos casos no suelen considerarse como auténticas técnicas de criptoanálisis, reservándose este término para aquellos mecanismos que explotan posibles debilidades intrínsecas en el algoritmo de cifrado.

Hemos de tener en cuenta no obstante que la capacidad de cálculo de las computadoras crece a gran velocidad, por lo que algoritmos que hace unos años eran resistentes frente a ataques por la fuerza bruta hoy pueden resultar

[15] Mensaje Cifrado

inseguros, como es el caso del DES. Sin embargo, existen longitudes de clave para las que resultaría imposible a todas luces un ataque de este tipo.

Un par de métodos de criptoanálisis que han dado interesantes resultados son *el criptoanálisis diferencial y el criptoanálisis lineal.*

El primero de ellos, descubierto por Biham y Shamir en 1990, permite efectuar un ataque de texto plano escogido a DES que resulta más eficiente que la fuerza bruta. Se basa en el estudio de los pares de criptogramas que surgen cuando se codifican dos textos planos con diferencias particulares, analizando la evolución de dichas diferencias a lo largo de las rondas de DES. Para llevar a cabo un criptoanálisis diferencial se toman dos mensajes cualesquiera (incluso aleatorios) idénticos salvo en un número concreto de bits. Usando las diferencias entre los textos cifrados, se asignan probabilidades a las diferentes claves de cifrado. Conforme tenemos más y más pares, una de las claves aparece como la más probable. Esa será la clave buscada.

El criptoanálisis lineal, descubierto por Mitsuru Matsui, basa su funcionamiento en tomar algunos bits del texto plano y efectuar una operación XOR entre ellos, tomar algunos del texto cifrado y hacerles lo mismo, y finalmente hacer un XOR de los dos resultados anteriores, obteniendo un único bit. Efectuando esa operación a una gran cantidad de pares de texto plano-criptograma diferentes podemos ver si se obtienen más ceros o más unos. Existen combinaciones de bits que, bien escogidas, dan lugar a un sesgo[16] significativo en la medida anteriormente definida, es decir, que el número de ceros (o unos) es apreciablemente superior. Esta propiedad

[16] Hay más unos que ceros o viceversa.

nos va a permitir poder asignar mayor probabilidad a unas claves sobre otras y de esta forma descubrir la clave que buscamos.

Otro tipo de análisis, esta vez para los algoritmos asimétricos, consistirá en tratar de deducir la llave privada a partir de la pública. Suelen ser técnicas analíticas que básicamente intentan resolver los problemas de elevado coste computacional en los que se apoyan estos criptosistemas: factorización, logaritmos discretos, etc.

La Criptografía no sólo se emplea para proteger información, también se utiliza para permitir su autenticación, es decir, para identificar al autor de un mensaje e impedir que nadie suplante su personalidad. En estos casos surge un nuevo tipo de criptoanálisis que está encaminado únicamente a permitir que elementos falsos pasen por buenos. Puede que ni siquiera nos interese descifrar el mensaje original, sino simplemente poder sustituirlo por otro falso y que supere las pruebas de autentificación.

Como se puede apreciar, la gran variedad de sistemas criptográficos produce necesariamente gran variedad de técnicas de criptoanálisis, cada una de ellas adaptada a un algoritmo o familia de ellos. Con toda seguridad, cuando en el futuro aparezcan nuevos mecanismos de protección de la información, surgirán con ellos nuevos métodos de criptoanálisis. De hecho, la investigación en este campo es tan importante como el desarrollo de algoritmos criptográficos.