Instituto Politécnico Nacional
Centro de Investigación en Computación

Criptoanálisis para la modificación de los estándares DES y Triple DES

Tesis de Doctorado del alumno

Víctor Manuel Silva García

Directores:

Dr. Juan Luis Díaz de León Santiago

Dr. Cornelio Yáñez Márquez

México, D.F.
Abril, 2007

INSTITUTO POLITÉCNICO NACIONAL

SIP-14

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 18:30 horas del día 26 del mes de Marzo de 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"CRIPTOANÁLISIS PARA LA MODIFICACIÓN DE LOS ESTÁNDARES DES Y TRIPLE DES"

Presentada por el alumno:

SILVA	**GARCÍA**	**VÍCTOR MANUEL**
Apellido paterno	materno	nombre(s)

Con registro:

B	0	4	1	2	6	7

aspirante al grado de: ***DOCTOR EN CIENCIAS DE LA COMPUTACIÓN***

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

DR. ALEXANDRE GUELBOUKH KAHN

Primer vocal
(Director de Tesis)

Secretario

DR. OLEKSIY POGREBNYAK

Segundo vocal
(Director de Tesis)

DR. JUAN LUIS DÍAZ DE LEÓN SANTIAGO

Tercer vocal

DR. CORNELIO YÁÑEZ MÁRQUEZ

Suplente

DR. CARLOS RENTERÍA MÁRQUEZ

DR. JUAN CARLOS CHIMAL EGUÍA

EL PRESIDENTE DEL COLEGIO

DR. HUGO CÉSAR COYOTE ESTRADA

INSTITUTO POLITECNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México D.F. el día 28 del mes de marzo del año 2007, el que suscribe **Víctor Manuel Silva García** alumno del Programa de Doctorado en Ciencias de la Computación con número de registro B041267, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Juan Luis Díaz de León Santiago y cede los derechos del trabajo intitulado Criptoanálisis para la modificación de los estándares DES y Triple DES, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección vsilvag@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Nombre y firma

AGRADECIMIENTOS

A mi esposa e hijo:

Huberta y Víctor Manuel

A mis padres:

Beatriz + y Eleazar

A mis hermanas:

Beatriz y Josefina

RESUMEN

En este trabajo se propone un ataque al criptosistema Data Encryption Standard (DES), el cual es diferente a los desarrollados hasta este momento. La estrategia de este ataque se basa en el Teorema LR que fue descubierto y probado por el autor. Este ataque puede ejecutarse en una computadora personal Pentium IV comercial. Obviamente, se requiere de más información que un simple trozo de texto claro y su correspondiente texto cifrado, sin embargo, este requerimiento se reduce a un mínimo de 24 bits.

El Teorema LR descubre 16 puertas traseras de las cuales 2 son de importancia. Teniendo en cuenta este tipo de conocimiento se propone un ataque a Triple-DES, el cual puede ejecutarse usando el computador personal arriba mencionado. Se propone una solución sencilla a este tipo de ataques para conservar la complejidad de Triple-DES, que es 2^{112}.

ABSTRACT

In this work an attack strategy on the Data Encryption Standard (DES) that is different from the existing ones is developed. The attack strategy is based on a theorem proved by the author, called LR theorem. The attack can be done by means of a personal computer i.e. a Pentium IV based machine. Obviously, more information than a sample of plaintext and ciphertext is required. However, this additional requirement is reduced to a minimum of just 24 bits.

The LR theorem uncovers 16 trapdoors 2 of which are of importance. With this knowledge, an attack on Triple-DES can be performed using the aforementioned personal computer. A simple solution to this kind of attack is proposed, preserving the complexity of the Triple-DES, 2^{112}.

Keywords: Data Encryption Standard, Triple DES, LR theorem, algorithm, cryptanalysis, attack.

Tabla de simbología utilizada

mod.	Módulo
PI	Permutación Inicial, en un ciclo de cifrado DES
E	Función de expansión-permutación.
B^i	Cadena de entrada a las cajas en la i-ésima ronda.
C^i	Cadena de salida de las cajas.
f	Función de mezclado.
L_i	Cadena izquierda de 32 bits en el proceso de mezclado.
R_i	Cadena derecha de 32 bits en el proceso de mezclado.
K_i	Cadena de 48 bits que es la *i*-ésima llave programada.
S_j	La caja número *j*.
B^i_j	Cadena de 6 bits. *j*-ésima parte de la cadena B^i.
B'	Entrada xor a las cajas.
C^i.	Salida de las cajas en la i-ésima ronda, la cual es una cadena de 32 bits.
C^i_j	Cadena de 4 bits. *j*-ésima parte de la cadena C^i.
K	Conjunto de todas las llaves. Su tamaño es de 2^{56}.
$\mathbf{K}_i$	Conjunto de llaves de tamaño 2^{16} donde se encuentra K_i.
K*	Conjunto de llaves de tamaño 2^{24} donde se encuentra la llave *K*.
$\mathbf{K}^{*,1}$	Conjunto de llaves de tamaño 2^{24} donde se encuentra la llave K^1 del primer y tercer ciclo de Triple-DES.
$\mathbf{K}^{*,2}$	Conjunto de llaves de tamaño 2^{24} donde se encuentra la llave K^2 del segundo ciclo de Triple-DES.
PC-1	Primera permutación-reducción del algoritmo de las llaves programadas.
PC-2	Segunda permutación-reducción del algoritmo de las llaves programadas.
$e_K(X)$	Un ciclo de cifrado DES para un texto claro *X* y una llave *K*.

ÍNDICE

CAPÍTULO 1

INTRODUCCIÓN

Este trabajo de tesis trata sobre una nueva técnica de criptoanálisis al criptosistema DES, que permitirá ejecutar un ataque al criptosistema Triple-DES.

1.1 OBJETIVO

Antes de enunciar el objetivo, es preciso ubicar el trabajo en el contexto pertinente.

Desde la antigüedad ha sido una necesidad del ser humano proteger la información que considera valiosa; para lograr este fin, ha inventado un gran número de procedimientos (algoritmos), siendo algunos de ellos muy ingeniosos. En esta actividad intervienen tres personajes: el que envía la información, el que recibe el mensaje y el que desea averiguar el mensaje entre los dos primeros [6].

Un ejemplo para ocultar un mensaje (texto claro) podría ser: sustituir una letra por otra, digamos A por Z, B por Y y así sucesivamente; en general, a este tipo de procedimiento se le llama cifrado o encripción de información. La encripción del ejemplo anterior adolece de una falla: la periodicidad con que aparecen las letras del alfabeto en una redacción, y un posible ataque a esta manera de ocultar un mensaje utilizaría esta debilidad. A este último modo de actuar (ataques a criptosistemas) se le conoce como criptoanálisis y, en buena medida, lo que se abordará en este trabajo tiene que ver con el criptoanálisis.

En la actualidad el criptosistema Data Encryption Standard, DES, aunque ya no es la norma, aún se sigue utilizando en algunos medios públicos y privados [16], como por ejemplo el área legal (**FIPS PUB 46-3**); la última versión de DES, Triple-DES, se utiliza en las áreas financiera y pública [17]. Vale la pena mencionar que desde su nacimiento en 1977, el criptosistema DES fue muy criticado; es más, muchos consideraron que sus días estaban contados y sin embargo, no sale de la norma hasta julio de 1998 [16]. En relación con Triple-DES, se puede decir que éste sustituye a DES en 1988 (**FIPS PUB 46-3**).

Los ataques que hasta el momento se han realizado en contra del criptosistema DES, consideran dos aspectos importantes, a saber: se conoce una parte de texto claro y su correspondiente texto encriptado, y se usa tecnología del tipo de las máquinas paralelas con gran capacidad de almacenamiento [6,12]; más adelante se analizarán los problemas que tiene este tipo de criptoanálisis.

En este trabajo no se presentará un ataque con tecnología de una máquina paralela y de gran capacidad de almacenamiento, sino con una Pentium IV comercial. Para compensar esta desventaja, se requiere pedir un poco más de información que un simple trozo de texto claro y su correspondiente texto encriptado; sin embargo, el autor reducirá al mínimo este requerimiento: únicamente 24 bits.

En este trabajo de tesis se dará respuesta, entre otras, a la pregunta ¿Cómo impacta este resultado a Triple-DES, que es un criptosistema aún vigente, tanto en el sector público como en el privado [17]?.

El objetivo de este trabajo de tesis se enuncia a continuación:

Encontrar una debilidad del criptosistema DES que no haya sido reportada; para posteriormente realizar un ataque; además, solucionar la debilidad descubierta y averiguar de qué manera impactan estos resultados a Triple-DES.

1.2 MOTIVACIÓN

El algoritmo de encripción Data Encryption Standard, DES, ha sido uno de los algoritmos más estudiados en el mundo y aún sigue siendo objeto de estudio [21,22,24,25,27] y del [28] al [34]; de hecho, para mucha gente código secreto es sinónimo de DES [7]. Este algoritmo es revisado cada cuatro años para realizar mejoras cuando se detectan fallas o debilidades; lo anterior, con objeto de que se mantenga su fortaleza (complejidad) ante posibles ataques [6]. La empresa Electronic Frontier Foundation construyó una máquina de $220,000 dólares que rompe el cifrado de DES en minutos [7]; en realidad, se puede considerar que éste es el ataque más importante que se le haya hecho al criptosistema DES. En el capítulo "ESTADO DEL ARTE" se describirán varios de los ataques más relevantes; de hecho, prácticamente desde los inicios de DES se empezaron a desarrollar diseños de máquinas para efectuar los ataques denominados de "fuerza bruta" [14].

No obstante que, de acuerdo con el estándar **ANSI X9.17,** Triple-DES se estandarizó para aplicaciones financieras en 1985, éste sustituye a DES hasta 1998. Por otro lado, el popular criptosistema Advanced Encryption Standard AES fue aprobado como estándar el 26 de mayo de 2002 y, apareció por primera vez en enero de 1997 [12]. Actualmente coexisten Triple-DES y AES; esto se debe al hecho de que Triple-DES aún es considerado un criptosistema seguro [17].

En este trabajo se presentarán algunas fallas (puntos débiles del protocolo de cifrado) que rompen al Triple-DES; esta fallas se deducen a partir del Teorema LR, el cual se probará en el capítulo de "DESARROLLO". Se debe señalar que el citado teorema es una de las aportaciones de este trabajo. En este momento aún hay una gran discusión en relación a la fortaleza de Triple-DES; de hecho, la mayoría de los investigadores coincide en que no hay una prueba matemática en relación a la seguridad de Triple-DES ni de ningún otro criptosistema [17]. Sin embargo, algunos investigadores han probado, empíricamente, que Triple-DES sigue siendo un criptosistema seguro [17]. En la última referencia se menciona que son pocas las ventajas que ofrece AES en relación con Triple-DES.

En relación con las fallas de Triple-DES que serán presentadas en este trabajo, se menciona también que se expondrá una solución sencilla a estas fallas, y esto significa que se conservan los tiempos de ejecución, se utilizan los mismos elementos de DES y se mantiene la complejidad de 2^{112} para Triple-DES ante el posible ataque que se reporta por primera vez en este trabajo.

Para el autor de esta tesis el significado de "se rompe Triple-DES", es cuando se realiza un ataque con una máquina Pentium IV, a partir del conocimiento que tiene el intruso de una pequeña parte de los bits involucrados en el proceso de encripción; estos bits, seleccionados de manera particular, conducen al conocimiento de las dos llaves de 56 bits utilizadas en el proceso. Además, es preciso hacer notar que este proceso, al ejecutarse en el equipo mencionado, se realiza en minutos y en algunos casos, en segundos.

1.3 PLANTEAMIENTO DEL PROBLEMA

En esta parte se hará una descripción a muy grandes rasgos del algoritmo DES, con objeto de describir el problema que se analizará en capítulos posteriores. El proceso se inicia con un bloque de 64 bits de texto claro y una llave de 64 bits. El algoritmo consta de dos partes, a saber: en una de ellas se realiza el mezclado de los 64 bits de texto claro en 16 rondas y, en la otra se obtienen 16 llaves de 48 bits a partir de la llave de 64 bits. Estas llaves se denominan llaves ronda o, también, llaves programadas.

La parte de mezclado de los 64 bits de texto claro se ejecuta en 16 rondas, y cada una de ellas da como resultado una cadena de 64 bits, la cual se divide en dos bloques de 32 bits cada uno. A estos bloques se les conoce como: bloque izquierdo L_i y bloque derecho R_i para $1 \leq i \leq 16$. Ahora bien, en la parte que se obtienen las 16 llaves de 48 bits cada una, se inicia el procedimiento con una llave de 64 bits; de los cuales se desechan ocho y a estos bits se les denomina bits de paridad. Los 56 bits restantes se manipulan de acuerdo con reglas especificadas en la norma, para obtener las16 llaves $K_1, K_2, \ldots, K_{16}$. Para concluir este breve planteamiento, se dirá que cada una de las llaves programadas K_i con $1 \leq i \leq 16$, interviene en la ronda correspondiente del proceso de mezclado.

En algunos de los ataques a DES, tales como el diferencial, el lineal e incluyendo el del autor [6,12], se busca obtener una de las llaves programadas, para posteriormente descubrir cuál es la llave. Entonces, además de conocer una parte de texto claro y su correspondiente texto encriptado, mínimo 64 bits, la primera parte del problema a resolver es determinar cuántos y de qué bloque serán los bits que se requiere conocer, para que ello conduzca al descubrimiento de una llave programada. Es deseable que el número de bits sea el mínimo posible y que el tiempo utilizado para encontrar la llave sea razonable; digamos que no pase de 24 hrs. La segunda parte del problema será buscar una solución a la debilidad que tienen las llaves programadas; para tal fin, se propondrá una variación al algoritmo de DES, cuidando que en el nuevo algoritmo propuesto no se disminuya la rapidez de ejecución de DES y se conserve su complejidad inicial de 2^{56}.

Se plantea, además, analizar de qué modo influye el conocimiento de cómo romper un DES simple, en la factibilidad de un ataque exitoso a Triple-DES.

1.4 CONTRIBUCIÓN

Se propone un ataque a DES diferente a los ya existentes, los cuales son: el criptoanálisis diferencial, el lineal, balance tiempo-memoria y el de fuerza bruta [6,12]. El marco teórico de este ataque será un teorema, denominado Teorema LR, el cual es propuesto y demostrado por el autor. De hecho, el citado teorema pone al descubierto 16 puntos débiles del algoritmo de cifrado, aunque únicamente dos de ellos son importantes.

Esta debilidad conduce a proponer un ataque a Triple-DES, el cual se puede ejecutar en una Pentium IV comercial y se realiza en minutos. Otro aspecto que se muestra en este trabajo son dos soluciones para resolver la debilidad expuesta por el Teorema LR, una de ellas para el DES simple y otra para el Triple-DES. En ambas soluciones se utilizan los mismos elementos del algoritmo DES y se aplican en la generación de las llaves programadas. Como consecuencia, se proponen dos algoritmos diferentes a los establecidos por la norma. Resumiendo: la contribución del autor es el Teorema LR, el ataque a Triple-DES y dos soluciones que resuelven las debilidades generadas por el Teorema LR.

1.5 DESCRIPCIÓN DE LA OBRA

El trabajo consta de 5 capítulos además del actual. En el capítulo 2, "ANTECEDENTES", se hace una descripción detallada del algoritmo de DES por ser materia de estudio en este trabajo. Además, se menciona a muy grandes rasgos los inicios de la "Criptografía". De hecho, se muestra un ejemplo muy famoso de los tiempos de Julio César. También se presenta el sistema de cifrado "Vigenere", que es en sí una generalización de algunos de los sistemas de cifrado que se utilizaban en aquellos tiempos, siglo dieciséis. También se presentan criptosistemas de cifrado en bloque porque son los antecedentes de DES.

El capítulo 3, "TÉCNICAS DE CRIPTOANÁLISIS", se mencionan algunos de los principales ataques que se han realizado en contra de DES; estos ataques se encuentran reportados en artículos o memorias de congresos. En este trabajo no nos interesan los ataques que no hayan sido reportados de otra forma. También es importante señalar que en esta investigación no se analizarán ataques por "hardware" [18], solamente por "software".

En el capítulo 4, "MARCO TEÓRICO", se proporcionan las herramientas teóricas necesarias para abordar el capítulo de "DESARROLLO", el cual es parte medular de esta investigación.

En el capítulo 5, "PROPUESTA", se demostrará el Teorema LR entre otras cosas. El Teorema LR como se mencionó arriba, es piedra angular para la realización de los ataques que serán descritos en este trabajo; además, se presentará un resultado empírico que es toral para el ataque que se hará a Triple-DES y una solución a este problema. Es importante señalar que los programas se corrieron en una máquina Pentium IV comercial de velocidad 2.0 Ghz.

Por último, en el capítulo 6 se presentarán las: "CONCLUSIONES y TRABAJO FUTURO"; se menciona que el autor tiene en mente proponer nuevos criptosistemas que sean tan rápidos como el AES [12], pero que el tamaño del conjunto donde están las llaves sea de 2^{500} o más y, estos criptosistemas podrían cifrar bloques de cadenas de 96 bits o más.

CAPÍTULO 2

ANTECEDENTES

La palabra Criptología tiene dos raíces griegas, a saber: criptos = oculto y logos = tratado, ciencia. Se puede considerar que es el nombre genérico de dos disciplinas que son opuestas y a su vez se complementan; Criptografía y Criptoanálisis. La Criptografía se encarga de cifrar o encriptar un mensaje de texto claro; esto es, de ocultar o enmascarar la información que se considera confidencial. En contraparte, el Criptoanálisis se encarga de romper el proceso de encriptado para recuperar la información original o el texto claro [10].

La Criptografía como una herramienta para proteger la información confidencial es un arte tan antiguo como la escritura, la cual desde sus inicios permaneció vinculada a la clase política y militar, debido al tipo de información que desde entonces se maneja en estos círculos. Se dará a continuación un ejemplo famoso que envió Julio César en el siglo I a.c. El alfabeto latino que se usaba en aquel entonces constaba de 21 letras, a saber:

A B C D E F G I J L M N O P Q R S T U V X

El procedimiento para ocultar la información en aquellos tiempos consistía en sustituir a la primera letra A por la cuarta D, la segunda B por la quinta E y así sucesivamente [10]. El mensaje famoso fue:

Mensaje	V I N I	V I D I	V I C I	(Vine vi y vencí)
Clave	D D D D	D D D D	D D D D	
Mensaje Cifrado	B M Q M	B M G M	B M F M	

Si se denomina como Y_i a las letras encriptadas, X_i a las letras del texto claro y Z_i a la clave. Entonces el cifrado anterior se puede expresar utilizando la aritmética modular [9] mediante la siguiente fórmula:

$$Y_i = X_i + Z_i \pmod{21}$$, en este caso Z_i es siempre D o 3.

Un procedimiento también famoso de cifrar es el "Cifrado Vigenere" (1586), [6,10]. Este procedimiento es una generalización del caso anterior y con objeto de ilustrarlo considere el siguiente ejemplo:

Suponga que se trabaja con el alfabeto del idioma español, el cual se escribe a continuación:

Tabla 2.1 Alfabeto Español

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

Como puede observarse las letras están numeradas del 0 al 26 con la finalidad de aplicar la aritmética modular.

Si el mensaje es "EXAMEN DE GRADO" y la clave es "VICTOR", entonces, el procedimiento de cifrado se puede expresar de la siguiente manera:

Tabla 2.2 Cifrado de mensaje

Mensaje	E	X	A	M	E	N		D	E		G	R	A	D	O
Clave	V	I	C	T	O	R		V	I		C	T	O	R	V
Mensaje Cifrado	Z	F	C	F	S	E		Y	M		I	L	O	U	K

Como puede verificarse fácilmente se aplicó la fórmula $Y_i = X_i + Z_i \pmod{27}$ y, de la misma manera que en el caso anterior Y_i representa las letras del texto cifrado, X_i las letras del texto claro y Z_i las de la clave.

Los antecedentes del criptosistema DES son los sistemas de cifrado en bloque, los cuales agrupan la información del texto claro en segmentos de dos o más símbolos cada uno, para posteriormente realizar la encriptación de la información de manera iterativa. LUCIFER [35], es un sistema de cifrado por bloques que convenientemente modificado dio origen a DES. El criptosistema LUCIFER mezcla la información dividiendo por mitades los bloques en cada una de las iteraciones o rondas [10]. De manera general se puede decir que la estructura de los sistemas de cifrado en bloque constan de los siguientes cuatro elementos:

1. Transformación inicial; que para el caso de DES es una permutación inicial PI.
2. Una función criptográfica iterada r-veces o rondas. Para DES es una función no lineal que tiene como argumentos a la mitad derecha del bloque y una clave. Esta función es iterada 16 veces.
3. Transformación final. En el sistema de encriptación DES toma la forma de una permutación que es la inversa de la inicial PI.

4. Algoritmo de expansión de clave; que en el caso de DES es un algoritmo que genera 16 llaves a partir de una cadena de 64bits. Cada una de estas llaves interviene en cada iteración.

En la actualidad, con el desarrollo de las comunicaciones y el uso masivo de las computadoras hace posible la transmisión y almacenamiento de grandes volúmenes de información confidencial. Entonces, la Criptografía pasa de ser una necesidad de élite y se convierte en una exigencia del ser humano [10]. En 1973 el NBS (National Bureau of Standards, USA) organizó un concurso solicitando un "algoritmo de encripción para la protección de datos de computador durante su transmisión y almacenaje". En 1974, la corporación IBM presentó, entre otros, una propuesta inspirada en su sistema propietario LUCIFER y le llamó: "Data Encryption Standard" (DES) [6].

Antes de iniciar la descripción del algoritmo DES será necesario precisar la simbología que será utilizada. Las letras mayúsculas como: *X,Y,L,R,K* representarán cadenas; entendiéndose por cadena a la concatenación de ceros y unos de longitud finita. En el caso de DES no serán mayor a 64. Se deja para el capítulo de "MARCO TEÓRICO" una definición más precisa de cadena. En la operación binaria "xor" se utiliza el símbolo $\oplus$ y se define de la siguiente manera:

Si x_i, y_i son elementos de las cadenas *X, Y* entonces

$$x_i \oplus y_i = \begin{cases} 0 & \text{si } x_i = y_i \\ 1 & \text{si } x_i \neq y_i \end{cases}$$

Note que los elementos de una cadena dada se escribirán con minúsculas.

Para los símbolos del texto claro y del texto cifrado serán utilizados caracteres de 8 bits, lo que corresponde a los símbolos del código ASCII. Las tablas que serán mostradas en lo que resta de este capítulo están de acuerdo con la norma internacional [39].

A continuación se hará una descripción del algoritmo DES; inicialmente se realizará una presentación de alto nivel (a grandes rasgos), para posteriormente hacer una descripción más detallada de algunos aspectos del mismo. El algoritmo procede de acuerdo con los siguientes 3 pasos:

a) Dada una cadena de texto claro, *X*, de 64 bits; se construye a partir de ésta otra cadena X_0 aplicando una permutación inicial fija PI. Esto último se escribe como: PI(*X*) = X_0. La cadena X_0 se divide en dos subcadenas de 32 bits cada una lo que se denota como: $X_0 = L_0 R_0$. L_0 serán los primeros 32 bits y R_0 los 32 bits restantes. La permutación inicial fija se ilustra a continuación:

Tabla 2.3 Permutación inicial PI

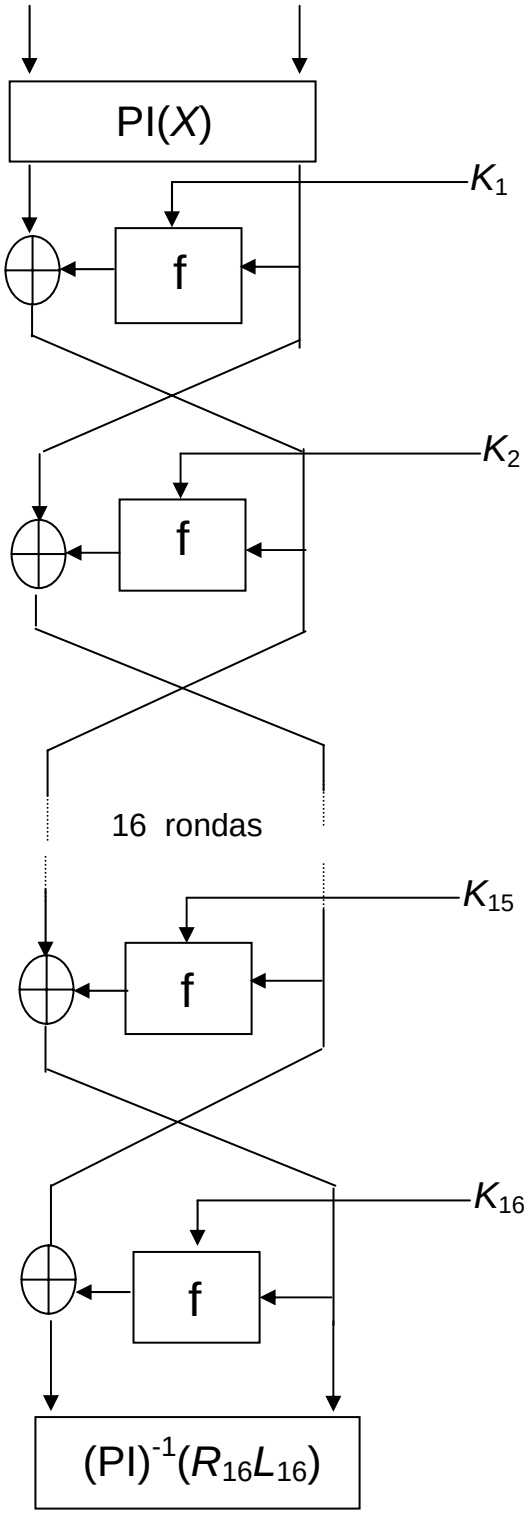
PI							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

b) Se aplican 16 iteraciones o rondas y en cada una de ellas se calculan L_i, R_i basándose en las siguientes expresiones:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus \mathrm{f}(R_{i-1}, K_i) \text{ con } i = 1,2,..,16.$$

f es una función que será descrita más adelante y con relación a la cadena K_i se dirá que es de 48 bits de longitud. Esta cadena de 48 bits se obtiene a su vez de una llave K de 64 bits de longitud. La descripción gráfica del algoritmo, (**FIPS PUB 46-3**), se puede expresar como sigue:

Texto claro (64 bits)



Texto cifrado (64 bits)

Figura 2.1 Gráfica del algoritmo DES

c) Se aplica la permutación inversa PI^{-1} a la cadena $R_{16}L_{16}$ (note que primero aparece R_{16} y después L_{16}) para obtener el texto encriptado *Y*. La tabla de PI^{-1} se presenta a continuación:

Tabla 2.4 Permutación inversa de PI

PI^{-1}							
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

Como se mencionó en el inciso b la función f tiene dos argumentos, a saber: el primero, R_{i-1}, es una cadena de 32 bits y el segundo, K_i, es una cadena de 48 bits. El resultado de la función f es una cadena de 32 bits. Todo este proceso se ejecuta de acuerdo con los siguientes pasos:

1) El primer argumento R_{i-1} es expandido a una cadena de 48 bits basándose en una función de expansión fija E. E(R_{i-1}) toma los bits de la cadena R_{i-1}, 32 en total, y los permuta-expande repitiendo a 16 de ellos. El resultado es una cadena de 48 bits. A continuación se presentará la tabla de la función de expansión:

Tabla 2.5 Función de expansión

E función de expansión					
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) Se calcula $E(R_{i-1}) \oplus K_i$ y el resultado se puede observar como la concatenación de 8 subcadenas de 6 bits cada una; esto es: $B_1 B_2 \ldots . B_8$.

3) En este paso se hace uso de 8 cajas; $S_1, S_2, \ldots, S_8$. Cada una de ellas es un arreglo fijo de 4x16 y los elementos del arreglo son enteros entre 0 – 15. Para cada subcadena $B_j = b_1 b_2 b_3 b_4 b_5 b_6$, con $1 \leq j \leq 8$, se calcula $S_j(B_j)$ como sigue: los bits $b_1 b_6$ indican el número de renglón y $b_2 b_3 b_4 b_5$ indican el número de columna. El resultado de $S_j(B_j)$ es un entero entre 0 – 15, el cual se expresa mediante una cadena de 4 bits. Es claro entonces que la cadena $C = S_1(B_1) S_2(B_2) \ldots S_8(B_8)$ es de 32 bits de longitud. Hay algo más que agregar en relación con las cajas de DES y es que éstas tienen varias propiedades. Estas propiedades pueden consultarse en [6]. En el capítulo de "MARCO TEÓRICO" se hablará de la más importante de estas propiedades, de acuerdo con el criterio del autor, que es la no linealidad de la salida de las cajas en relación con la entrada. Para ilustrar cómo trabajan las cajas se presenta a continuación la primera de ellas con una entrada particular.

Tabla 2.6 Primera caja de sustitución

S_1															
14	4	13	1	2	15	**11**	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Si se supone que la entrada a la caja es la cadena 001100 entonces $S_1(001100)=$ 1011.

En este punto será conveniente mostrar las 7 cajas restantes.

Tabla 2.7 Segunda caja de sustitución

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Tabla 2.8 Tercera caja de sustitución

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Tabla 2.9 Cuarta caja de sustitución

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Tabla 2.10 Quinta caja de sustitución

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Tabla 2.11 Sexta caja de sustitución

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Tabla 2.12 Séptima caja de sustitución

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Tabla 2.13 Octava caja de sustitución

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	13	12	9	0	3	5	6	11

4) La cadena *C* de 32 bits es permutada de acuerdo con una permutación fija P y el resultado P(*C*) es lo que se define como f (R_{i-1}, K_i). La permutación P se presenta a continuación:

Tabla 2.14 La permutación P en la salida de las cajas

P			
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 representación gráfica de la función f es como sigue:

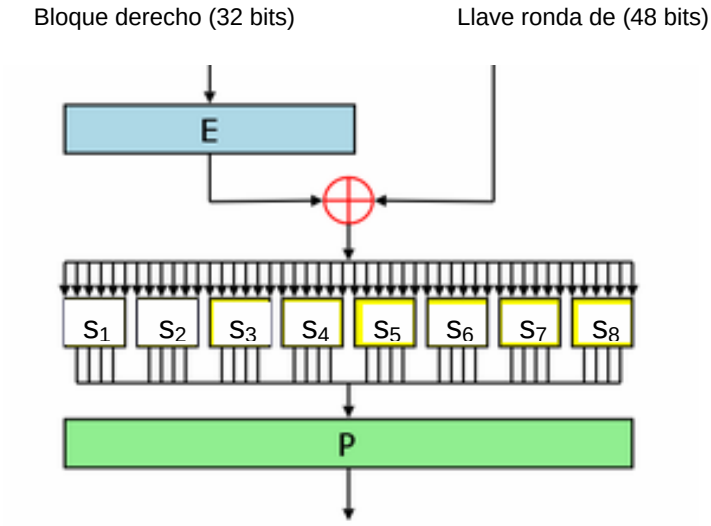


Figura 2.2 Representación gráfica de la función f de Feistel

Finalmente es necesario mencionar cómo se calculan las llaves programadas a partir de una llave *K*, la cual es inicialmente una cadena de 64 bits. A la llave *K* se le aplica la permutación PC-1 que es fija, esta permutación elimina 8 bits. Por norma las posiciones de

estos bits son: 8, 16, 24, 32, 40, 48, 56 y 64. A estos bits se les denomina 'bits de paridad'. Para obtener las llaves programadas se deben ejecutar los siguientes pasos:

a) A la llave *K* de 64 bits se le aplica la permutación fija PC-1. Se escribe el resultado de esta permutación como: PC-1(*K*) = $C_0 D_0$. C_0 son los 28 primeros bits y D_0 son los 28 restantes. La tabla de la permutación PC-1 se muestra a continuación:

Tabla 2.14 La permutación PC-1

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

b) Para $1 \leq i \leq 16$ se evalúan las siguientes expresiones:

$$C_i = LS_i(C_{i-1}),$$
$$D_i = LS_i(D_{i-1}) \text{ y}$$
$$K_i = \text{PC-2}(C_i D_i)$$

LS_i representa un corrimiento de una o dos posiciones a la izquierda dependiendo del valor de *i*. Por norma se corre una posición cuando *i* = 1,2,9 o 16 y se corren dos posiciones de otra manera. PC-2 es otra permutación fija que elimina 8 bits, dando como resultado una cadena de 48 bits. La permutación PC-2 es la siguiente:

Tabla 2.14 La permutación PC-2

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Para ilustrar lo anteriormente expuesto a continuación se describe gráficamente el cálculo de las llaves programadas.

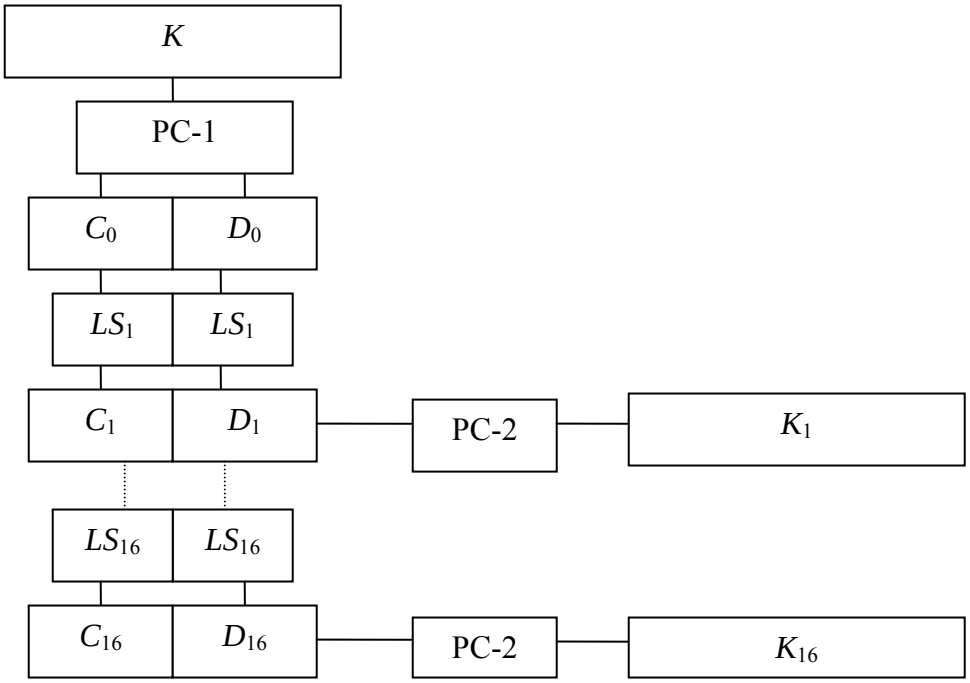


Figura 2.2 Algoritmo de las llaves programadas o ronda

Para concluir esta parte será necesario hacer una mención del Criptoanálisis ya que es de importancia toral en este trabajo. En este momento se verán aspectos generales de este tópico; para que en el capítulo "TÉCNICAS DE CRIPTOANÁLISIS" se analicen los ataques más importantes que se le han hecho a DES.

Como ya se comentó al principio de esta sección, la disciplina del Criptoanálisis se encarga de romper los sistemas de cifrados; esto es, encontrar la clave o llave utilizada. Además, busca los puntos débiles de un algoritmo de encripción con el objeto proponer soluciones para fortalecerlo, este aspecto también será presentado en este trabajo. Otra situación importante que se debe mencionar es que la robustez de un algoritmo de cifrado no depende de su desconocimiento; al contrario, éste debe ser público para que la comunidad criptográfica lo analice y de esta forma se fortalezcan sus puntos débiles.

No se puede hablar de un procedimiento general de Criptoanálisis, pues cada algoritmo de cifrado ha de ser atacado mediante una técnica adecuada dependiendo de su estructura. Es claro que el Criptoanálisis depende también de cuánta información se posea del procedimiento de cifrado, a continuación se dará una lista en orden de facilidad creciente de situaciones frecuentes [10].

a) Sólo se conoce el texto cifrado; en realidad esta es la situación más difícil si se considera que los sistemas de cifrado actuales son bastante robustos.
b) Se conoce una parte del texto claro y su correspondiente texto cifrado; ésta es una situación común en todos los ataques que se le han hecho a DES sin considerar el que se presentará en este trabajo.
c) Se conoce una parte del texto claro y su correspondiente texto cifrado; además, de una pequeña parte del protocolo de cifrado. Esta situación es la que se presentará en este trabajo.
d) Se conoce parte de la clave o llave, o se puede limitar el espacio de los posibles valores que pudiera tomar la clave o llave.

CAPÍTULO 3

TÉCNICAS DE CRIPTOANÁLISIS

En esta parte se verán cuatro formas conocidas de ataque a DES; en cada una de ellas se da por descontado que se conoce una parte del texto claro y su correspondiente texto cifrado. El tamaño mínimo de los bloques de texto claro y cifrado que se deben conocer es de 64 bits. También estas formas de ataque se pueden clasificar en dos grandes rubros, a saber: las aleatorias y las determinísticas.

Las aleatorias como su nombre lo indica, son aquéllas en que la búsqueda de la llave desconocida, bajo las condiciones del ataque, puede tener éxito o no. Las determinísticas son aquéllas que, bajo las condiciones del ataque, siempre el resultado será éxito.

3.1 ATAQUE DE FUERZA BRUTA

Este será el primer tipo de ataque que analicemos, el cual consiste en probar todas las llaves posibles; esto es, 2^{56} posibilidades, y con ello localizar la llave en uso. Desde un punto de vista teórico se puede decir que este es el ataque más simple; claro está, de ninguna manera se desea quitarle el mérito al diseño y desarrollo tecnológico que son muy importantes. A continuación se expondrá una breve reseña histórica de los ataques de este tipo.

Desde sus inicios en 1977 DES fue muy criticado por lo corto de su llave, ya que consta de 56 bits solamente. También desde sus inicios se propuso la construcción de máquinas para realizar un ataque de fuerza bruta. La primera de ellas y que está bien documentada [14], presenta un diseño para hacer un ataque de esta clase con un costo de 20 millones de dólares. El diseño de esta máquina tenía como propósito probar 2^{38} llaves por segundo.

En las siguientes dos décadas se publicaron varios diseños con esta característica, se puede decir que el más importante de ellos [15], proyectaba la construcción de una máquina con un costo de 10 millones de dólares y un promedio de búsqueda de $2^{44.71}$ llaves por segundo.

En julio de 1998 la corporación “Electronic Frontier Foundation”, EFF, construyó la máquina EFF DES Cracker machine, también denominada como “Deep Crack”. Esta máquina encontró la llave en 56,05 horas y su promedio de búsqueda fue de $2^{36.37}$ llaves por segundo; esta información puede ser localizada en http://www.eff.org/descrack. Menos de un año después, enero de 1999, esta misma corporación presentó una máquina que encontró la llave en 22.5 horas con un promedio de búsqueda de $2^{37.53}$ llaves por segundo, esta información puede ser localizada en http://crytome.org/cracking-des.htm. De los ataques más recientes y realizados por esta misma corporación están referidos en [7].

Por último, se menciona que este es el ataque más importante que se le ha hecho al criptosistema DES.

3.2 BALANCE DE TIEMPO- MEMORIA

El análisis Balance de Tiempo – Memoria [1], el cual es aleatorio, es el segundo ataque que se analizará. En este tipo de ataque se definen dos funciones que se denotan como: R, g. R es una función de reducción con dominio en el conjunto de cadenas de 64 bits e imagen en el conjunto de cadenas de 56 bits; esto es, elimina 8 bits.
Como ejemplo de una función de reducción de 64 a 56 bits podría ser aquélla que elimina los bits 8,16,24,…,64; o sea, los de paridad.

Si se denomina como: $e_K(X) = Y$ a la aplicación de un ciclo del algoritmo DES al texto claro X con la llave K y el resultado es Y; entonces, se define a la función g con dominio en el conjunto de las llaves **K**, cuyos elementos son cadenas de 56 bits, como sigue: $g(K) = R(e_K(X))$. Se debe aclarar que las llaves se consideran de 56 bits y no de 64 y esto porque es equivalente tomar como llave a $K = C_0D_0$. Ahora bien, la intención es hacer trabajo de cómputo previo para posteriormente realizar un ataque, utilizando de manera ingeniosa a las funciones R, g. Teniendo esto en mente, se eligen al azar m cadenas de 56 bits que se denotan como: $X(i,0)$ para $1 \leq i \leq m$ y se efectúan cálculos para determinar las cadenas $X(i,j)$ de acuerdo con la siguiente regla de recurrencia $X(i,j) = g(X(i,j-1))$ para $1 \leq j \leq t$; como puede observarse hay dos enteros positivos m,t a los que un poco más adelante se les darán valores dependiendo de los recursos computacionales disponibles.

Se construye una tabla de parejas ordenadas que se representa como: $T(X(i,t),X(i,0))$, es importante señalar que no se almacenan a los elementos $X(i,1)\ldots X(i,t-1)$. Ahora, si la llave que se busca fuese $K = X(i,t-j)$ para algún j con $1 \leq j \leq t$; entonces

$$\begin{aligned} g(X(i,t-j)) &= X(i,t-j+1) \\ g(X(i,t-j+1)) &= X(i,t-j+2) \\ g(X(i,t-j+2)) &= X(i,t-j+3) \qquad j \text{ - veces} \\ &\vdots \\ g(X(i,(t-j) + (j-1))) &= X(i,t) \end{aligned}$$

Si se denota a la aplicación de j-veces la función g como g^j, entonces $g^j(X(i,t-j)) = X(i,t)$. Sin embargo, $g^j(K)$ se puede expresar como

$$\begin{aligned} g^j(K) &= g^{j-1}(g(K)) \\ &= g^{j-1}(R(e_K(X))) \\ &= g^{j-1}(R(Y)) \text{ donde } Y \text{ es el texto cifrado.} \end{aligned}$$

De aquí, si se define de forma recursiva a Y_j como sigue

$$Y_j = \begin{cases} \mathrm{R}(Y) \text{ si } j = 1 \\ \mathrm{g}(Y_{j-1}) \text{ si } 2 \leq j \leq t \end{cases}$$

Entonces si $Y_j = X(i,t) = \mathrm{g}^j(K)$ se puede utilizar este hecho para averiguar cuál es el valor de j que hace $\mathrm{g}^j(X(i,t\text{-}j)) = X(i,t)$. Resumiendo, se computan las cadenas $Y_1, Y_2, \ldots Y_j$ $1 \leq j \leq t$; si alguna de ellas y para algún i se tiene $Y_j = X(i,t)$, entonces es posible que $X(i,t\text{-}j)$ sea la llave K.

Se dice que es posible que $X(i,t\text{-}j)$ sea la llave porque para ir de $X(i,t\text{-}j)$ a $X(i,t)$ existen más de una forma; de hecho, en el cálculo de cualquier cadena $X(i,t\text{-}j)$ hay 256 posibilidades previas que dan este resultado y esto por la función de reducción R. Entonces para la búsqueda de la posible llave se procedería de la siguiente manera:

a) Calcule $Y_1 = \mathrm{R}(Y)$
b) Para $j = 1$ hasta t; ejecute:
c) Si $Y_j = X(i,t)$, para algún i entonces:
d) Calcule $X(i,t\text{-}j)$ a partir de $X(i,0)$ como $\mathrm{g}^{t-j}(X(i,0))$
e) Si $\mathrm{e}_{X(i,t-j)}(X) = Y$
f) Haga $K = X(i,t\text{-}j)$ y termine
g) Calcule $Y_{j+1} = \mathrm{g}(Y_j)$

Si se cumple que $\mathrm{e}_{X(i,t-j)}(X) = Y$ de acuerdo con el inciso e, entonces se habrá encontrado la llave K; claro está, existe la posibilidad de dos escenarios negativos, a saber:

1) $\forall\, j\ Y_j \neq X(i,t)$ con $1 \leq j \leq t$, $1 \leq i \leq m$
2) Si $Y_j = X(i,t)$ entonces $\mathrm{e}_{X(i,t-j)}(X) \neq Y$

Esta forma de criptoanálisis sugiere para un caso práctico que $t \cong m \cong \mathrm{N}^{1/3}$ donde $\mathrm{N} = 2^{56}$. Además, se construyen aproximadamente $\mathrm{N}^{1/3}$ tablas del tipo $\mathrm{T}(X(i,0),X(i,t))$ cada una con una función de reducción diferente. Entonces se tiene en teoría $\mathrm{N}^{2/3}$ cadenas $X(i,t)$ diferentes, sin embargo, en la práctica muchas de ellas se repiten. Se estima, de acuerdo con el mismo trabajo que aproximadamente el 20% de ellas se repiten [1]; por lo que el 80% de ellas son diferentes. El número total de llaves es $\mathrm{N} = 2^{56}$, entonces la posibilidad de que $Y_j = X(i,t)$ se estima que es

$$[(0.8)mt]/\mathrm{N} \cong [(0.8)\mathrm{N}^{2/3}]/\mathrm{N}$$

$$\cong (0.8)/\mathrm{N}^{1/3}$$

Esta sería la posibilidad para una función de reducción y como se tienen $N^{1/3}$ de ellas, entonces la posibilidad de la igualdad anterior sería de aproximadamente 0.8. El tamaño de memoria requerida se aproxima de la siguiente forma.

Si para cualquier par $X(i,0)$, $X(i,t)$ se requieren de 112 bits y se tienen $N^{2/3}$ de ellos, entonces se requerirá de $112*N^{2/3}$ bits de memoria, lo que aproximadamente es 2^{45} bits.

3.3 CRIPTOANÁLISIS DIFERENCIAL

Ésta es la tercera forma de ataque que se abordará. En este tipo de procedimiento se utiliza un número grande (del orden de 2^{47}) de parejas de textos claros y sus correspondientes textos cifrados [3]. Los conceptos y la notación que será utilizada a continuación fueron tomados de [6].

En el criptoanálisis diferencial se utiliza básicamente la operación xor en las cadenas que intervienen en la entrada y la salida de las cajas. Se sabe del capítulo "ANTECEDENTES" que para la *i*-ésima ronda con $1 \leq i \leq 16$ la entrada a las cajas se puede expresar como: $B=E(R_{i-1}) \oplus K_i$; de hecho, la cadena B se puede observar como la concatenación de 8 subcadenas de 6 bits cada una; esto es, $B=B_1B_2\ldots B_8$. Entonces, para dos cadenas de entrada a las cajas B y B^*, la cadena $B'=B\oplus B^*$ no depende de la llave K y se le llamará la entrada xor a las cajas. De la misma manera, la salida xor de las cajas será definida como: $C'=C\oplus C^*$ la cual si depende de la llave. Todos los conceptos y procedimientos que serán mencionados de aquí en adelante se supondrán que suceden en la *i*-ésima ronda, con $1 \leq i \leq 16$, y antes de entrar en materia será necesario que se hagan las siguientes definiciones:

Definición 3.1.- Dada una caja S_j con $1 \leq j \leq 8$ considere al par ordenado de cadenas de longitud 6 (B_j, B^*_j); entonces, llamamos a $B'_j = B_j \oplus B^*_j$ la entrada xor a la caja S_j y a $C'_j = Sj(B_j) \oplus Sj(B^*_j)$ la salida xor de la caja S_j.

Definición 3.2.- Para una B'_j dada se define al conjunto $\Delta(B'_j)$ como: $\Delta(B'_j) = \{(B_j, B_j \oplus B'_j) \mid B_j \in (\mathbf{Z}_2)^6\}$, aquí $\mathbf{Z}_2$ es el conjunto $\{0,1\}$.

Se desprende fácilmente que el número de elementos de $\Delta(B'_j)$ es 2^6 ya que B'_j es fija. El número de elementos de este conjunto se le denotará como $|\Delta(B'_j)|$. Es conveniente mencionar en este punto que se utilizará como subíndice a '*j*' cuando se refiera a bloques o subcadenas e *i* cuando se refiera a rondas.

Considere que se pasa por la caja S_j a cada uno de los elementos de $\Delta(B'_j)$; o sea, al conjunto de parejas ordenadas $(B_j, B^*_j \oplus B'_j)$ para obtener a las cadenas $C'_j = S_j(B_j) \oplus S_j(B^*_j) = S_j(B_j) \oplus S_j(B_j \oplus B'_j)$. Los 64 elementos de $\Delta(B'_j)$ se dividen en conjuntos disjuntos tal que a

cada uno de ellos le corresponde un C'_j específico. Con base en esta idea se define a $\mathbf{IN}_j(B'_j, C'_j)$ como sigue:

Definición 3.3.- Para $1 \le j \le 8$ y dos cadenas B'_j de 6 bits y C'_j de 4 bits dadas, entonces se define al conjunto $\mathbf{IN}_j$ como: $\mathbf{IN}_j(B'_j, C'_j) = \{ B_j \mid S_j(B_j) \oplus S_j(B_j \oplus B'_j) = C'_j \}$

Con objeto de aclarar los conceptos anteriormente expuestos será conveniente presentar un ejemplo. Suponga que de forma particular se sabe que B'_1=110100 es la entrada xor a la primera caja S_1. Entonces el conjunto $\Delta(110100) = \{(000000,110100), (000001,110101),\ldots, (111111,001011\}$; ahora bien, si para cada pareja de $\Delta(110100)$ se obtiene la salida xor de la caja S_1, la distribución de las salidas xor es la siguiente:

Tabla 3.1 Frecuencias de salida xor para una entrada xor 110100

0000	0001	0010	0011	0100	0101	0110	0111
0	8	16	6	2	0	0	12

1000	1001	1010	1011	1100	1101	1110	1111
6	0	0	0	0	8	0	6

Se pueden mostrar de manera particular cuáles son los 16 elementos de $\mathbf{IN}_1$ cuando C'_1=0010; o sea , $\mathbf{IN}_1(110100,0010)$= {00100,000101,001110,010001,010010,010100, 011010,011011,100000,100101,010110,101110,101111,110000,110001,111010}

Se sabe que la entrada a las cajas se puede expresar como: $B=E(R_{i-1}) \oplus K_i$; por lo que:

$$B' = B \oplus B^*$$
$$B' = (E(R_{i-1}) \oplus K_i) \oplus (E(R^*_{i-1}) \oplus K_i)$$
$$B' = E(R_{i-1}) \oplus E(R^*_{i-1})$$

Con la intención de simplificar la notación se escribirán a $E(R_{i-1})$ y $E(R^*_{i-1})$ como: E y E^* respectivamente. Entonces de lo anteriormente expuesto se sigue que $B'=E'$.

Ahora bien, las cadenas B, E y K_i se pueden ver como la concatenación de 8 subcadenas de 6 bits cada una; esto es:

$$B=B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ B_8$$

$$E=E_1\ E_2\ E_3\ E_4\ E_5\ E_6\ E_7\ E_8$$

$$K_i=K_{i1}\ K_{i2}\ K_{i3}\ K_{i4}\ K_{i5}\ K_{i6}\ K_{i7}\ K_{i8}$$

Supóngase por un momento que se conoce una subcadena K_{ij}, se sigue que $B_j = E_j \oplus K_{ij}$ y además, que B_j es un elemento del conjunto $\mathbf{IN}_j$. Entonces es lógico que a cada uno de los elementos de $\mathbf{IN}_j$ se les aplique la operación xor con E_j y que alguno de ellos sea el que tiene la información de la llave, ya que $K_{ij} = B_j \oplus E_j$.

Con esta idea en mente se puede definir al conjunto $\mathbf{Prueba}_j$ como sigue:

Definición 3.3.- Supóngase que se conocen las cadenas E_j , E^*_j de 6 bits de longitud y a C'_j de 4 bits de longitud. Entonces se define al conjunto $\mathbf{Prueba}_j (E_j , E^*_j, C'_j) = \{ B_j \oplus E_j \mid B_j \in \mathbf{IN}_j (E'_j , C'_j) \}$.

Para aclarar los conceptos considere el siguiente ejemplo:

Suponga que E_1= 000001, E^*_1 = 110101 y C'_1 = 1101, con esta información se pueden obtener los conjuntos $\mathbf{IN}_1$ y $\mathbf{Prueba}_1$ de la siguiente manera: dado que $B'_1 = E'_1$ = 110100 se sigue que $\mathbf{IN}_1$(110100,1101) = {000110, 010000, 010110, 011100, 100010, 100100, 101000, 110010}. De aquí el conjunto $\mathbf{Prueba}_1$ sería :

$\mathbf{Prueba}_1$(000001, 110101, 1101) = {000111, 010001, 010111, 011101, 100011, 100101, 101001, 110011}.

Ahora bien, con esta información la intención es tener varias tercias de E_j , E^*_j y C'_j con la finalidad de construir varios conjuntos $\mathbf{Prueba}_j$; obviamente la subcadena de la llave correcta estará en la intersección de estos conjuntos dado que la llave K de donde se calculan las llaves programadas se considera fija. En la práctica se elige a la subcadena que más se repite. La técnica del Criptoanálisis diferencial encuentra los bits de la llave programada K_{16}, aunque podría ser K_1, pues siguiendo el procedimiento descrito en [3], se llega a conocer la entrada y salida xor de las cajas en la ronda 16. En realidad se busca que la entrada y la salida queden en función de los bits del texto claro y del texto cifrado. Para hacer uso de este procedimiento, es necesario utilizar la herramienta que se conoce como "n-ronda característica"; en la referencia [6] se define este concepto. De acuerdo con la misma referencia [6] es necesario escoger 2^{47} juegos de textos claros y su correspondiente texto cifrado para llevar a cabo este ataque.

3.4 CRIPTOANÁLISIS LINEAL

Antes de iniciar la descripción del ataque lineal será conveniente dar algunas definiciones y ejemplos con objeto de comprender este ataque [11,12].

Definición 3.4.- Se dice que una función f es booleana si su dominio está en $(\mathbf{Z}_2)^n$ y su imagen está en $\mathbf{Z}_2$, aquí $\mathbf{Z}_2 = \{0,1\}$.

Como ya se ha mencionado anteriormente, la entrada a las cajas en cualquier ronda se representa como la cadena $B = B_1 B_2 \ldots B_8$; en donde cada subcadena B_j es de 6 bits de longitud. Entonces se puede decir que $B_j \in (\mathbf{Z}_2)^6$ o también se puede expresar como $(x_1, x_2, x_3, x_4, x_5, x_6)$ con $x_j \in \{0,1\}$. Siguiendo esta misma idea la salida de las cajas puede denotarse como $C = C_1 C_2 \ldots C_8$; en donde cada subcadena C_j es de 4 bits de longitud y de igual forma que en el caso anterior C_j se puede expresar como $(y_1 y_2 y_3 y_4)$ con $y_i \in \{0,1\}$. Entonces utilizando la definición 3.4 se desprende fácilmente que se pueden definir 4 funciones booleanas para cada una de las cajas como sigue:

Si B_j, C_j son la entrada y la salida de la caja j entonces: $f_1(x_1, x_2, x_3, x_4, x_5, x_6) = y_1$; $f_2(x_1, x_2, x_3, x_4, x_5, x_6) = y_2$; $f_3(x_1, x_2, x_3, x_4, x_5, x_6) = y_3$; $f_4(x_1, x_2, x_3, x_4, x_5, x_6) = y_4$.

Definición 3.5.- Si f_i es una función booleana, con $i = 1,..4$, entonces se define a $\rho(f_i) = \sum_{X \in Z_2^6} (-1)^{f_i(X)}$

Con base en la definición 3.4 f_i puede tomar valores de 0 ó 1; entonces cuando la función ρ tiene un valor negativo quiere decir que f_i toma más veces el valor 1 que el 0 y, si tiene un valor positivo sucede lo contrario. Entonces podemos decir que la función ρ puede medir los sesgos de las cajas; esto es, que un bit de salida tome más veces un valor que otro para los diferentes valores de entrada. Para seguir con este análisis será necesario mencionar algunos tipos de notación que serán usados; por ejemplo: $D_{\mathbf{a}} = \bigoplus_{j=1}^{6} a_i x_i$ significa una sumatoria xor; o sea, $\bigoplus_{j=1}^{6} a_i x_i = a_1 x_1 \oplus a_2 x_2 \oplus a_3 x_3 \oplus a_4 x_4 \oplus a_5 x_5 \oplus a_6 x_6$ y la notación de valor absoluto utilizando dos barras $\left| \quad \right|$.

Definición 3.6.- Dadas las funciones f_i y la sumatoria $D_{\mathbf{a}}$; entonces la transformada de Walsh se define como: $W_{fi}(\mathbf{a}) = \rho(f_i \oplus D_{\mathbf{a}})$.

Es importante señalar que **a** es un vector de la forma: $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)$ con las $a_i \in \{0,1\}$.

En la referencia [11] se hace uso de la transformada de Walsh para definir una medida de la no linealidad de una función booleana f_i, y en el caso de las cajas utilizadas en DES esta medida sería como sigue:

$$2^5 - (1/2) \max_{\mathbf{a} \in z_2^6} \left| W_{fi}(\mathbf{a}) \right|$$

El mayor grado de no linealidad se obtiene cuando la expresión anterior toma el valor 32, esto querría decir que no hay sesgo. Esta situación no ocurre en el caso del criptosistema DES; de hecho, el ataque lineal utiliza la propiedad del sesgo que tienen las cajas de DES.

Será conveniente en este punto ilustrar con un ejemplo lo anteriormente expuesto. El conjunto de todas las posibles cadenas $x_1\ x_2\ x_3\ x_4\ x_5\ x_6$ sería: {000000, 000001, …, 111111}. Si se considera en este ejemplo a la primera caja, la función booleana f_1 y al vector **a** = (0,0,1,0,1,1) entonces podríamos calcular a: $2^5 - (1/2) \left| W_{f_1} (0,0,1,0,1,1) \right|$. El valor de $(1/2) \left| W_{f_1} (0,0,1,0,1,1) \right| = (1/2) \left| -38 + 26 \right| = 6$. Se sigue que el sesgo para este vector particular es de 6; además, el nivel de no linealidad para este caso particular es de 32 – 6 = **26**. Ahora bien, si se desea obtener el valor de no linealidad para la primera función booleana de la primera caja, entonces se debe calcularse el valor de: $2^5 - (1/2) \max_{\mathbf{a} \in Z_2^6} \left| W_{f_i} (\mathbf{a}) \right|$, el cual es 18 y se toma cuando el vector **a** = (1,1,1,0,1,1). De aquí se desprende fácilmente que el valor de no linealidad de una caja se obtiene a partir del mayor de sus sesgos, que para este caso es de 14.

A continuación se muestran en cuatro tablas los valores de no linealidad de las f_i con $1 \leq i \leq 4$, para cada una de las 8 cajas:

Tabla 3.2 Medida de la no linealidad para las cajas 1,2

	CAJA 1		CAJA 2	
	Vector **a**	No linealidad	Vector **a**	No linealidad
f_1	111011	18	011111	18
f_2	011011	22	110111	18
f_3	111100	22	111110	20
f_4	111000	22	110100	22

Tabla 3.3 Medida de la no linealidad para las cajas 3, 4

	CAJA 3		CAJA 4	
	Vector **a**	No linealidad	Vector **a**	No linealidad
f_1	101111	18	111011	22
f_2	110111	20	111010	22
f_3	111011	22	111010	22
f_4	111011	18	111011	22

Tabla 3.4 Medida de la no linealidad para las cajas 5, 6

	CAJA 5		CAJA 6	
	Vector **a**	No linealidad	Vector **a**	No linealidad
f_1	110111	20	111101	20
f_2	111111	18	111111	20
f_3	110100	20	111111	20
f_4	100111	22	110110	22

Tabla 3.5 Medida de la no linealidad para las cajas 7, 8

	CAJA 7		CAJA 8	
	Vector **a**	No linealidad	Vector **a**	No linealidad
f_1	111011	20	111110	22
f_2	111011	14	100111	22
f_3	111111	22	111101	20
f_4	111111	18	111010	22

De las cuatro tablas anteriormente expuestas, se puede observar fácilmente que el valor más alto de no linealidad es de 22 y que el valor más bajo es de 14; este último, se da en la caja 7 para la función f_2 y el vector **a** = (1,1,1,0,1,1). De hecho, no es complicado verificar para esta última situación usando la caja 7 que la frecuencia con que toma el valor cero la expresión: $x_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus f_2(x_1, x_2, x_3, x_4, x_5, x_6)$ es 14.

Una vez mostrados los conceptos anteriores, a partir de este punto se presentará el ataque lineal mediante una descripción de alto nivel; esto es, a grandes rasgos. A continuación se presentan los pasos que ejecuta el ataque lineal:

a) Suponga usted que se construye una tabla, en la cual se indican del lado izquierdo las 2^{48} posibles entradas a las cajas de DES, y del lado derecho las salidas correspondientes de cada una de estas entradas. Para su mejor comprensión se muestra la siguiente gráfica:

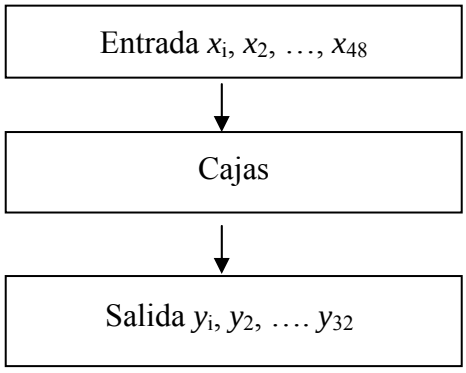


Figura 3.1. Todas las posibles entradas a las cajas con su salida

b) Se proponen 15 funciones lineales del siguiente tipo: $T_k = \left(\bigoplus_{i=1}^{48} a_i x_i\right) \oplus \left(\bigoplus_{j=1}^{32} b_j y_j\right)$

donde la a_i , b_j son constantes binarias, con $1 \leq k \leq 15$. La construcción de estas funciones lineales persigue dos objetivos, a saber:

1) Si se denota a la frecuencia con que $T_k = 0$ como: N($\mathbf{a}_k$, $\mathbf{b}_k$), donde $\mathbf{a}_k = (a_1,\ldots, a_{48})$ y $\mathbf{b}_k = (b_1,\ldots, b_{32})$. Entonces el sesgo ε_k de T_k se calcula de la siguiente forma:

$$\varepsilon_k = (N(\mathbf{a}_k , \mathbf{b}_k) - 2^{47}) / (2^{48})$$

Con este antecedente, se busca que la construcción de T_k sea de tal manera que su sesgo asociado, ε_k, tenga un valor absoluto lo más grande posible. Lo anterior se logra debido al sesgo de las cajas y la frecuencia N($\mathbf{a}_k$, $\mathbf{b}_k$) puede tomar valores lejanos a 2^{47}. Particularmente, los sesgos ε_k cumplen con la siguiente desigualdad: $-\frac{1}{2} \le \varepsilon_k \le \frac{1}{2}$

2) Que la variable aleatoria $\bigoplus_{k=1}^{15} T_k$ sea función de los bits del texto claro y de los bits de la cadena de entrada a las cajas en la ultima ronda. Hay una parte no aleatoria que es función de los bits de las llaves programadas y ésta siempre toma el valor de 1 ó 0 [12]. La consideración de no aleatoriedad se hace bajo la suposición de que la llave K de donde se derivan las llaves programadas K_i, es fija.

c) Se calcula el sesgo de la variable aleatoria $\bigoplus_{k=1}^{15} T_k$ de la siguiente manera:

$\varepsilon = 2^{14} \prod_{k=1}^{15} \varepsilon_k$ [12], bajo la suposición que las variables aleatorias T_k, con $1 \le k \le 15$, son independientes.

d) Se elimina de $\bigoplus_{k=1}^{15} T_k$ a la parte no aleatoria y se denota como T' , esto se hace con el objeto de efectuar el análisis sobre la parte aleatoria únicamente.

Ahora bien, los bits de la cadena de entrada a las cajas en la última ronda definirán los bloques o subcadenas de la última llave programada, K_{16}, que serán tomados en cuenta. Llamémosle a un valor particular de los bloques de la última llave programada como “la subllave candidato”.

Sea $|T|$ = el número de parejas de texto claro y su correspondiente texto cifrado, llamemos γ al conjunto de estas parejas. Para cada pareja de γ descifremos

parcialmente proponiendo un bloque particular de 6 bits al cual se le denomina "subllave candidato" y, además, utilizando el texto encriptado. La intención es sustituir los valores de los bits que aparecen en la función T′.

e) Se lleva un contador para cada una de las "subllaves candidatos"; la idea es que cuando la subllave candidato no es la adecuada entonces $\Pr[\ T'=0]=\frac{1}{2}$; o sea, $\varepsilon = 0$. Sin embargo, si la subllave es la adecuada entonces $\Pr[\ T'=0]=\frac{1}{2}\pm\varepsilon$

El inventor del criptoanálisis lineal es Matsui, ver referencia [13], él realizó un ataque a DES en 1994 y utilizó 2^{43} parejas de textos claros y su correspondiente texto encriptado. La ejecución de este ataque duró aproximadamente 50 días [12]. Si el lector desea profundizar más sobre este tipo de ataque, puede consultar las referencias [12,13]. De hecho, la primera de estas dos referencias muestra el ataque con un ejemplo de un cifrado iterativo sencillo.

3.5 El ataque mejorado de Davies.

Éste será el último tipo de criptoanálisis que se verá. Mientras que el análisis lineal y diferencial son técnicas generales y pueden aplicarse a multitud de esquemas diferentes, el ataque de Davies es una técnica especializada para DES. Propuesta por vez primera por Davies en los 80 y, mejorada por Biham y Biryukov (1997). La forma más potente del ataque requiere 2^{50} textos claros escogidos y tiene una complejidad computacional de 2^{50}; además, tiene un 51% de probabilidad de éxito [38].

Algunas reflexiones del autor. Se debe aclarar que el ataque más importante que se le ha hecho a DES es el de fuerza bruta. En realidad, cuando DES sale de la norma en 1998 [16], también es el año en que se realiza el ataque con la máquina "EFF DES Cracker machine" [7]. Sin embargo los ataques como: Diferencial y Lineal aunque no influyeron directamente en la salida de DES de la norma; ya que el primero salió a la luz en 1991 [37] y el segundo en 1994 [13]. Si se puede afirmar que ambos criptoanálisis señalaron debilidades del algoritmo de encripción DES. Lo anterior influyó en que el nuevo estándar que es AES, no se utilizan las cajas de seguridad de DES, debido a que éstas tienen sesgos. En relación a los otros ataques, se puede decir que ponen de manifiesto lo corto de la longitud de la llave. Esto último tiene como consecuencia que AES usa longitudes de llaves como: 128,192 y 256 [13].

CAPÍTULO 4

MARCO TEÓRICO

Antes de iniciar con este capítulo será necesario hacer algunas aclaraciones en relación con la notación que se usará. Cuando se escriban letras mayúsculas y en negritas denotarán conjuntos. Los símbolos como: e_K, y d_K denotarán funciones.

Es importante establecer de forma clara lo que se entiende por criptosistema, y aunque este concepto está descrito en las referencias [6] es conveniente proporcionar una definición.

Definición 4.1.- Un criptosistema es una tupla de cinco elementos (**P, C, K, E, D),** cada uno de estos elementos cumple a su vez con lo siguiente:

a) **P** es el conjunto finito de todos los posibles textos claros.
b) **C** es el conjunto finito de todos los posibles textos encriptados
c) **K** es el conjunto de las llaves, el cual tiene un número finito de elementos. El número de llaves puede ser grande pero finito.
d) Para cada $K \in \mathbf{K}$ se definen dos funciones inyectivas, a saber: $e_K \in \mathbf{E}$ y $d_K \in \mathbf{D}$ tal que e_K: $\mathbf{P} \rightarrow \mathbf{C}$ y d_K: $\mathbf{C} \rightarrow \mathbf{P}$. Además, $d_K(e_K(X)) = X$ para cualquier $X \in \mathbf{P}$; esto es, d_K es la función inversa de e_K.

Los conjuntos **P, C** descansan en un alfabeto; por ejemplo, para textos claros se podría utilizar el alfabeto del idioma español o inglés, a continuación se dará una definición de alfabeto:

Definición 4.2.- Un conjunto no vacío finito de símbolos se le llamará alfabeto [19].

El término cadena es un concepto que se citará con frecuencia en este trabajo, entonces es conveniente definirlo.

Definición 4.3.- Una secuencia finita de elementos de un alfabeto se le llamará cadena.

Si se supone que el alfabeto es el conjunto $\{0,1\}$, entonces una cadena podría ser 001111010011 que como se observa es una secuencia de 12 elementos del alfabeto $\{0,1\}$. De hecho, el texto claro y el texto encriptado se pueden considerar cadenas de símbolos; en este sentido **P y C** son conjuntos cuyos elementos son cadenas. Es claro que no son de interés la cadena vacía ni cadenas de longitud infinita.

Otro concepto muy utilizado en este trabajo es el de permutación, por lo cual es definirlo.

Definición 4.4.- Dado un conjunto **S** con un número finito de elementos, una permutación P definida sobre **S** es una función biyectiva que va de **S** a **S** y se representa como: P: $\mathbf{S} \rightarrow \mathbf{S}$.

Como ejemplo para aclarar la definición anterior suponga que $\mathbf{S}=\{1,2,3,4,5\}$; entonces un caso particular para P: $\mathbf{S} \rightarrow \mathbf{S}$ sería: P(1)=2, P(2)=5, P(3)=4, P(4)=1, P(5)=3. En este trabajo

se usarán permutaciones sobre conjuntos cuyos miembros representan las posiciones de los elementos de una cadena.

Aunque fue descrita la operación binaria ⊕ 'xor' en el capítulo de "ANTECEDENTES", no fueron mencionadas algunas propiedades importantes de esta operación. También se señala que esta operación se define en cadenas de la misma longitud sobre el alfabeto $\{0,1\}$. A continuación se presentan algunas de estas propiedades:

Sea $\mathbf{H}_m$ = { Las cadenas X de longitud m, sobre el alfabeto $\{0,1\}$ }; entonces la operación binaria 'xor' cumple con las siguientes propiedades:

a) Si $X,Y \in \mathbf{H}_m$ entonces $X \oplus Y = Y \oplus X$ & $X \oplus Y \in \mathbf{H}_m$
b) Si X,Y y $Z \in \mathbf{H}_m$ entonces $(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)$
c) $\forall\ X \in \mathbf{H}_m$ se cumple que $X \oplus X = \mathrm{O}$ donde O es una cadena de ceros de longitud m.
d) $\forall\ X \in \mathbf{H}_m$ se cumple que $X \oplus \mathrm{O} = X$.

De hecho, esta operación define un grupo abeliano [9], en el conjunto $\{0,1\}$.

Lema.- Sea P una permutación sobre las posiciones de cadenas de longitud m y sean dos cadenas X, Y de esta misma longitud; entonces:

$$\mathrm{P}(X \oplus Y) = \mathrm{P}(X) \oplus \mathrm{P}(Y).$$

Sea $x_i \oplus y_i$ el i-ésimo elemento de $X \oplus Y$; además, suponga que la permutación P lo lleva a la posición j; esto es, $\mathrm{P}(x_i \oplus y_i) = (x_i \oplus y_i)_j$

$$= (x_i)_j \oplus (y_i)_j$$
$$= \mathrm{P}(x_i) \oplus \mathrm{P}(y_i)$$

■

A continuación se analizarán dos aspectos con objeto de ilustrar el Teorema LR, los cuales son:

a) Como se vio en el capítulo 2, la permutación PC-2 se aplicaba a una cadena de 56 bits y daba como resultado una cadena de 48 bits. La cual se representa para la i-ésima ronda del algoritmo de DES como sigue: PC-2($C_i D_i$). Ahora bien, suponga que se conoce PC-2($C_i D_i$), la cadena de 48 bits, y que con esta información se desea conocer a $C_i D_i$. Parece claro que únicamente se podrán conocer 48 bits de $C_i D_i$ y que se tendrán 8 incógnitas, variables binarias, en aquellas posiciones que PC-2 eliminó. Ilustremos con un ejemplo lo anteriormente mencionado:

   Se sabe del capítulo de "ANTECEDENTES" que PC-2 es como sigue:

Tabla 4.1 Permutación de reducción PC-2

PC-2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Es sencillo observar que PC-2 elimina las posiciones: 9, 18, 22, 25, 35, 38, 43 y 54. Como un caso particular, suponga que se tiene a la primera llave programada, esto es:
$PC\text{-}2(C_1D_1)$ = 00001111010000010000010100100010001010011000111 y con esta información se desea conocer a C_1D_1, por lo que será necesario saber quien es $(PC\text{-}2)^{-1}$. A continuación se muestra esta permutación inversa:

Tabla 4.2 Permutación inversa de PC-2

$(PC\text{-}2)^{-1}$					
5	24	7	16	6	10
20	18	12	3	15	23
1	9	19	2	14	22
11	13	4	17	21	8
47	31	27	48	35	41
46	28	39	32	25	44
37	34	43	29	36	38
45	33	26	42	30	40

Entonces, se sigue que:

$(PC\text{-}2)^{-1}[PC\text{-}2(C_1D_1)]$= 11111100000000000100000111111110000010000000000 1

Sin embargo, la cadena anterior es de 48 bits y se desea una cadena de 56 bits ya que se busca a C_1D_1. La forma de resolver esta situación es agregando 8

variables binarias en las posiciones que PC-2 eliminó. Lo anterior se representa a continuación:

$(C_1D_1)^*$= 111100$x_1$00000000$x_2$010 $x_3$00 $x_4$001111111 $x_5$10 $x_6$0000 $x_7$1000000000 $x_8$01

El asterisco se escribe para indicar que hay 8 variables binarias. Por último, la respuesta a la pregunta: ¿Si se conoce a PC-2(C_1D_1) qué se puede decir de C_1D_1?. La respuesta es que hay 2^8=256 posibles soluciones.

b) El segundo aspecto tiene que ver con la pregunta ¿Qué se puede decir de la entrada a las cajas, si se conoce la salida?. Lo anterior tiene que ver con el hecho de cruzar las cajas en dirección inversa a la del algoritmo DES. De la misma forma que en el caso anterior será conveniente ilustrar esta situación con un ejemplo.

Suponga que se tiene la siguiente entrada a la primera caja en la primera ronda: B_1= 100011. Ahora, si se cruza la caja en la dirección del algoritmo se tendría lo siguiente:

Entrada B_1= 100011

↓

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	**12**	8	2	4	9	1	7	5	11	3	14	10	0	6	13

↓

Salida S(B_1= 100011) = 1100

Ahora bien, si se conoce a la salida 1100 ¿qué se puede decir de la entrada?.

Para responder a esta pregunta será necesario cruzar las cajas en sentido inverso lo que se ilustra a continuación:

Posibles entradas = 010110,010101,110010,100011

↑

S_1																
14	4	13	1	2	15	11	8	3	10	6	**12**	5	9	0	7
0	15	7	4	14	2	13	1	10	6	**12**	11	9	5	3	8
4	1	14	8	13	6	2	11	15	**12**	9	7	3	10	5	0
15	**12**	8	2	4	9	1	7	5	11	3	14	10	0	6	13

↑

Salida $S(B_1) = 1100$

Entonces como puede observarse, si se conoce la salida de las cajas y se desea saber qué sucede con la entrada, hay 4 posibles soluciones para cada bloque de 4 bits. Lo que significa que, si se conocen los 8 bloques de 4 bits cada uno, entonces habrá $4*4*4*4*4*4*4*4 = 2^{16}$ posibles soluciones [9].

Por último, en la demostración del Teorema LR también se utilizarán a las permutaciones P^{-1} y $(PC\text{-}1)^{-1}$; entonces es conveniente que se muestren a continuación:

Tabla 4.3 Permutación inversa de P

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

La permutación P^{-1} se construye a partir de la permutación P. De hecho, se desprende fácilmente que el número 1 de la permutación P está en la posición 9; razón por la cual la permutación P^{-1} inicia con el número 9. Siguiendo este mismo criterio la permutación $(PC\text{-}1)^{-1}$ es como sigue:

Tabla 4.4 La permutación inversa de PC-1

$(PC\text{-}1)^{-1}$						
8	16	24	56	52	44	36
7	15	23	55	51	43	35
6	14	22	54	50	42	34
5	13	21	53	49	41	33
4	12	20	28	48	40	32
3	11	19	27	47	39	31
2	10	18	26	46	38	30
1	9	17	25	45	37	29

Vale la pena mencionar que cuando se construyen a $(PC\text{-}2)^{-1}$ y $(PC\text{-}1)^{-1}$, se debe considerar que PC-2 elimina las posiciones: 9, 18, 22, 25, 35, 38, 43 y 54, y que PC-1 elimina las posiciones: 8, 16, 24, 32, 40, 48, 56 y 64.

CAPÍTULO 5

PROPUESTA

5.1 Técnica de criptoanálisis propuesta.

Se iniciará la exposición del criptoanálisis de DES, mencionando que la seguridad del criptosistema DES está basada en el desconocimiento de la llave; todos los demás elementos son del dominio público, este es uno de los principios de Kerckhoff [6]. Además, se da por descontado que se tiene información de un bloque de texto claro y su correspondiente texto cifrado; mínimo una cadena de 64 bits. Con esto en mente la pregunta sería ¿Qué información adicional se requerirá para romper DES?. Claro está, no se pedirá información de las llaves programadas. Antes de describir y demostrar el Teorema LR que es la clave de este asunto, será conveniente escribir un par de definiciones.

Definición 5.1.- Para una llave $K \in \mathbf{K}$ se pueden obtener $K_1, K_2, \ldots, K_{16}$ llaves programadas o también llaves rondas. Entonces, se define a $\mathbf{K_i}$ como $\mathbf{K_i} = \{$Cadenas de bits de longitud 48$\mid K_i \in \mathbf{K_i}$ y$\mid \mathbf{K_i} \mid = 2^{16} \}$ con $0 \leq i \leq 16$.

Definición 5.2.- Suponga que se aplica una llave $K^* \in \mathbf{K}$ en un criptosistema DES, entonces se define a $\mathbf{K}^* = \{$ Cadenas de bits de tamaño 64$\mid K^* \in \mathbf{K}^*$ y$\mid \mathbf{K}^* \mid = 2^{24} \}$.

A partir de estas dos definiciones se puede enunciar el Teorema LR como sigue:

Teorema LR.- Supóngase que L_{i-1}, R_{i-1}, L_i, R_i y K_i son las cadenas involucradas en la i-ésima ronda de encripción del criptosistema DES, con $0 \leq i \leq 16$. Entonces dadas las cadenas L_{i-1}, R_{i-1}, y R_i, o, L_{i-1}, L_i, y R_i existe un conjunto $\mathbf{K_i}$; además, para una llave desconocida $K^* \in \mathbf{K}$ y un conjunto $\mathbf{K_i}$ dado se puede construir un conjunto $\mathbf{K}^*$.

Demostración.

Primera parte. Una primera observación es señalar que el conocimiento de L_{i-1}, R_{i-1} y R_i es equivalente al conocimiento de L_{i-1}, L_i, y R_i ya que $L_i = R_{i-1}$. Entonces, bastará hacer la prueba para L_{i-1}, R_{i-1} y R_i. También se menciona que la demostración del teorema consta de dos partes, a saber: en la primera se construye a $\mathbf{K}_i$ y en la segunda a $\mathbf{K}^*$**.** Se inicia con la primera.

La entrada y la salida de las cajas en la *i*-ésima ronda se pueden expresar de la siguiente manera:

Entrada $\quad B^i = \mathrm{E}(R_{i-1}) \oplus K_i \qquad 5.1$

Salida $\quad C^i = \mathrm{P}^{-1}(R_i \oplus L_{i-1}) \qquad 5.2$

Por hipótesis del Teorema LR, en la expresión 5.2 se puede conocer a C^i.

Si se divide a la cadena C^i en subcadenas de tamaño 4; esto es, $C^i = C^i_1, C^i_2, \ldots, C^i_8$ donde cada C^i_j es la salida de la caja *j*. De la misma forma, también la cadena B^i se puede dividir en 8 subcadenas de longitud 6; quedándonos como: $B^i = B^i_1, B^i_2, \ldots, B^i_8$. Cada B^i_j es la entrada a la caja *j*.

Ahora bien, de la misma forma como se vio en el capítulo de "MARCO TEÓRICO" la idea es pasar de $C^i_j \rightarrow B^i_j$ (note que es la dirección contraria a la de encriptar). Se sabe del mismo capítulo citado que hay 4 posibles soluciones, y esto, porque el entero C^i_j aparece en el primero, segundo, tercero y cuarto renglón de la caja j. Esto último, conduce a la construcción de un conjunto, llamémosle $\mathbf{B}^i$, con $4*4*4*4*4*4*4*4 = 2^{16}$ elementos.

$$\mathbf{B}^i = \{\, B^i \mid \mathbf{S}(B^i) = C^i \,\} \qquad 5.3$$

Se entiende a $\mathbf{S}(B^i)$ como el proceso de pasar a la cadena B^i por las cajas S_j en la ronda *i*, con $1 \leq j \leq 8$.

Definimos a $\mathbf{K}_i = \{\, E(R_{i-1}) \oplus B^i \mid B^i \in \mathbf{B}^i \,\}$ se cumple que $K_i \in \mathbf{K}_i$ por la expresión 5.1 y $|\, \mathbf{K}_i \,| = 2^{16}$ por la manera en que fue construido $\mathbf{B}^i$.

En este punto queda demostrada la primera parte; para la segunda se procederá, como primer paso, a construir las cadenas $C_0 D_0$ de 56 bits. Si se denomina a los elementos del conjunto $\mathbf{K}_i$ como: $(C_iD_i)_{48,j}$ con $1 \leq j \leq 2^{16}$; entonces para todo $(C_iD_i)_{48,j} \in \mathbf{K}_i$ se aplica la permutación inversa PC-2;esto es, $(\text{PC-2})^{-1}(C_iD_i)_{48,j}$. Nótese que la operación se realiza para 2^{16} elementos.

El resultado de $(\text{PC-2})^{-1}$ $(C_iD_i)_{48,j}$ es una cadena de 48 bits y para llevarla a una cadena de 56 bits se agregan variables binarias en aquellas posiciones que PC-2 eliminó. De hecho, las variables binarias x_1, x_2, x_3, x_4, x_5, x_6, x_7 y x_8 se agregan en las posiciones 9, 18, 22, 25, 35, 38, 43 y 54, tal como se realizó en el capítulo de "MARCO TEÓRICO". Se denotan las cadenas de tamaño 56 como $(C_iD_i)^*_j$. El asterisco señala el hecho de que las cadenas de longitud 56 tienen entre sus bits 8 variables binarias.

Ahora bien, para cada una de las cadenas $(C_iD_i)^*_j$ se recorren las posiciones de sus bits tantas veces, dependiendo del valor de *i* el índice de $\mathbf{K}_i$, en el sentido inverso al de la construcción de las llaves programadas con la finalidad de obtener $(C_0D_0)^*_j$.

Si se aplica la permutación inversa de PC-1 a las cadenas $(C_0D_0)^*_j$; o sea,$(\text{PC-1})^{-1}(C_0D_0)^*_j$ el resultado es una cadena de longitud de 56 bits y para llevarla a una cadena de 64 bits se

agrega una marca, por ejemplo un 8, en las posiciones 8, 16, 24, 32, 40, 48, 56 y 64, que en realidad son las posiciones de los bits de paridad. A las cadenas de 64 bits denotémoslas como K^*_j, recuerde que el subíndice j va de $1 \leq j \leq 2^{16}$.

Para construir al conjunto $\mathbf{K}^*$ se sustituyen valores de 0 ó 1 en las posiciones donde aparecen las variables binarias, lo que nos da 256 posibilidades para cada K^*_j. Entonces $\mathbf{K}^* = \{ K^* \mid$ para todo K^*_j se sustituyen las variables binarias por valores de 0 ó 1, 256 posibilidades en total$\}$.

Se desprende fácilmente que la llave desconocida $K \in \mathbf{K}^*$ y $|\mathbf{K}^*| = 2^{24}$

Después del Teorema LR queda claro que si se desea localizar una llave desconocida $K \in \mathbf{K}$, es necesario un trozo de texto claro y su correspondiente texto encriptado, mínimo de 64 bits. Lo anterior, con objeto de averiguar cuál de las llaves de $\mathbf{K}^*$ aplicada al texto claro da como resultado el texto cifrado, lo que conduce al siguiente corolario.

Corolario 5.1.- Dadas las cadenas L_{i-1}, R_{i-1} y R_i que se utilizan en la i-ésima ronda de un proceso de encriptado DES. Entonces encontrar una llave desconocida $K \in \mathbf{K}$ es equivalente a resolver un problema de tamaño 2^{24} a lo sumo.

La demostración es una consecuencia inmediata del Teorema LR.

Ahora se está en posición de resolver la pregunta que se planteó al principio del capítulo ¿qué información adicional se requerirá para romper DES?. Claro está, se quiere que sea la mínima posible.

Se desprende fácilmente del corolario 5.1 que si $i = 1$, entonces L_{i-1} y R_{i-1} serían L_0, R_0 los cuales se obtienen de manera sencilla del texto claro. Por lo tanto, solamente se requerirá conocer a R_1, 32 bits. De otro modo si $i = 16$, entonces L_i y R_i serían L_{16} y R_{16} los cuales se pueden conocer fácilmente del texto cifrado. De aquí se sigue que únicamente será necesario conocer a L_{15}, 32 bits.

Resumiendo, el conocimiento del bloque R_1 o L_{15} y usando el corolario 5.1, permite reducir la búsqueda de la llave desconocida $K \in \mathbf{K}$, que es de tamaño 2^{56}, por el conjunto $\mathbf{K}^*$ de tamaño 2^{24}. Esto último, puede solucionarse en una computadora Pentium IV comercial en minutos. De hecho, no es necesario conocer los 32 bits de R_1 o L_{15}; ya que si sólo se conocen a 24 de ellos, de cualquiera de los dos bloques, también resuelve el problema.

En este orden de ideas, de inmediato surge la pregunta ¿cómo impacta este resultado a Triple-DES?. Si en el DES sencillo es necesario conocer a un bloque de al menos 32 bits, ¿en Triple-DES será necesario conocer al menos 3 bloques de 32 bits?

Antes de resolver estas últimas interrogantes, se mostrará un resultado empírico que el autor descubrió.

Se mencionó anteriormente que para localizar la llave desconocida $K \in \mathbf{K}$ bastaba con probar las llaves de $\mathbf{K}^*$ y averiguar cuál de ellas da como resultado el texto encriptado. Sin embargo no es necesario llegar hasta el texto cifrado, puesto que puede utilizarse un bloque derecho de una ronda anterior a la 16 como texto cifrado; por ejemplo R_4. Entonces únicamente será necesario conocer a R_1, R_4 además del trozo de texto claro para solucionar el problema.

El resultado anterior es toral para romper Triple-DES, dado que si se conocen a R_1, R_4 del primer ciclo y L_{15} del segundo ciclo, el problema de encontrar dos llaves desconocidas K^1, $K^2 \in \mathbf{K}$ puede transformarse en un problema de complejidad a lo sumo 2^{25}. ¿Cómo se actuaría entonces para romper Triple-DES con esta información?.

El procedimiento sería el siguiente:

a) Utilizando al texto claro y R_1 se puede construir a $\mathbf{K}^{*,1}$ de tamaño 2^{24} en donde se sabe se encuentra K^1, la primera llave con que se cifra en el primer ciclo de Triple-DES.

b) Se prueban las llaves de $\mathbf{K}^{*,1}$ tomando como texto cifrado a R_4.

c) Habiendo localizado la llave K^1 ésta se utiliza para desencriptar el texto cifrado del tercer ciclo, dando como resultado el texto claro del segundo ciclo. Llamémosle al texto claro del segundo ciclo PTK^1. Se desprende fácilmente que si se aplica la llave K^1 al texto claro del primer ciclo se obtiene el texto cifrado del primer ciclo. El cual es a su vez el texto cifrado del segundo ciclo, llamémosle CTK^1.

d) Conociendo a PTK^1, CTK^1 y L_{15} del segundo ciclo, se puede aplicar el corolario 5.1 para obtener la llave K^2; claro está, para conseguirlo se construye a $\mathbf{K}^{*,2}$.

La complejidad del problema se deduce de los incisos a y d; ya que la búsqueda de K^1, K^2 se restringe a localizarlas en el conjunto $\mathbf{K}^{*,1} \cup \mathbf{K}^{*,2}$. Por lo que dicha búsqueda es a lo sumo de $2^{24} + 2^{24} = 2^{25}$. La siguiente figura podrá aclarar lo anteriormente expuesto:

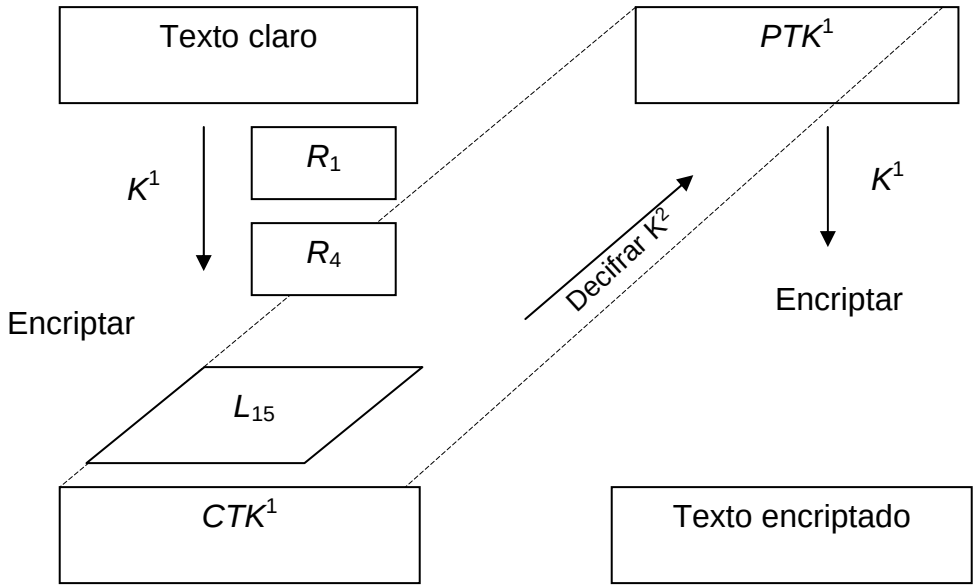


Figura 5.1

Como puede observarse, los procedimientos anteriormente expuestos ponen de manifiesto que la debilidad de DES en ambas modalidades, DES simple y Triple-DES, son sus llaves programadas. De alguna manera esta debilidad es también señalada por el criptoanálisis diferencial y lineal [6,12]. Las soluciones para ambos casos DES y Triple DES serán expuestas en la sección 5.3. A continuación se hará un bosquejo de la solución que es aplicable a DES y Triple-DES:

El Teorema LR deja claro que pasar las cajas en el sentido inverso genera un conjunto de tamaño 2^{16}. Entonces no es extraño proponer la aplicación de dos rondas a las llaves programadas con lo cual se logra incrementar la complejidad, 2^{32}, cuando se va en dirección inversa; obviamente en estas rondas no habría otra llave de por medio. Por último, si utilizando el Teorema LR se localiza una llave programada a la que le fueron aplicadas dos rondas. Entonces encontrar la llave $K \in \mathbf{K}$ tendría una complejidad de $2^{24} * 2^{32} = 2^{56}$, que es la complejidad inicial de DES.

Es importante señalar que el procedimiento anteriormente expuesto de aplicar dos rondas a las llaves programadas, se haría utilizando las mismas cajas y la permutación de expansión del criptosistema DES. Para Triple-DES la solución será similar, ya que si para un DES simple son necesarias 2 rondas, parece lógico pensar que para Triple-DES serán necesarias 6 rondas a las llaves programadas.

5.2 Experimentos de la técnica de criptoanálisis propuesta.

Se inicia con la descripción de la complejidad o tamaño del problema a resolver utilizando la notación big *O* [20]; esto es, el valor máximo del número de llaves utilizadas en la búsqueda de K^1 y K^2 en un ataque a Triple-DES como el descrito en este trabajo. También, una cota para la variable tiempo la cual dependerá de un parámetro que es a su vez función del procesador y la plataforma con que se esté trabajando.

Como se mencionó en el capítulo de "DESARROLLO", el tamaño del problema para romper un criptosistema Triple-DES es de 2^{25}; claro está, bajo la suposición de que se conocen a R_1, R_4 del primer ciclo y L_{15} del segundo ciclo. En este orden de ideas, si se considera a *n* como el número de llaves que se verifican para localizar a K^1 y K^2, entonces $n = O(2^{25}+1)$. Para la variable tiempo se tendría de manera lógica que $t = O(\tau\ 2^{25} + \varepsilon)$ donde τ es el tiempo promedio para verificar una llave y $0 < \varepsilon < 1$.

A manera de ilustración a continuación se muestra por medio de 8 tablas los rompimientos de los criptosistemas DES y Triple-DES con textos claros de palabras conocidas. En estas tablas se expresan la llave o llaves desconocidas y los bloques que intervienen en sistema hexadecimal. Además, el número de segundos que se tarda en encontrar la llave o llaves en cuestión. Es importante también mencionar que los programas se escribieron en lenguaje C++ , y la máquina que se utilizó fue una Pentium IV de 2.0 GHz.

Se inicia la exposición del caso en el que se conocen los siguientes bloques: Texto claro y cifrado de 64 bits; además, el bloque izquierdo de la ronda 15 de 32 bits.

Tabla 6.1 Rompimiento de DES conociendo L_{15}

LLAVE	TEXTO CLARO	TEXTO CIFRADO	L_{15}	TIEMPO EN SEG.
028382ª38382838F	HUBERTHA	874DC1DABB3E59BB	8022D195	556
	MAURICIO	5D802D8CE013255F	4C201FF4	9
	CORNELIO	B3BE9B7EEE2FCA3F	2DA05330	803
	RENTERIA	6FEA2C29E122E1CA	7BA55F30	716

Tabla 6.2 Rompimiento de DES conociendo L_{15}

LLAVE	TEXTO CLARO	TEXTO CIFRADO	L_{15}	TIEMPO EN SEG.
0F239233EB6C0D72	HUBERTHA	028FF46FCBCBFAB8	133B7B63	682
	MAURICIO	FC2065691EABBB82	2FEDA502	17
	CORNELIO	B6BDA571CC3A1EE9	EC2F9563	652
	RENTERIA	7D62624510343377	F0B27CD3	88

Como puede observarse de las dos tablas anteriores, para la primera llave y los textos claros HUBERTHA, MAURICIO los tiempos de rompimiento son menores respecto a los tiempos

utilizados en la segunda llave con los mismos textos claros, sin embargo, esta misma situación se invierte cuando los textos claros son CORNELIO, RENTERIA.

En realidad lo que esto muestra, es que los tiempos de rompimiento dependen de la llave y del texto claro que en particular se estén utilizando; sin embargo, en general nosotros podemos afirmar lo siguiente:

a) El algoritmo puede correr con suerte y la primera llave que se pruebe del conjunto $\mathbf{K}^*$ sea la que se busca, o
b) Se puede tener mala suerte y la última llave que se pruebe sea la que se busca.

Cuando se conoce el bloque derecho de la primera ronda se obtienen los siguientes resultados:

Tabla 6.3 Rompimiento de DES conociendo R_1

LLAVE	TEXTO CLARO	TEXTO CIFRADO	R_1	TIEMPO EN SEG.
028382ª38382838F	HUBERTHA	874DC1DABB3E59BB	63FA8578	335
	MAURICIO	5D802D8CE013255F	C1E21AA5	332
	CORNELIO	B3BE9B7EEE2FCA3F	D8F80F0A	775
	RENTERIA	6FEA2C29E122E1CA	6BF39722	778

Tabla 6.4 Rompimiento de DES conociendo R_1

LLAVE	TEXTO CLARO	TEXTO CIFRADO	R_1	TIEMPO EN SEG.
0F239233EB6C0D72	HUBERTHA	028FF46FCBCBFAB8	548CEA55	407
	MAURICIO	FC2065691EABBB82	E5B269C8	411
	CORNELIO	B6BDA571CC3A1EE9	C43ED645	853
	RENTERIA	7D62624510343377	46077E0C	847

Si se toma a R_4 como texto encriptado y se conoce al bloque R_1, entonces los resultados son los siguientes:

Tabla 6.5 Rompimiento de DES conociendo R_1 y R_4

LLAVE	TEXTO CLARO	TEXTO CIFRADO R_4	R_1	TIEMPO EN SEG.
028382ª38382838F	HUBERTHA	F40AFB2D	63FA8578	96
	MAURICIO	1ª5CA03E	C1E21AA5	94
	CORNELIO	53FFD52D	D8F80F0A	220
	RENTERIA	654848DE	6BF39722	221

Tabla 6.6 Rompimiento de DES conociendo R_1 y R_4

LLAVE	TEXTO CLARO	TEXTO CIFRADO R_4	R_1	TIEMPO EN SEG.
0F239233EB6C0D72	HUBERTHA	D644F5EB	548CEA55	116
	MAURICIO	A06125F5	E5B269C8	117
	CORNELIO	69D00F02	C43ED645	244
	RENTERIA	B75906D6	46077E0C	242

Como puede observarse de las tablas anteriores; los tiempos se reducen de manera significativa en relación con las tablas de más arriba. La razón de estos resultados se debe a que se calculan únicamente 4 llaves programadas o llaves ronda, en lugar de 16, cuando se están probando las llaves del conjunto $\mathbf{K}_1$.

De acuerdo con lo descrito en el capítulo de, "DESARROLLO", a continuación se presentan los resultados de un rompimiento de Triple-DES cuando se conocen los bloques R_1, R_4 del primer ciclo y el bloque L_{15} del segundo ciclo.

Tabla 6.7 Rompimiento de Triple-DES conociendo R_1, R_4 y L_{15}

LLAVES	TEXTO CLARO	TEXTO CFRADO	PRIMER CICLO R_1	PRIMER CICLO R_4	SEGUNDO CICLO L_{15}	TIEMPO EN SEG.
028382A38382838F 0F239233EB6C0D72	HUBERTHA	A2D4F98CCE573C63	63FA8578	F40AFB2D	3CA4C7F6	773
	MAURICIO	0678268B8D9BC1ED	C1E21AA5	1A5CA03E	7C3B0E9A	103
	CORNELIO	E2A9A68665139860	D8F80F0A	53FFD52C	3D4F7E40	501
	RENTERIA	3A739AE170E3FF08	6BF39722	654848DE	52A1AEBA	224

Tabla 6.8 Rompimiento de Triple-DES conociendo R_1, R_4 y L_{15}

LLAVES	TEXTO CLARO	TEXTO CFRADO	PRIMER CICLO R_1	PRIMER CICLO R_4	SEGUNDO CICLO L_{15}	TIEMPO EN SEG.
834312C383A2434E 331182AB891E4347	HUBERTHA	51E696166F6BB1C3	E533962F	F629EACA	35F4198E	974
	MAURICIO	0D9C702545E5EDAD	473B0DF2	27D3888A	A8DB8889	559
	CORNELIO	0FCB54FEBCFE11B6	5EA1185E	8821C049	43822870	820
	RENTERIA	FD814AF8853D6D40	EDAA8077	B0B8DD77	90D5400F	587

Ahora tratemos una situación un poco diferente: considérese que se tiene el cifrado del siguiente trozo de poema como texto claro:

Siembro una rosa blanca

en julio como en enero

para el amigo sincero

que me da su mano franca

Supóngase también, que del texto claro anterior se conocen los 94 caracteres de su texto cifrado. Sin embargo, únicamente para el bloque "una rosa" se conoce los bloques R_1, R_4 del primer ciclo y L_{15} del segundo ciclo. De tal forma que se puede realizar un ataque a Triple DES con esta información. A continuación se dan los valores particulares para el texto cifrado y los bloques R_1, R_4 y L_{15}:

TEXTO CLARO = una rosa
TEXTO CIFRADO= $53B614006810D1CF_{16}$
R_1= $1385D87A_{16}$
R_4= $B03AC402_{16}$
L_{15}= $77AE3E04_{16}$

Con esta información se obtienen los siguientes valores para las llaves:

K^1= $038283A38283828E_{16}$
K^2= $89129332EA6D0C72_{16}$

El tiempo que consumió el computador descrito más arriba es de 854 seg.

Por último, se muestra a continuación una tabla donde se comparan los ataques a DES descritos en el capítulo "TÉCNICAS DE CRIPTOANÁLISIS" con el que aquí se propone:

Tabla 6.9 Análisis comparativo

	Textos Claros	Textos Cifrados	Bloque R_1 o L_{15}
Criptoanálisis Diferencial	2^{47}	2^{47}	0
Criptoanálisis Lineal	2^{44}	2^{44}	0
Ataque Mejorado de Davies	2^{50}	2^{50}	0
Teorema LR	2	2	24 bits

Se señala que los ataques del criptoanálisis diferencial y lineal se utilizan máquinas del tipo "Power 5" de IBM (Consultar página de IBM), que son muy superiores a la Pentium IV que se utilizó en los ataques anteriores usando el Teorema LR. Sin embargo, los criptoanálisis diferencial y lineal no solicitan información de los bloques R_1 o L_{15}. Lo que el autor desea poner de manifiesto es lo siguiente: los bloques R_1 ó L_{15} son tan débiles, además de las llaves programadas, que el conocimiento de una parte de ellos, 24 bits, nos conduce a realizar un ataque con una Pentium IV. Así también, es posible proponer un ataque a Triple-DES utilizando la misma Pentium IV a partir de esta debilidad. El autor no ha encontrado, hasta ahora, alguna referencia de ataque a Triple-DES.

El ataque "Balance Tiempo – Memoria" no se presenta en esta tabla, ya que este ataque procede de manera diferente a los mencionados en ella; esto es, no tiene como objetivo conocer alguna de las llaves programadas para posteriormente encontrar la llave, sino que este procedimiento trata de localizar directamente a la llave, a partir de proponer 2^{19} cadenas de 56 bits y 2^{19} funciones de reducción diferentes [6].

5.3 Algoritmo de doble ronda.

El Corolario 5.1 deja al descubierto 16 posibilidades para encontrar la llave desconocida del criptosistema DES; esto es lo que el autor denomina 16 puertas traseras. Esta debilidad afecta también a Triple-DES; pues como se vio anteriormente al menos se puede construir una puerta, con la que se rompe Triple-DES en minutos utilizando una Pentium IV comercial.

Por fortuna hay una solución sencilla para proteger a DES y Triple DES; con esto se quiere decir que ante un ataque como el descrito en este trabajo se puede conservar una complejidad de 2^{56} para DES y 2^{112} para Triple DES.

Se le llamará a la solución para el DES simple como: "ALGORITMO DE DOBLE RONDA", porque se propondrá una modificación del algoritmo DES en la parte donde se generan las llaves programadas o llaves ronda, y esto por la debilidad del criptosistema DES en sus llaves programadas.

A continuación se explicará de manera concreta en qué consiste el algoritmo de doble ronda. Es claro que si se cruzan las cajas en sentido contrario a la del algoritmo DES, se construye un conjunto solución de 2^{16} elementos de acuerdo con el Teorema LR. Entonces parece razonable proponer que se apliquen dos rondas a las llaves programadas de la siguiente manera:

Suponga que se obtuvieron las 16 llaves programadas K_1, K_2,…., K_{16} con base en el algoritmo del criptosistema DES, utilizando una llave K de 64 bits. A partir de aquí cada una de ellas, dado que son cadenas de 48 bits, se pueden pasar por las cajas, dando como resultado cadenas de 32 bits. Entonces, para volverlas a pasar nuevamente por las cajas en una segunda ronda, será necesario aplicar la tabla de expansión-permutación E, ya que ésta da como resultado una cadena de 48 bits. Finalmente, como el resultado de esta segunda ronda es una cadena de 32 bits y se requieren de llaves de 48 bits, se sigue que será necesario aplicar nuevamente la tabla de expansión-permutación E.

Nótese que en la salida de las cajas no se aplicó la permutación P, sino la permutación-expansión E. También es importante señalar que al pasar por las cajas a cualquier llave programada, se hace sin aplicar la operación "xor" con otra cadena de 48 bits, dado que esto significaría tener una llave para la llave lo cual no tiene sentido.

El objetivo de aplicar dos rondas a las llaves programadas es evitar la posibilidad de un ataque al criptosistema, "ALGORITMO DE DOBLE RONDA", utilizando el Teorema LR. Pues en caso de ser así, se construye un conjunto de $2^{24} * 2^{32} = 2^{56}$ elementos, con lo cual se conserva la complejidad de DES. Con objeto de ilustrar este procedimiento, puesto que es muy importante para este trabajo, a continuación se hará una representación gráfica del mismo:

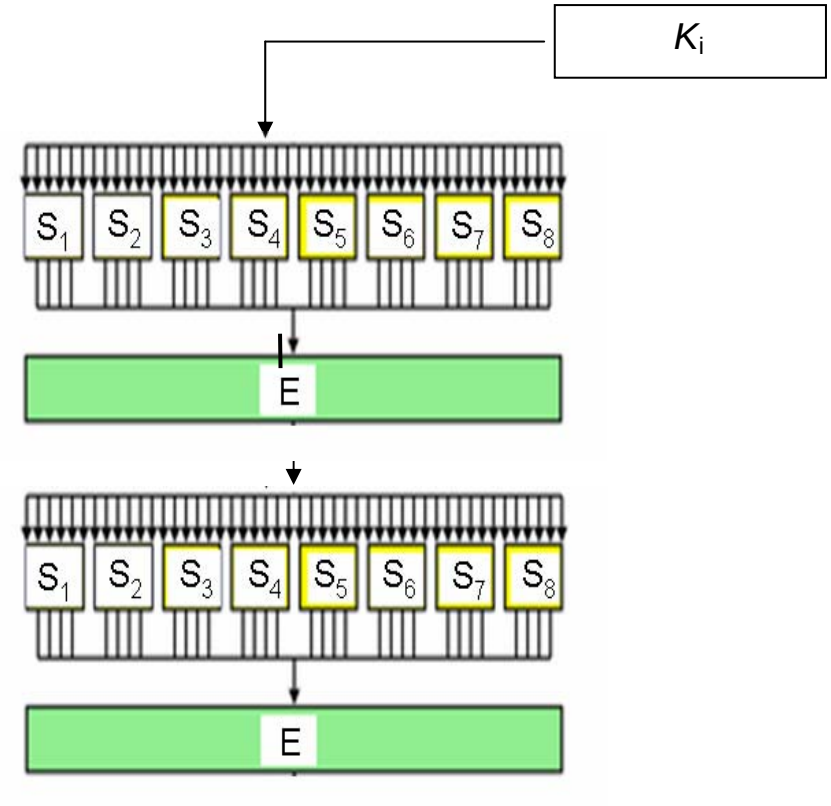


Figura 7.1 Gráfica de la obtención de las llaves programadas de doble ronda

En esta gráfica se representa un ciclo de doble ronda para una llave programada K_i , con $1 \leq i \leq 16$. Para terminar con la descripción del "ALGORITMO DE DOBLE RONDA", se dirá que una vez obtenidas las llaves programadas o llaves ronda de acuerdo al procedimiento descrito anteriormente, el algoritmo de doble ronda procede de la misma forma como lo hace el algoritmo DES. Se desprende fácilmente que este proceso prácticamente no afecta los tiempos de ejecución; dado que las llaves programadas se calculan una sola vez y de ahí en adelante la forma de cifrar los textos claros se lleva a cabo como el algoritmo DES.

CAPITULO 6

CONCLUSIONES Y TRABAJO FUTURO

Después de lo expuesto en el capítulo anterior se concluye que: **la debilidad de DES y Triple-DES descubierta en este trabajo son sus llaves programadas**. Esto se debe al algoritmo que las genera porque lo realiza de manera lineal; esto es, que las permutaciones y corrimientos que se aplican son funciones lineales. Aunque actualmente DES no es el estándar internacional, Triple-DES está aún vigente. Por lo tanto, es importante la solución que se propuso para DES en el capítulo 6, debido a que ésta se puede extender para Triple-DES. **De hecho, el "ALGORITMO DE DOBLE RONDA" elimina la linealidad en la generación de las llaves programadas.**

La solución para Triple-DES se le llamará "ALGORITMO DE SEIS RONDAS". Se desprende fácilmente que se proponga como solución para Triple-DES seis rondas en las llaves programadas. Esto último, nos garantiza que si se realiza un ataque como el presentado en el capítulo 5, la complejidad sería de $2^{25} * 2^{96} = 2^{121}$. Por lo tanto, esto hace que se conserve la complejidad de Triple-DES.

En el anexo de esta investigación se presenta la norma para el criptosistema de "DOBLE RONDA"

Por último, se menciona que el autor está trabajando en este momento en la creación de nuevos criptosistemas que compitan en complejidad y rapidez con Advance Encrytion Standard (AES) [12]. De hecho, se trabaja en un criptosistema que tiene una complejidad computacional de 2^{500} [36].

ANEXO

La norma internacional para el criptosistema DES establece como patrones de texto claro y de llave a: $123456789ABCDEF_{16}$ y $133457799BBCDFF1_{16}$ correspondientemente [6,7]. A partir de estas dos cadenas la norma construye ronda a ronda, 16 en total, las llaves programadas, los bloques izquierdo y derecho, la función $f(R_i, K_i)$ y algunas otras cadenas de menor importancia [6,7].

Siguiendo este procedimiento a continuación se muestra para cada ronda del: "ALGORITMO DE DOBLE RONDA", las cadenas que intervienen en él.

Llave: $133457799BBCDFF1_{16}$
Texto claro: $0123456789ABCDEF_{16}$

L_0=11001100000000000110011001111111
L_1=R_0= 11110000101010101111000010101010

Primera Ronda

$E(R_0)$= 011110100001010101010101011110100001010101010101
K_1= 001000001000001100000010100010100101010101011100
$E(R_0) \oplus K_1$=010110101001011001010111111100000100000000001001
Salida de
las cajas= 110000111100110000001010010010 10
$f(R_0, K_1)$= 01011000101100111101000101000000
L_2=R_1= 10010100101100110001110110111111

Segunda Ronda

$E(R_1)$= 11001010100101011010011010001111101111010111111111
K_2= 01000000000000011010110101110000111111001111100 1
$E(R_1) \oplus K_2$=10001010100101000000101111111111010000010000011 0
Salida de
las cajas = 00010011000111110011010000100100
$f(R_1, K_2)$= 11100100010000000101010011101010
L_3=R_2= 00010100111010101010010001000000

Tercera Ronda

$E(R_2)$= 000010101001011101010101010000100000100000000
K_3= 101011110110101101011010101011111100000100000010
$E(R_2) \oplus K_3$=101001011111110000001111111111110100001100000010
Salida de
las cajas= 010001011011001100110100100000010
$f(R_2, K_3)$= 10100100010000101100000110011101
$L_4=R_3$= 00110000111100011101110000100010

Cuarta Ronda

$E(R_3)$= 000110100001011110100011111011111000000100000100
K_4= 001000001101011101010111111101010000000110101100
$E(R_3) \oplus K_4$=001110101100000011110100000110101000000010101000
Salida de
las cajas= 100011010111001100010010101110001
$f(R_3, K_4)$= 10101110111010010100110000010101
$L_5=R_4$= 10111010000000111110100001010101

Quinta Ronda

$E(R_4)$= 11011111010000000000011111110101000000101010101 1
K_5= 011111111110101111110111111011010011010010101 1001
$E(R_4) \oplus K_5$=101000001010101111110000001011110110101111110010
Salida de
las cajas= 110110110111111101111010011101 10
$f(R_4, K_5)$= 11110110111111111101010011100110
$L_6=R_5$= 11000110000011100000100011000100

Sexta Ronda

$E(R_6)$= 110111111011110110101111111001011010100100001111
K_7= 101001010001010111111101011010100101011111111110
$E(R_6) \oplus K_7$=011110101010100001010010100011111111111011110001
Salida de
las cajas= 011101000001001010001101001011111
$f(R_6, K_7)$= 000111010100001010101110000111010
$L_8=R_7$= 110110110100110010100110111111110

Séptima Ronda

$E(R_5)$= 011000001100000001011100000001010001011000001001
K_6= 101100000010101010100000000011111111110100000110
$E(R_5) \oplus K_6$=110100001110101011111100000010101110101100001111
Salida de
las cajas= 100101001001100011000011011101000
$f(R_5, K_6)$= 000001111011010000100101011100 10
$L_7=R_6$= 101111011011011111001101001001111

Octava Ronda

$E(R_7)$= 011011110110101001011001010100001101011111111101
K_8= 000010100001010111110111110011111101011101011000
$E(R_7) \oplus K_8$=011001010111111110101110100111110000000010100101
Salida de
las cajas= 100110100111110101110111101111 10
$f(R_7, K_8)$= 111011101010111100110100111011111
$L_9=R_8$= 010100110001100011111001110010000

Novena Ronda

$E(R_8)$= 00101010011010001111000101111111001111001010000
K_9= 10010000101010101011000000101010000010010101 10
$E(R_8) \oplus K_9$=10111010110000100101110101110101101111 0000000110
Salida de
las cajas= 10111101001111101000101110100100
$f(R_{8,}K_9)$= 00010101111010000111011001110111
$L_{10}=R_9$= 11001110101001001101000010001001

Décima Ronda

$E(R_9)$= 11100101110101010000100101101010000101000101001 1
K_{10}= 01101111011010011111111010110101011111011010 1101
$E(R_9) \oplus K_{10}$=10001010101111001111011111011111011010011111111 0
Salida de
las cajas= 00011111111111011100110101000 1000
$f(R_{9,}K_{10})$= 11111101011010010100000101010111
$L_{11}=R_{10}$= 10101110011100011011100010011111

Décima Primera Ronda

$E(R_{10})$= 11010101110000111010001111011111000101001111111 1
K_{11}= 11101111101010101010011111111111110101010111100 11
$E(R_{10}) \oplus K_{11}$=00111010011010010000010000100000101111100000110 0
Salida de
las cajas= 10001011010011100111110000001011
$f(R_{10,}K_{11})$= 01111000110011110101100011001000
$L_{12}=R_{11}$= 10110110011010111000100001000001

Décima Segunda Ronda

$E(R_{11})$= 11011010110000110101011111000101000000100000011
K_{12}= 00100000111111010100011110101010110101110101100
$E(R_{11}) \oplus K_{12}$=11111010001111011111010000010000011010011010111
Salida de
las cajas= 00001000001100110100111111011101
$f(R_{11,}K_{12})$= 10011110011111000010100000101101
$L_{13}=R_{12}$= 00110000000011011001000010110010

Décima Tercera Ronda

$E(R_{12})$= 00011010000000000101101111001010000101010110100100
K_{13}= 11111010100000010000110101110000010000100000011
$E(R_{12}) \oplus K_{13}$=11100000100000010101011010111010010101111010011 1
Salida de
las cajas= 00110110000001011000001000010111
$f(R_{12,}K_{13})$= 11000011001000100001101000110010
$L_{14}=R_{13}$= 01110101010010011001001001110011

Décima Cuarta Ronda

$E(R_{13})$= 10111010101010100101001111001010010000111010011 0
K_{14}= 01110101001010100101110000100101111010111111 10001
$E(R_{13}) \oplus K_{14}$=11001111100000000000111111101111101010000101011 1
Salida de
las cajas= 10111001101000110100110101101011
$f(R_{13,}K_{14})$= 10011000110111100110111100001110
$L_{15}=R_{14}$= 10101000110100111111111110111100

Décima Quinta Ronda

$E(R_{14})$= 010101010001011010100111111111111111110111111001
K_{15}= 010111111010100100001111111000000110101011110001
$E(R_{14}) \oplus K_{15}$=000010101011111110101000000111111001011100001000
Salida de
las cajas= 010011110111110011000110011001l0
$f(R_{14,}K_{15})$= 010001010011111111010100011111l00
$L_{16}=R_{15}$= 00110000011101100100011000001111

Décima Sexta Ronda

$E(R_{15})$= 100110100000001110101100001000001100000001011110
K_{16}= 100111110001011001010010101010100100000010101010
$E(R_{15}) \oplus K_{16}$= 000001010001010111111110100010101000000011110100
Salida de
las cajas= 000011001110010000100010000l1010
$f(R_{15,}K_{16})$= 000010000010101100010001l0010100
R_{16}= 10100000111110001110111000101000

$(IP)^{-1}$ $(R_{16}$ $L_{16})$

= 0000001000101110001011100001011110110000111101010011110001010100

$(IP)^{-1}$ $(R_{16}$ $L_{16})$= $022E2E17B0F53C54_{16}$

Referencias

[1] Helfman M.E., 1980, "A cryptanalytic time–memory trade-off". *IEEE Transactions on Information Theory*, pp 401-406.

[2] Fiat A. and Naor M., 1991, "Rigorous time/space trade-off, for inverting functions", *In Proceedings of the 23rd symposium on the Theory of Computing*, pp. 534-541, ACM Press.

[3] Biham E. and Shamir A., 1993, "Differential cryptanalysis of the full 16-round DES", *Lecture Notes in Computer Science*, pp 494-502.

[4] Eberle H., "A High-speed DES implementation for network applications", *Lecturer Notes in computer Science*, pp 527-545.

[5] www.cl.com.ac.uk/user's/irnc1/descrock/reaction.htm.2500.

[6] Douglas R. Stinson, 1995, *CRYPTOGRAPHY: Theory and practice*, CRC Press, pp. 70-113.

[7] Grabbe J. Orlin, 2003, "Data Encryption Standard: The DES algorithm illustrated", *Laissez faire City time*, vol. 2, no 28.

[8] Menezes Alfred J., van Oorschot Paul, and Vanstone Scott, 1997, *Handbook of Applied Cryptography*, CRC Press, Boca Raton.

[9] Rosen K., 2003, *Discrete Mathematics and its Applications*, Mc. Graw Hill, fifth edition, pp. 301-349.

[10] Fúster Sabater A. et al, 2001, *Técnicas Criptográficas de protección de datos*, Alfaomega 2ª Edición, pp. 5-92.

[11] Carlet C., 2005, "On highly nonlinear S-boxes and their inability to thwart DPA attacks", *6th International Conference on Cryptology of the Springer-Verlag*, pp. 49-62

[12] Douglas R. Stinson, 2002, *CRYPTOGRAPHY: Theory and practice*, CHAPMAN & HALL/ CRC Press, second edition, pp. 74-116.

[13] Matsui M, 1994, "Linear Cryptanalysis method for DES cipher", *Lecture Notes in Computer Science*, pp 386-397.

[14] Diffie W. and Hellman M. E., 1977 "Exhaustive Cryptanalysis of the NBS Data Encryption Standard ", *IEEE Computer Society Press*, vol. 10, pp. 74-84

[15] Wiener M., 1993, "Efficient DES key search", *TR-244 Carleton University*. También aparece en : Stallings W., 1996, "Practical Cryptography for Data Internetworks", *IEEE Computer Society Press*, pp. 31-79.

[16] Stalling W, March 2006, *"Encryption Options Beyond DES"*, www.commsdesign.com.

[17] Ritter T, 2006, *"Triple-DES is Proven to be Very Secure?"*, http://www.ciphersbyritter.com/NEWS5/PROVSEC.HTM.

[18] Boneh D., DeMillo R., Lipton R., 2001, On the Importance of Checking Cryptographic Protocols for Faults, *Journal of cryptology, Spring-Verlag*, Vol. 14, No. 2, pp. 101-119.

[19] Green L. R., 1988, *Fundamentals of the Theory of Computation*, Morgan Faufmann, 1ª edición, pp 19-64.

[20] Koblitz M., 1987, *A Course in Number Theory and Cryptography*, Springer-Verlag, pp. 53-80, New York Inc.

[21] Valizadeh A., Sabe M., Sadeghian B., Mehdipour F., Najafi B., 2004, "A High Performance Reconfigurable Implementation of DES-Like Algorithms", *Proceeding IEEE*, pp. 140-143.

[22] Rijmen V., 2004, "Equivalent descriptions for the DES", *Electronics Letters IEEE,* vol. 40, no. 4.

[23] Shaffer T., Glaser A., Franzon P., 2004, "Chip-Package Co-Implementation of a Triple DES Processor", *IEEE Transaction on Advanced Packaging,* vol. 27, no. 1, pp. 194-202.

[24] Link H. and Neumann W., 2005, "Clarifying Obfuscation: Improving the security of White-Box DES", *Proceeding of the International Conference on Information Technology of the IEEE*, pp. 679-684.

[25] Wilson P. and Brown A., 2005, "DES in Four Days using Behavioural Modeling & Synthesis", *Proceeding IEEE*, pp. 82-87.

[26] Sanli M., Zengin F., Urhan O., 2005, "Smart Card Based Pre-paid System Application Using Session Update Key in 3-DES Algorithm", *Proceeding IEEE*, pp. 76-79.

[27] Lee T., Zeien R., Roach A., Robinson P., 2006, "DES decoding Using FPGA and Custom Instructions", *Proceeding of the third International Conference on Information technology of the IEEE*, pp.575-577.

[28] Celikel E., Davidson J., Kern C., 2006, "Parallel Performance of DES in ECB Mode", *Proceedings of the seventh IEEE International Symposium on Computer Networks*, pp. 134-139.

[29] Monnet Y., Renaudin M., Leveugle R., Moitrel F., M'Buwa F., 2006, "Practical Evaluation of Fault Countermeasures on an Asynchronous DES", *Proceeding of the 12th IEEE International On-Line Symposium*, pp.125-130.

[30] Knudsen L. and Mathiassen J., 2001, "A Chosen-Plaintext Linear Attack on DES", *Springer-Verlag,* pp.262-272.

[31] Akkar M.L. and Giraud Ch., 2001, "An Implementation of DES and AES, Secure against Some Attacks", *Springer-Verlag*, pp. 309—318.

[32] Schramm K., Wollinger T., and Paar Ch., 2003, "A New Class of Collision Attacks and Its Application to DES", *International Association for Cryptologic Research,* pp. 206-222.

[33] Chow S., Eisen P., Johnson H. and Oorschot P., 2003, "A White-Box DES Implementation for DRM Applications", *Spriger-Verlag,* pp. 1-15.

[34] Tsunoo Y., Saito T., Suzaki T.,Shigeri M. and Miyauchi., 2003, "Cryptanalysis of the DES Implemented on Computers with Cache", *Springer-Verlag,* pp. 62-76.

[35] Sorking A., 1980, *LUCIFER: A cryptographic algorithm*, Cryptología 8, pp 22-35.

[36] Silva V M et al, 2006, "Construcción de una función biyectiva con dominio en los naturales e imagen en el conjunto de las permutaciones: Una aplicación a la Criptografía", *Publicación en proceso*.

[37] E. Biham and Shamir, 1991, "Diferencial Cryptanalyisis of DES-like cryptosystems", *Journal of Cryptology*, pp 3-72

[38] E. Biham and Biryukov, 1997, An Improved Davis'Attack on DES, *Journal of Cryptology*, Vol. 10, Number 3, pp. 195-205.

[39] Buchmann Johannes A., 2004, *Introduction to Cryptography*, Springer