VNiVERSiDAD D SALAMANCA

Facultad de Ciencias
Grado en Matemáticas

Criptoanálisis diferencial

Trabajo de Fin de Grado

Autora: Esther Luna Sánchez
Tutor: José Ignacio Iglesias Curto

Julio 2023

Universidad de Salamanca
Facultad de Ciencias
Grado en Matemáticas

Criptoanálisis diferencial

Trabajo de Fin de Grado

Autora: Esther Luna Sánchez
Tutor: José Ignacio Iglesias Curto

LUNA SANCHEZ ESTHER - 70922058V

Firmado digitalmente por LUNA SANCHEZ ESTHER - 70922058V
Fecha: 2023.07.05 11:24:15 +02'00'

Firma de la autora

VNiVERSiDAD D SALAMANCA
CAMPUS DE EXCELENCIA INTERNACIONAL

Facvltad D Ciencias
VNiVERSiDAD D SALAMANCA

Certificado de los tutores TFG Grado en Matemáticas

D. José Ignacio Iglesias Curto, profesor del Departamento de Matemáticas de la Universidad de Salamanca,

HACE CONSTAR:

Que el trabajo titulado "*Criptoanálisis diferencial*", que se presenta, ha sido realizado por Dña. Esther Luna Sánchez, con DNI 70922058-V y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Matemáticas en esta Universidad.

Salamanca, a fecha de firma electrónica.

Firmado digitalmente por IGLESIAS CURTO JOSE IGNACIO - 70866916Y
Fecha: 2023.07.05 11:50:17 +02'00'

Fdo.:

A mi familia.
A mi pareja, Álvaro.
A mi tutor, José Ignacio, por hacerme pensar, su paciencia y su dedicación.

Índice general

Introducción

"Superficial complications can be illusory, for they can provide the cryptographer with a false sense of security"
- Bruce Schneier

La criptografía es el estudio de las técnicas que se encargan de proteger la información entre dos partes, evitando que un tercero acceda a ellas. Paralelamente, el criptoanálisis es el estudio de estas técnicas con el fin de obtener la información oculta. El desarrollo de ambas áreas ha sido casi simultáneo, ya que mientras algunos trabajan en ocultar la información, otros se esfuerzan por acceder a ella. El término que engloba a ambos campos es la criptología [27, 19].

En este trabajo nos centraremos en un procedimiento concreto de criptoanálisis, el criptoanálisis diferencial. Este método trata de investigar cómo afecta cambiar algunos bits de los textos planos (textos sin cifrar) a los textos cifrados resultantes. La idea consiste en que dada una diferencia en los textos de entrada, se va a generar, con una alta probabilidad, una determinada diferencia en los textos de salida. Dicha diferencia en la salida puede ser útil para buscar un patrón y, así, hallar la clave [16].

Se considera que el criptoanálisis diferencial tiene un doble origen. Por un lado, IBM desarrolló, en secreto, este criptoanálisis a mediados de los años 70 [9]. Lo hizo a la par que creaban el *Data Encryption Standard* (DES). Este es un algoritmo de encriptación en colaboración con el gobierno de los Estados Unidos, aunque actualmente está en desuso (su variante Triple DES sí se utiliza). Al ser conocedores de este criptoanálisis, hicieron al DES resistente a él.

Por otra parte, este criptoanálisis fue presentado públicamente por Eli Biham y Adi Shamir en la *Annual International Cryptology Conference* (CRYPTO) del año 1990, donde atacaban al DES [4]. Observaron que, aunque era particularmente resistente al criptoanálisis diferencial, algunas modificaciones del cifrado sí lo hacían vulnerable.

A pesar de que el ataque fue más teórico que práctico para el DES, su publicación demostró la debilidad de otros muchos cifrados usados en la época. En la actualidad es considerado para diseñar cifrados. Un ejemplo de ello es el *Advanced Encryption Standard* (AES) (también conocido como Rijndael), algoritmo estandarizado por el *National Institude of Standards and Technology* (NIST) para la encriptación de la información digital [12].

No fue hasta mediados de los años 90 cuando Don Coppersmith, principal autor del DES en IBM, reveló que ya sabían de la existencia de este criptoanálisis [9]. Al colaborar con el gobierno de Estados Unidos, decidieron ocultarlo estratégicamente.

El objetivo general de este trabajo es desarrollar el concepto de criptoanálisis diferencial desde el rigor matemático con ayuda de otros criptoanálisis relevantes. Los

objetivos específicos son:

1. Dar a conocer diferentes tipos de cifrados por bloques.
2. Describir la importancia de las cajas de sustitución en el diseño del cifrado y en el criptoanálisis.
3. Explicar el criptoanálisis lineal para facilitar la compresión del criptoanálisis diferencial.
4. Evidenciar la utilidad de combinar el álgebra y la estadística.

La estructura del documento se divide en cuatro capítulos.

- En el Capítulo 1 se desarrollan seis secciones que constan de una introducción y un repaso a conceptos que se usan en el documento para elaborar el tema principal. En primer lugar se abordan los conceptos básicos de la Teoría de la Información para posteriormente detallar unas nociones generales sobre ataques criptográficos. Por último, se plantea la teoría sobre funciones booleanas necesaria.
- En el Capítulo 2 se describen de forma pormenorizada los dos tipos principales de cifrados por bloques. Además, se concreta con el cifrado PRESENT desde un punto de vista booleano.
- En el Capítulo 3 se expone el criptoanálisis lineal. Se considera, según la literatura científica, como uno de los dos principales ataques, junto con el diferencial. Se describe su funcionamiento, cómo accedemos a la clave y a los aspectos fundamentales de este. También se estudia cómo se ha aplicado en la práctica al PRESENT.
- En el Capítulo 4 se explica detalladamente el criptoanálisis diferencial y se aborda estructuralmente de forma semejante al criptoanálisis lineal pero teniendo en cuenta sus particularidades.

Finalmente, se presentan cinco apéndices que complementan la lectura del documento. Se indica a lo largo del mismo cómo acudir a ellos.

Capítulo 1

Preliminares

1.1. Terminología básica

Empezaremos con conceptos relevantes del área de la criptografía y, por consiguiente, del criptoanálisis [23].

Definición 1.1. *Se llama* ***texto plano*** *a una cadena de caracteres de un alfabeto $\mathcal{A}$ que tienen significado o se pueden interpretar. El conjunto de textos planos se denota como $\mathcal{M}$.*

Definición 1.2. *Se llama* ***texto cifrado*** *a una cadena de caracteres resultado de cifrar un texto plano. Puede estar escrita en un alfabeto $\mathcal{A}' \neq \mathcal{A}$. El conjunto de textos cifrados se denota como $\mathcal{C}$.*

Definición 1.3. *Se llama* ***clave*** *a una cadena de caracteres utilizada para transformar un texto plano en texto cifrado. Puede estar escrita en un alfabeto $\mathcal{A}'' \neq \mathcal{A}, \mathcal{A}'$. El conjunto de claves se denota como $\mathcal{K}$.*

Definición 1.4. *Se llama* ***encriptación*** *a la aplicación*

$$\begin{aligned} \varepsilon : \mathcal{M} \times \mathcal{K} &\longrightarrow \mathcal{C} \\ (m, e) &\longmapsto c \end{aligned}$$

Definición 1.5. *Se llama* ***desencriptación*** *a la aplicación*

$$\begin{aligned} \delta : \mathcal{C} \times \mathcal{K} &\longrightarrow \mathcal{M} \\ (c, d) &\longmapsto m \end{aligned}$$

Definición 1.6. *Se denomina* ***criptosistema*** *a la quíntupla $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \{\varepsilon\}, \{\delta\})$ donde:*

- *$\mathcal{M}$ el conjunto de textos planos que se van a encriptar*
- *$\mathcal{C}$ el conjunto de textos cifrados resultantes de encriptar los mensajes de $\mathcal{M}$*
- *$\mathcal{K}$ el conjunto de claves utilizadas*
- *$\{\varepsilon\}$ el conjunto de las aplicaciones de encriptación que se utilizan para transformar $\mathcal{M}$ en $\mathcal{C}$*

- $\{\delta\}$ *el conjunto de las aplicaciones de desencriptación que se utilizan para transformar* $\mathcal{C}$ *en* $\mathcal{M}$

y cumple que para cada $e \in \mathcal{K}$ *hay una única clave* $d \in \mathcal{K}$ *tal que* $\delta = \varepsilon^{-1}$*, es decir,* $\delta(\varepsilon(m, e), d) = m \quad \forall m \in \mathcal{M}$.

Observación 1.7. En algunos textos de consulta, se utiliza el término cifrado para referirse a la Definición 1.6. Sin embargo, en este documento, el término **cifrado** se refiere específicamente al proceso de encriptación de textos planos mediante la aplicación del algoritmo de encriptación ε. Este proceso conlleva que haya la correspondiente aplicación de desencriptación δ (respecto de ε) y la propiedad que cumplen descrita por la Definición 1.6.

En lo que sigue de trabajo, se empleará el alfabeto $\mathbb{F}_2$. Se especificará explícitamente si se hace uso de una variación o extensión.

La criptografía se divide en dos grandes categorías: *criptografía simétrica o de clave secreta* y *criptografía asimétrica o de clave pública*. La criptografía de clave simétrica se caracteriza por utilizar la misma clave para encriptar que para desencriptar mientras que la criptografía de clave asimétrica utiliza dos claves, una pública y otra privada.

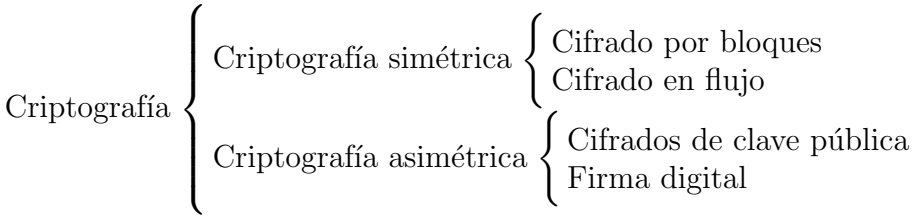

Figura 1.1: División principal de la criptografía

Se deduce que las claves e y d de las Definiciones 1.4 y 1.5 pueden ser iguales o distintas dependiendo del tipo de criptografía mencionada.

Observación 1.8. En criptografía simétrica,

$$\delta(\varepsilon(m, e), d) = m' \quad \text{donde} \quad m = m' \Leftrightarrow e = d$$

1.2. Cifrado por bloques

El tema central de este trabajo es el criptoanálisis diferencial, un ataque aplicado fundamentalmente a cifrados por bloques. Por este motivo, en lo que resta de documento, la teoría estará referida a este tipo de cifrados.

Definición 1.9. *Se llama* ***cifrado por bloques*** *a un tipo de cifrado de criptografía simétrica que consiste en dividir el texto plano en partes de la misma longitud (llamadas* ***bloques****) y encriptar cada uno de ellos.*

Sea un cifrado por bloques de longitud n. Al encriptar un texto plano, obtenemos un texto cifrado de la misma longitud. Por tanto, podemos considerar nuestro espacio de textos planos y textos cifrados como un conjunto de n-tuplas, V_n.

$$\begin{array}{rcl} \varepsilon : V_n \times \mathcal{K} & \longrightarrow & V_n \\ (P, K) & \longmapsto & C \end{array} \qquad \begin{array}{rcl} \delta : V_n \times \mathcal{K} & \longrightarrow & V_n \\ (C, K) & \longmapsto & P \end{array}$$

Sea $K \in \mathcal{K}$ una clave del espacio de claves. Para K fija, la aplicación de encriptación $\varepsilon_K := \varepsilon(\cdot, K)$ es una biyección. En consecuencia, existe una única aplicación de desencriptación $\delta_k := \delta(\cdot, K)$ que se corresponde con su inversa, $\delta_K = \varepsilon_K^{-1}$.

$$\begin{array}{rcl} \varepsilon_K : V_n & \longrightarrow & V_n \\ P & \longmapsto & C \end{array} \qquad \begin{array}{rcl} \delta_K : V_n & \longrightarrow & V_n \\ C & \longmapsto & P \end{array}$$

La gran mayoría de cifrados por bloques son **cifrados iterados**, llamados así porque cifran el texto plano P aplicando una función repetidas veces para obtener el texto cifrado C. Cada iteración se llama **ronda**. Estos cifrados se componen de un algoritmo de generación de claves y funciones de ronda [30, 32].

Consideremos un cifrado por bloques de longitud n de r rondas. Cada ronda se denotará con el subíndice i. Los textos usados en cada iteración se denotarán por z_j con $j \in \{0, \ldots, r\}$. Según esta notación, el texto plano $P = z_0$ y el texto cifrado, $C = z_r$.

Los **algoritmos de generación de claves** son algoritmos conocidos que generan r subclaves k_i a partir de una clave $K \in \mathcal{K}$ (llamada **clave maestra**). Cada ronda i tiene una subclave k_i diferente. La longitud de la clave maestra K y de las subclaves k_i puede diferir.

Las **funciones de ronda** f son las funciones que se aplican sucesivamente al texto de entrada P para encriptarlo. Tienen dos valores de entrada: la subclave k_i y el texto de salida de la iteración anterior, z_{i-1}.

$$\begin{array}{rcl} f : K \times V_n & \longrightarrow & V_n \\ (k_i, z_{i-1}) & \longmapsto & z_i \end{array}$$

El proceso que sigue la función de ronda f para encriptar el texto z_{i-1} es el mismo en todas las rondas con la excepción de la subclave k_i (se recuerda que es diferente $\forall i$). Como consecuencia, se considera que la aplicación de encriptación ε_K está formada por las r funciones de ronda f.

Si en cada iteración i se fija la subclave k_i, entonces las f son inyectivas. Denotemos por $f_i \equiv f(k_i, \cdot)$ a la función de ronda de la iteración i que utiliza la subclave k_i. Entonces, es biyectiva y su inversa permite la desencriptación. Por tanto, para cada iteración i:

$$\begin{array}{rcl} f_i : V_n & \longrightarrow & V_n \\ z_{i-1} & \longmapsto & z_i \end{array} \qquad \begin{array}{rcl} (f_i)^{-1} : V_n & \longrightarrow & V_n \\ z_i & \longmapsto & z_{i-1} \end{array}$$

En definitiva, el cifrado por bloques se puede describir como:

$$z_0 = P \xrightarrow{f_1} z_1 \xrightarrow{f_2} z_2 \xrightarrow{f_3} \cdots \xrightarrow{f_{n-1}} z_{n-1} \xrightarrow{f_n} z_n = C$$

Figura 1.2: Encriptación de un cifrado por bloques

1.3. Tipos de ataques

Como se ha explicado previamente, el criptoanálisis es el área de la criptología que se encarga de atacar los cifrados para descifrar los mensajes. El atacante tiene la intención de acceder a la información y, adicionalmente, corromperla. Su capacidad para ello puede variar pero siempre se ha de considerar el llamado **principio de Kerckhoffs**[1]: el adversario conoce el funcionamiento del criptosistema.

Partiendo de esta base, existen varias clasificaciones de ataques. Se puede clasificar un ataque en función del objetivo del atacante: recuperar información o la clave, entre otros. En este documento, utilizaremos una clasificación de los ataques según la información de la que dispone el adversario. Se enumeran comenzando por el ataque más práctico a los ataques más hipotéticos:

1. **Ataque de texto cifrado** (*ciphertext-only attack*, COA): se observan los textos cifrados para deducir la clave. Se considera que un cifrado es completamente inseguro si es vulnerable a este ataque.

2. **Ataque de texto plano conocido** (*known-plaintext attack*, KPA): se tienen varios textos planos y sus correspondientes textos cifrados.

3. **Ataque de texto plano elegido** (*chosen-plaintext attack*, CPA): se eligen textos planos y se obtienen los correspondientes textos cifrados.

4. **Ataque de texto cifrado elegido** (*chosen-ciphertext attack*, CCA): se eligen textos cifrados y se obtienen los correspondientes textos planos.

5. **Ataques adaptativos de textos elegidos**: son aquellos en los que se eligen los textos, planos o cifrados, dependiendo de las anteriores elegidos u obtenidos.

Observación 1.10. Se deduce que si un cifrado es resistente a un ataque de tipo CPA, entonces es resistente a un ataque de tipo KPA y COA.

1.4. Parámetros de un ataque

El éxito al aplicar un ataque a un cifrado es una cuestión central en el mundo de la criptografía. Dicho éxito está medido según los recursos que se necesitan para organizar el ataque:

1. **Información**: cantidad de datos que se requieren para realizar el ataque.

[1]Claude Shannon reformuló este principio como "*El enemigo conoce el sistema general que se está utilizando*"[32].

2. **Tiempo**: cantidad de tiempo que se requiere para procesar el ataque computacionalmente. Se suele medir en ciclos CPU (tiempo que tarda la CPU en procesar en lenguaje máquina las instrucciones) o en términos de operaciones necesarias (frecuentemente, encriptaciones) [18].

3. **Memoria**: cantidad de memoria o almacenamiento necesario que se requiere para procesar el ataque computacionalmente. Se puede medir en unidades de bytes pero en ataques teóricos se mide por la cantidad de bloques que tiene un cifrado.

En el mundo de la criptografía se evalúan primero los ataques teóricamente y luego se determina si, con la tecnología disponible del momento, pueden llevarse a cabo. En otras palabras, aunque los recursos necesarios para organizar un ataque fuesen muy altos, no hay que susbestimar la capacidad de ciertos ordenadores que podrían ejecutarlo.

En la literatura, para referirse a la cantidad de los anteriores recursos que se necesitan para realizar un ataque utilizan los términos *data complexity*, *time complexity* y *memory complexity*, respectivamente. En este trabajo aludiremos a ellos como recursos necesarios en datos, tiempo de procesamiento y uso o consumo de la memoria, en el mismo orden.

Por otro lado, otro parámetro muy importante para evaluar el funcionamiento del ataque es la probabilidad de éxito.

Definición 1.11. *Se define la* ***probabilidad de éxito*** *como la probabilidad de que la clave hallada por el criptoanálisis sea la correcta.*

1.5. Rotura de un cifrado

No existe una definición exacta sobre cuándo se considera un cifrado roto, esto es, no seguro [23, 18]. En [23], se considera que un cifrado por bloques está **totalmente roto** si se puede hallar la clave completa. Además, se considera que está **parcialmente roto** si se puede hallar parte del texto plano a partir del texto cifrado pero no la clave.

Comenzaremos explicando el ataque por fuerza bruta, un ataque genérico usado frecuentemente como punto de partida para dar como roto un cifrado. Se le denomina genérico porque puede aplicarse a cualquier cifrado sin necesidad de estudiar su estructura y funcionamiento.

El **ataque por fuerza bruta** (o **búsqueda exhaustiva de la clave**) es un ataque que consiste en que dado una pareja de texto plano y su correspondiente texto cifrado (P, C), se encripta el texto plano P con todas las claves posibles hasta obtener el texto cifrado C. Otro enfoque de este ataque es realizarlo al revés: se desencriptan con todas las claves posibles el texto cifrado C hasta obtener el texto plano P. En este documento, utilizaremos el primer enfoque.

Para un cifrado por bloques cuya clave tiene una longitud de l bits, se tardaría como máximo 2^l encriptaciones en hallar la clave. Esto es un tiempo de ejecución máximo de 2^l encriptaciones. En realidad, con la mitad de encriptaciones, 2^{l-1}, se hallaría en promedio la clave. La razón de esto es que si dividimos en dos las operaciones que tenemos que realizar, hay la misma probabilidad de hallar la clave en la primera mitad que en la

segunda. Dicha probabilidad es de $\frac{2^{l-1}}{2^l} = \frac{1}{2}$. Por tanto, al probar la mitad de operaciones podría encontrar la clave con una probabilidad de $\frac{1}{2}$.

Se deduce que cuanto mayor sea el tamaño de la clave, mayor será la capacidad computacional que debemos tener para procesar todas las encriptaciones. Sin embargo, de forma teórica es siempre posible y se usa como cota máxima del tiempo de ejecución que conviene que tenga un ataque. En otras palabras, al ser un ataque que se puede aplicar a cualquier cifrado, sirve como límite que no se debe superar a la hora de desarrollar otros ataques.

El **ataque de diccionario** o **ataque con tabla** es otro ataque genérico importante. El atacante calcula y almacena cada texto plano con su correspondiente texto cifrado para cada clave posible $(P, C_i = \varepsilon_{K_i}(P), K_i)$. Así, cuando intercepte un mensaje cifrado, sólo tiene que buscar en la tabla para averiguar el texto plano. Para un cifrado por bloques de longitud n bits, los recursos en términos de datos que se necesitan son 2^n textos planos.

Se deduce que, aunque el tamaño del bloque sea pequeño y se tengan los recursos computacionales para llevarlo a cabo, es un ataque que se prolonga excesivamente en el tiempo. De nuevo, de forma teórica es siempre posible y se usa como cota máxima de la cantidad de datos que debe tener un ataque.

Observación 1.12. La cota máxima de los datos necesarios dada por el ataque por diccionario es menos frecuente que la cota máxima del tiempo de ejecución dada por el ataque por fuerza bruta.

Como consecuencia de ambos ataques, se consideran las siguientes definiciones:

Definición 1.13. *Se considera que un cifrado por bloques está* ***teóricamente roto*** *si existe un ataque cuyos recursos necesarios en términos de datos o en términos de tiempo de ejecución son menores que uno de los indicados por los anteriores ataques.*

Definición 1.14. *Se considera que un cifrado por bloques está* ***roto*** *si existe un ataque que teóricamente rompa el cifrado y se tienen los recursos computacionales para llevarlo a cabo.*

Visto de otra manera, un **cifrado** será considerado **seguro** si es inviable en la práctica el ataque por fuerza bruta y por diccionario y no se conoce ningún otro potencialmente posible que necesite recursos menores a los ataques mencionados.

En la práctica, si un ataque logra romper varias rondas (aunque no sean todas las que componen el cifrado) de forma más eficiente que el ataque por fuerza bruta, se exploran otros cifrados más seguros.

1.6. Funciones booleanas

Consideremos el cuerpo finito $(\mathbb{F}_2, \oplus, \cdot)$ donde denotaremos a la suma de este cuerpo como la operación *exclusive-or* (XOR).

Definición 1.15. *Se llama* ***función booleana*** *a toda función*

$$f : \mathbb{F}_2^n \to \mathbb{F}_2$$

Las funciones booleanas se pueden representar de diversas maneras. Este documento se limita a la **forma algebraica normal** (*Algebraic Normal Form*, ANF) por ser la forma más usual en criptografía de expresarlas:

$$f(x) = f(x_1, \ldots, x_n) = \bigoplus_{i=1}^{n} \bigoplus_{j_1,\ldots,j_i} a_{j_1,\ldots,j_i} x_{j_1} \cdot \ldots \cdot x_{j_i} \bigoplus a_0 =$$
$$= a_{1,2,\ldots,n} x_1 x_2 \ldots x_n \oplus \ldots \oplus a_1 x_1 \oplus \ldots \oplus a_n x_n \oplus a_0$$

donde para cada i, los índices $j_1, \ldots, j_i$ son distintos dos a dos, los conjuntos $\{j_1, \ldots, j_i\}$ son todos los subconjuntos posibles de $\{1, \ldots, n\}$ y los coeficientes $a_{j_1,\ldots,j_i} \in \mathbb{F}_2$.

Con esta representación, las funciones booleanas son polinomios en n variables con coeficientes en $\mathbb{F}_2$ y variables distintas en cada monomio:

$$f(x) \in \mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n)$$

A estos polinomios se les conocen como **polinomios de Zhegalkin**. En [7], se demuestra la existencia y unicidad de la representación ANF.

Definición 1.16. *Se llama* ***grado algebraico de una función booleana*** *al número de variables en el monomio de mayor orden con coeficiente no nulo.*

A continuación introduciremos algunos conceptos de teoría de la información como la distancia y el peso de Hamming por su relevancia en el ámbito de las funciones booleanas.

Definición 1.17. *Se denomina* ***distancia de Hamming entre dos vectores*** $u = (u_1, \ldots, u_n)$ *y* $v = (v_1, \ldots, v_n)$ *al número de componentes diferentes en ambos vectores.*

$$d_H(u, v) = |\{i \mid u_i \neq v_i\}| = |\{i \mid u_i - v_i \neq 0\}|$$

Definición 1.18. *Se denomina* ***peso de Hamming de un vector*** $u = (u_1, \ldots, u_n)$ *al número de componentes de* u *diferentes de 0.*

$$w_H(u) = |\{i \mid u_i \neq 0\}|$$

Observación 1.19. Se deduce de ambas definiciones que

$$d_H(u, v) = w_H(u - v)$$

En términos de funciones booleanas, las definiciones anteriores se extienden como [7]:

Definición 1.20. *Se denomina* ***distancia de Hamming entre dos funciones booleanas*** f *y* g *al número de vectores de* $\mathbb{F}_2^n$ *cuya imagen por* f *y* g *son distintos.*

$$d_H(f, g) = |\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}| = |\{x \in \mathbb{F}_2^n \mid f(x) \oplus g(x) \neq 0\}|$$

Definición 1.21. *Se denomina* ***peso de Hamming de una función booleana*** f *en* $\mathbb{F}_2^n$ *como el cardinal del soporte de la función* f*.*

$$w_H(f) = |\{x \in \mathbb{F}_2^n \mid f(x) \neq 0\}|$$

Observación 1.22. Se deduce de ambas definiciones que

$$d_H(f, g) = w_H(f \oplus g)$$

En algunas ocasiones se trabajará con el cuerpo $\mathbb{F}_2^n$ y necesitaremos a siguiente noción:

Definición 1.23. *Se llama* ***producto interno*** *o* ***producto punto*** *de dos n-uplas $\alpha, x \in \mathbb{F}_2^n$ a la operación tal que:*

$$\begin{aligned} \cdot : \mathbb{F}_2^n \times \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2 \\ (\alpha, x) &\longmapsto \alpha \cdot x = \textstyle\bigoplus_{i=1}^{n} \alpha_i \cdot x_i \end{aligned}$$

Definición 1.24. *Se denominan* ***funciones (booleanas) afines*** *a las funciones booleanas que, en ANF, se expresan como:*

$$f(x) = f(x_1, \ldots, x_n) = a_1 x_1 \oplus \ldots \oplus a_n x_n \oplus a_0$$

Definición 1.25. *Se denominan* ***funciones (booleanas) lineales*** *a las funciones booleanas que, en ANF, se expresan como:*

$$f(x) = f(x_1, \ldots, x_n) = a_1 x_1 \oplus \ldots \oplus a_n x_n$$

Las funciones lineales son funciones afines con $a_0 = 0$.

Observación 1.26. Todas las funciones lineales son de la forma $l_a(x) = a \cdot x$ con $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$. Como consecuencia, las funciones afines pueden ser expresadas como $f(x) = l_a(x) \oplus a_0$ con $a_0 \in \mathbb{F}_2$

Antes de describir conceptos relevantes sobre funciones booleanas utilizados en criptografía, son necesarias algunas nociones de transformadas de Fourier aplicadas a dichas funciones.

Definición 1.27. *Se llama* ***función pseudo-booleana*** *a toda función*

$$f : \mathbb{F}_2^n \to \mathbb{R}$$

Definición 1.28. *Se llama* ***transformada discreta de Fourier de una función pseudo-booleana*** *φ a la aplicación lineal que transforma φ en otra función pseudo-booleana $\hat{\varphi}$ definida como:*

$$\hat{\varphi}(u) = \sum_{x \in \mathbb{F}_2^n} \varphi(x)(-1)^{x \cdot u}$$

Para calcular $\hat{\varphi}$, se utiliza un algoritmo que recibe el nombre de Transformada rápida de Fourier (*Fast Fourier Transform*, FFT). Puede consultarse en la Sección 2.2 de [7].

Definición 1.29. *Se llama* ***transformada discreta de Fourier de una función booleana*** *f a la aplicación lineal que transforma f en otra función pseudo-booleana $\hat{f} : \mathbb{F}_2^n \to \{0, 1\} \subset \mathbb{Z}$ definida como:*

$$\hat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u}$$

Observación 1.30. Se deduce que

$$\hat{f}(0) = \sum_{x\in\mathbb{F}_2^n} f(x) = w_H(f) \quad \text{y} \quad \widehat{f\oplus g}(0) = w_H(f\oplus g) = d_H(f,g)$$

Se utiliza este tipo de transformada de Fourier sobre una función booleana porque nos da a conocer el peso de todas las funciones que son del tipo $f \oplus l$ donde f es una función booleana y l una función afín o lineal.

Definición 1.31. *Se denomina* ***función signo*** *de una función booleana f a la función definida como*

$$sgn(f)(x) = (-1)^{f(x)} = 1 - 2f(x)$$

Definición 1.32. *Llamaremos* ***transformada de Walsh-Hadamard de una función booleana*** *f a la transformada de Fourier discreta de la función signo $sgn(f)$ de f:*

$$\mathcal{W}(f)(u) = \widehat{sgn(f)}(u) = \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)\oplus x\cdot u} = 2^n\delta_0 - 2\hat{f}(u)$$

donde $\delta_0(u) = \begin{cases} 1 & u \text{ es el vector nulo} \\ 0 & \text{en otro caso} \end{cases}$ *siendo* $u \in \mathbb{F}_2^n$.

Proposición 1.33. *Sea $f : \mathbb{F}_2^n \to \mathbb{F}_2$ una función booleana y $l_a : \mathbb{F}_2^n \to \mathbb{F}_2$ una función booleana lineal. Entonces,*

$$w_H(f) = 2^{n-1} - \frac{\mathcal{W}(f)(0)}{2} \quad y \quad d_H(f,l_a) = 2^{n-1} - \frac{\mathcal{W}(f)(a)}{2}$$

Demostración. Para la primera parte del enunciado, por la Definición 1.32, $\hat{f}(u) = 2^{n-1}\delta_0 - \frac{\mathcal{W}(f)(u)}{2}$. Por la Observación 1.30,

$$w_H(f) = \hat{f}(0) = 2^{n-1} - \frac{\mathcal{W}(f)(0)}{2}$$

Para la segunda parte del enunciado, por la Observación 1.30,

$$d_H(f,l_a) = w_H(f\oplus l_a) = \widehat{f\oplus l_a}(0) = 2^{n-1} - \frac{\mathcal{W}(f\oplus l_a)(0)}{2}$$

Con todo ello,

$$\begin{aligned}\mathcal{W}(f\oplus l_a)(0) &= \sum_{x\in F_2^n} (-1)^{(f\oplus l_a)(x)\oplus x\cdot 0} = \sum_{x\in F_2^n} (-1)^{f(x)}(-1)^{l_a(x)} \\ &= \sum_{x\in F_2^n} (-1)^{f(x)}(-1)^{a\cdot x} = \sum_{x\in F_2^n} (-1)^{f(x)\oplus a\cdot x} = \mathcal{W}(f)(a)\end{aligned}$$

□

1.6.1. Propiedades de la funciones booleanas aplicadas a la criptografía

En el diseño de los cifrados, algunas propiedades de las funciones booleanas tienen un papel muy importante. Con ellas, es posible estimar la resistencia de los cifrados por bloques frente a ataques conocidos.

Grado algebraico

Se ha demostrado que las funciones booleanas utilizadas en criptografía deben tener un grado algebraico alto. De lo contrario, son susceptibles a ataques denominados ataques algebraicos, de interpolación o diferenciales de orden superior [7]. También se ha probado que si $f : \mathbb{F}_2^n \to \mathbb{F}_2$ es una función booleana con grado algebraico n o $n-1$, no va a ser posible exigirle que tenga algunas propiedades consideradas fundamentales para una buena resistencia a ataques debido a la incompatibilidad entre algunas de ellas.

No-linealidad

Definición 1.34. *Se define la* ***no-linealidad de una función booleana*** f*,* $nl(f)$*, como la mínima distancia de Hamming entre la función* f *y todas las funciones afines.*

$$nl(f) = \min\{d_H(f, g) \text{ con } g \text{ función afín}\}$$

En criptografía, es deseable que las funciones booleanas tengan una alta no-linealidad. El motivo es evitar encontrar una relación entre el vector de entrada y el de salida. Si existiese una alta relación entre las funciones de un criptosistema y las funciones afines/lineales, entonces se podrían efectuar ataques que se describen en [7]. En el caso de cifrados por bloques, debemos consultar la Sección 1.6.2 de este documento.

La no-linealidad también se puede definir en términos de la transformada de Walsh-Hadamard y, así, poder calcularla eficientemente [10]. Por la Proposición 1.33 tenemos la distancia una función booleana f y una función lineal l_a. Se deduce de la misma que:

$$d_H(f, l_a \oplus 1) = 2^{n-1} + \frac{\mathcal{W}(f)(a)}{2}$$

Con ello,

Definición 1.35. *Se define la* ***no-linealidad de una función booleana*** f*,* $nl(f)$ *como*

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} |\mathcal{W}(f)(a)|$$

Se deduce que una función booleana f tiene una alta no-linealidad si la transformada de Walsh-Hadamard es pequeña.

Balance

Definición 1.36. *Se denomina* ***función booleana balanceada*** *a la función booleana* f *tal que su salida tiene el mismo número de* 0 *y* 1*, es decir, su peso de Hamming es* 2^{n-1}*.*

Se puede intuir que las funciones booleanas utilizadas en criptografía deben ser funciones balanceadas con el fin de evitar una dependencia estadística entre el texto plano y cifrado. Así, prevenimos ataques estadísticos como los que se tratarán en este trabajo.

En términos de la transformada de Walsh-Hadamard:

Proposición 1.37. *Una función booleana f es balanceada si y sólo si $\mathcal{W}(f)(0) = 0$.*

Demostración. El peso de Hamming de una función booleana balanceada f es $w_H(f) = 2^{n-1}$. Por la Proposición 1.33, $w_H(f) = 2^{n-1} - \frac{\mathcal{W}(f)(0)}{2}$. Despejando $\mathcal{W}(f)(0)$ y sustituyendo el peso, obtenemos el enunciado. □

Autocorrelación

Definición 1.38. *Se llama **autocorrelación de una función booleana** f a la función*

$$r_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(x \oplus a)} \in [-2^n, 2^n]$$

La autocorrelación mide si una función se parece a ella misma, desplazando con un valor fijo $a \in \mathbb{F}_2^n$ todos los valores de la función. Por ello, tener una función de autocorrelación baja significa que un cambio en el vector de entrada de una función no nos diría nada sobre el posible cambio que se produzca en el vector de salida.

Propagación

Con el término de "propagación" nos referiremos a la propiedad que pueden tener los cifrados cuando cumplen el llamado criterio de propagación y, particularmente, el criterio de avalancha estricto.

En general, se denomina efecto avalancha a un gran cambio en la salida de una función o algoritmo cuando hacemos una pequeña modificación en la entrada. En términos de cifrados, esta propiedad tiene gran importancia porque dificulta el análisis estadístico de los textos cifrados al dificultar relacionar la entrada de una función booleana con su salida. En el ámbito de las funciones booleanas, este concepto se denomina criterio estricto de avalancha [7, 10].

Definición 1.39. *Una función booleana $f : \mathbb{F}_2^n \to \mathbb{F}_2$ cumple el **criterio de avalancha estricto** (Strict Avalanche Criterion, SAC) si la función $f(x) \oplus f(x \oplus a)$ es balanceada para cada $a \in \mathbb{F}_2^n$ con $w_H(a) = 1$. En términos de la función de autocorrelación:*

$$r_f(a) = 0 \quad \forall\, a \in \mathbb{F}_2^n \quad \textit{tal que} \;\; w_H(a) = 1$$

Ejemplo 1.40. La siguiente función booleana $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ cumple el SAC: [10],

Entrada	000	001	010	011	100	101	110	111
Salida	1	1	1	0	0	1	1	1

Mediante *SageMath* (véase el Apéndice C) se calcula la función de autocorrelación de esta función, $r_f(a)$, y obtenemos:

$$(8, 0, 0, 0, 0, 0, 0, 8)$$

donde el orden de los coeficientes es el marcado por la "Entrada" de la tabla. Por tanto, los vectores $(001)_2, (010)_2, (100)_2$, tienen como coeficiente de autocorrelación $r_f(a) = 0$.

Definición 1.41. *Una función booleana* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *cumple el* ***criterio de propagación de grado*** k *(Propagation criterion of degree k, PC(k)) si la función* $f(x) \oplus f(x \oplus a)$ *es balanceada para cada* $a \in \mathbb{F}_2^n$ *con* $1 \leq w_H(a) \leq k$*. En términos de la función de autocorrelación,*

$$r_f(a) = 0 \quad \forall\, a \in \mathbb{F}_2^n \quad tal\ que\ \ 1 \leq w_H(a) \leq k$$

El criterio de propagación generaliza el criterio de avalancha, siendo el SAC igual a PC(1).

En conclusión, las funciones booleanas en criptografía deben ser balanceadas, con un alto grado algebraico, una alta no-linealidad y satisfacer los criterios SAC (o PC con un alto grado). Han sido ampliamente investigados métodos para obtener funciones booleanas con algunas de las propiedades. Sin embargo, no se puede alcanzar el nivel óptimo en todas ellas, por lo que, los diseñadores de cifrados deben tener en cuenta las incompatibilidades que surgen entre ellas [7].

1.6.2. Funciones booleanas vectoriales

Definición 1.42. *Se llama* ***función booleana vectorial*** *a toda función*

$$F : \mathbb{F}_2^n \to \mathbb{F}_2^m$$

Definición 1.43. *Se llaman* ***funciones coordenadas de una función booleana vectorial*** $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *a las funciones booleanas* $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$ *que forman la función* F*.*

$$F = (f_1, \ldots, f_m)$$

Definición 1.44. *Se llama* ***funciones componentes de la función booleana vectorial*** $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *a todas las funciones* $a \cdot F(x)$ *para* $a \in \mathbb{F}_2^m - \{0\}$*, es decir, a todas las combinaciones lineales de las funciones coordenadas de* F*.*

Observación 1.45. El conjunto de las funciones componentes es un espacio vectorial generado por las funciones coordenadas si éstas son linealmente independientes en $\mathbb{F}_2$ (y sin la función nula) [8].

Las funciones booleanas vectoriales también tienen diversas formas de presentación pero nos limitaremos a la ANF. Sabemos que las funciones booleanas vectoriales $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ pueden ser descompuestas en sus funciones coordenadas. Como cada función coordenada f_i de F $\forall i \in \{1, \ldots, m\}$ puede ser representada de manera única en ANF, la función F también puede ser representada de manera única como un polinomio en n variables con coeficientes en $\mathbb{F}_2^m$ y variables distintas en cada monomio. Esto es,

$$F(x_1, \ldots, x_n) \in \mathbb{F}_2^m[x_1, \ldots, x_n]/(x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n)$$

Observación 1.46. Si se toma la coordenada i de cada coeficiente, se obtiene la función coordenada i de F.

Definimos la siguiente operación por su importancia en los próximos capítulos.

Definición 1.47. *Se llama* ***grado algebraico*** *de una función booleana vectorial al máximo grado algebraico entre todas sus funciones coordenadas.*

Definición 1.48. *Llamaremos* ***transformada de Walsh-Hadamard de una función booleana vectorial*** F *a la función que transforma cualquier par* $(u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ *en el valor en* u *de la transformada de Walsh-Hadamard de la función componente* $v \cdot F$, $v \neq 0$, *que es*

$$\mathcal{W}(f)(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) \oplus u \cdot x}$$

Definición 1.49. *Se dice que una* ***función booleana vectorial*** $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *con* $n \geq m$ *es* ***balanceada*** *si toma cada valor de* $\mathbb{F}_2^m$ *un total de* 2^{n-m} *veces.*

Si $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ es biyectiva, entonces $n = m$, F es balanceada y sus funciones coordenadas también lo son. Además, F es una permutación en $\mathbb{F}_2^n$ [8].

Definición 1.50. *Se define la* ***no-linealidad de una función booleana vectorial*** F, $nl(F)$, *como la mínima no-linealidad de todas las funciones componentes* $v \cdot F$ *con* $v \in \mathbb{F}_2^m$.

Los siguientes conceptos serán útiles para los capítulos sobre criptoanálisis lineal y diferencial.

Definición 1.51. *Sea* $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *una función booleana vectorial. Se llama* ***aproximación lineal*** *de* F *a un par ordenado* $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ *donde* α, β *reciben el nombre de* ***máscaras lineales*** *o, simplemente,* ***máscaras****. Específicamente,* α *es la máscara de entrada y* β *es la máscara de salida. Se dice que la aproximación lineal tiene una probabilidad* $p = \mathbb{P}_{x \in \mathbb{F}_2^n}(\alpha \cdot x = \beta \cdot F(x)) = \mathbb{P}_{x \in \mathbb{F}_2^n}(\alpha \cdot x \oplus \beta \cdot F(x) = 0)$

Las aproximaciones lineales siempre existen pero no necesariamente esa probabilidad es cercana a 1. No son únicas por el mismo motivo, ya que habrá aproximaciones mejores, es decir, con mayor probabilidad, que otras de una misma función F. A lo largo del documento, utilizaremos como sinónimo de este concepto "expresión lineal".

Definición 1.52. *Sea* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *una función booleana. Se llama* ***derivada direccional*** *de* f *en* $a \in \mathbb{F}_2^n$ *a la función*

$$\begin{aligned} f_a(x) : \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2 \\ x &\longmapsto f(x \oplus a) \oplus f(x) \end{aligned}$$

Esta definición también se amplía a las funciones booleanas vectoriales.

Definición 1.53. *Sea* $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *una función booleana vectorial. Se llama* ***derivada direccional*** *de* F *en* $a \in \mathbb{F}_2^n$ *a la función*

$$\begin{aligned} F_a(x) : \mathbb{F}_2^n &\longrightarrow \mathbb{F}_2^n \\ x &\longmapsto F(x \oplus a) \oplus F(x) \end{aligned}$$

El operador derivada direccional comparte propiedades como la regla de la suma o del producto (Regla de Leibniz) del operador derivada del cálculo diferencial [14].

Capítulo 2

Cifrados por bloques

Dentro de los cifrados por bloques se analizan dos estructuras diferenciadas: las **redes de sustitución-permutación** (*Substitution–permutation network*, SPN) y las **redes de Feistel** (*Feistel network*). La gran mayoría de cifrados modernos de clave secreta derivan de alguna de estas dos estructuras. De hecho, la diferencia entre cifrados radica, fundamentalmente, en la red utilizada, además del algoritmo de generación de claves[2]. Ejemplos de redes de sustitución-permutación son el ya mencionado AES o el SERPENT, uno de los cifrados que compitió junto con el primero por ser el estándar para cifrar la información digital. Con respecto a ejemplos de redes de Feistel, podemos nombrar el DES o el GOST. Éste último es considerado el equivalente soviético del DES para cifrar la información secreta de gobierno de la antigua URSS.

Los criptoanálisis desarrollados en este trabajo se aplicarán a redes SPN, por tanto, dichas redes serán el tema central de este capítulo.

2.1. Redes de sustitución-permutación

En esta sección se describen de manera esquemática las redes SPN. Posteriormente se ilustra con el cifrado PRESENT cómo se aplican estas redes. Usaremos las referencias bibliográficas [16, 30, 3].

Una **red de sustitución-permutación** (SPN) es una estructura de cifrados por bloques de r rondas cuya función de ronda f consiste en tres fases: combinación con claves, sustitución σ y permutación π. Como se mencionó en la Sección 1.2, las subclaves k_i son generadas por algún algoritmo de generación de claves a partir de la clave maestra $K \in \mathcal{K}$. Este algoritmo varía entre cifrados: algunos tienen su propio procedimiento y otros utilizan los llamados **generadores de claves**, protocolos ampliamente utilizados en diferentes ámbitos de la criptografía que hacen uso de números pseudoaleatorios. En esta sección asumiremos que las subclaves k_i ya han sido generadas.

Combinación con claves

En esta etapa se realiza la operación XOR (bit a bit) entre la subclave k_i y el bloque de salida de la fase de permutación de la ronda anterior. En la primera ronda, el XOR se hace

[2] Algunos como el cifrado Kuznyechik utilizan ambas redes: es una SPN pero su algoritmo de generación de claves es una red de Feistel.

directamente con el texto plano. En la última ronda, el XOR se aplica después de la fase de sustitución ya que no hay fase de permutación. Por este motivo, tenemos $r+1$ subclaves k_i para una SPN de r rondas. A este procedimiento se le conoce como blanqueamiento de la clave (*key whitening* o *whitening*). Se utiliza para aumentar la seguridad del cifrado, además de hacer que el proceso de desencriptación sea algorítmicamente idéntico al proceso de encriptación.

Sustitución $\sigma : \mathbb{F}_2^n \to \mathbb{F}_2^n$

Esta etapa consiste en transformar el bloque de tamaño n en otro del mismo tamaño a través de las denominadas cajas de sustitución. Matemáticamente, la pensaremos como una aplicación σ compuesta por las aplicaciones de las cajas de sustitución.

Definición 2.1. *Una **caja de sustitución** (o S-box) es una función booleana vectorial no lineal* $\mathcal{S} : \mathbb{F}_2^{m_1} \to \mathbb{F}_2^{m_2}$.

En lo que queda de documento se utilizará el término "S-box", ya que es un término ampliamente aceptado en la literatura en español.

Nótese que la longitud del vector de entrada de una S-box puede ser distinta a la del vector resultante. En general, en las redes SPN $m_1 = m_2$ pero en las redes de Feistel puede darse el caso $m_1 \neq m_2$. Al ser esta sección sobre redes SPN, se supondrá que $m_1 = m_2$ y lo denotaremos simplemente por m. Además, en estas redes, va a ser necesario que las S-boxes sean biyectivas para permitir la desencriptación [21].

Esta fase consiste en, primero, dividir el bloque de tamaño n en subbloques de tamaño m. Después, se aplica a cada subbloque una S-box de manera independiente. Finalmente, se concatenan los vectores de salida de cada S-box para obtener, de nuevo, el bloque de tamaño n. Se deduce que el número de S-boxes de esta fase depende de los tamaños de bloque n y el texto de entrada de la S-box m. Por tanto, σ está formada por varias S-boxes $\mathcal{S}_j \; \forall j \in \left\{0, \ldots, \frac{n}{m}\right\}$.

La seguridad de los cifrados reside en esta etapa ya que las S-boxes son el único componente no lineal de estas redes. Como consecuencia, su diseño está extensamente estudiado, teniendo en cuenta las propiedades descritas en la Sección 1.6 sobre funciones booleanas, entre otras.

Con respecto a los cifrados, las S-boxes pueden ser utilizadas de diferentes maneras. Por un lado, algunos cifrados utilizan la misma aplicación S-box para todas las cajas de una misma ronda y esta disposición se mantiene en todas las rondas. Por otro lado, en otros cifrados, como en el DES, se utilizan diferentes aplicaciones S-box en una misma ronda pero la disposición, de nuevo, se mantiene entre ellas.

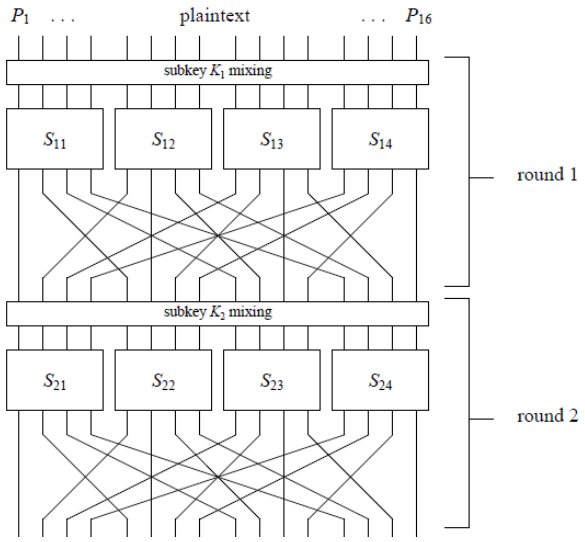

Figura 2.1: 2 rondas de una SPN con $n = 16$ y $m = 4$

Permutación $\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$

Esta fase consiste en una permutación de los n bits del bloque resultante después de la etapa de sustitución. Denotaremos a la aplicación de permutación como π.

Al diseñar esta fase del cifrado, se busca que los bits de salida de las S-boxes se distribuyan por el mayor número posible de bits de entrada de las S-boxes de la iteración siguiente. Dicho de otro modo, se trata de distribuir los bits por tantas S-boxes que no sea posible hallar relaciones entre las distintas rondas del cifrado.

En la última ronda del cifrado ya no hay permutación: después de la sustitución el bloque se combina con la subclave. La finalidad de esto es asegurar que el proceso de desencriptación sea idéntico a nivel de etapas que el de encriptación (al igual que con el blanqueamiento de la clave).

En conclusión, una **SPN de r rondas** es la composición de $r-1$ funciones de ronda $f_i(x) = \pi(\sigma(z_{i-1} \oplus k_i))\ \forall i \in \{1, \dots, r-1\}$ más la última ronda descrita como $f_r(x) = \sigma(z_{r-1} \oplus k_r) \oplus k_{r+1}$.

2.1.1. PRESENT

PRESENT es una red SPN de 31 rondas con un tamaño de bloque de 64 bits y que permite dos tamaños de clave (maestra), 80 ó 128 bits. Fue diseñado por un grupo de colaboración entre la Universidad del Ruhr de Bochum, la Universidad Técnica de Dinamarca y la empresa Orange [6]. PRESENT pertenece a un tipo de cifrado llamado *ultra-lightweight* o *lightweight* ya que se implementa en dispositivos muy pequeños, como sensores. Junto a CLEFIA y LEA, es uno de los cifrados estandarizados para ello por la Organización Internacional de Normalización (ISO) [13]. véase el Apéndice A para una representación esquemática del cifrado.

Combinación con claves

En la ronda i donde $i \in \{1, \dots, 31\}$, se realiza un XOR entre la subclave $k_i = k_{63}^i \dots k_0^i \ \ \forall i \in \{1, \dots, 32\}$[3] de tamaño 64 bits, generada por el algoritmo de generación de claves del PRESENT, y el texto de la iteración $i-1$ que denotaremos como $z_{63} \dots z_0$.

$$\begin{aligned} \oplus : \mathbb{F}_2^{64} &\longrightarrow \mathbb{F}_2^{64} \\ z_j &\longrightarrow z_j \oplus k_j^i \end{aligned} \qquad \forall j \in \{0, \dots, 63\}$$

Sustitución $\sigma : \mathbb{F}_2^{64} \to \mathbb{F}_2^{64}$

La sustitución del bloque de 64 bits se hace mediante la aplicación de 16 S-boxes en paralelo de tamaño 4 bits cada una. Se utiliza una misma aplicación $\mathcal{S} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ para todas las S-boxes del cifrado y para cada ronda.

Si $z_{63} \dots z_0$ es el texto de la iteración i, se divide en 16 bloques $w_j\ \forall j \in \{0, \dots, 15\}$ de 4 bits cada uno.

$$w_j = z_{4j+3} \parallel z_{4j+2} \parallel z_{4j+1} \parallel z_{4j} \quad \forall j \in \{0, \dots, 15\}$$

[3] Observemos que hay 32 subclaves k_i debido a las 31 rondas del PRESENT

El funcionamiento de la S-box viene descrito por el Cuadro 2.1 en notación hexadecimal. La primera fila es el bloque de entrada de 4 bits que tenemos en la S-box. La segunda fila es la imagen de la aplicación $\mathcal{S}$.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\mathcal{S}(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Cuadro 2.1: S-box del cifrado PRESENT

Permutación $\pi : \mathbb{F}_2^{64} \to \mathbb{F}_2^{64}$

La permutación se hace bit a bit y se expresa como

$$\pi(k) = \begin{cases} 16 \cdot k \bmod 63 & k \in \{0, \ldots, 62\} \\ 63 & k = 63 \end{cases} \tag{2.1}$$

En la literatura, la representación de esta fase se hace en forma de tabla. Puede consultarse en la Figura A.1.

Generación de claves

PRESENT admite dos tamaños de clave maestra K, una de 80 bits y otra de 128 bits. Para ilustrar el algoritmo de generación de claves usaremos la clave de longitud de 80 bits. En [6] puede consultarse el correspondiente a la clave de longitud 128 bits.

El usuario proporciona la clave maestra de tamaño 80 bits $K = \kappa_{79} \ldots \kappa_0$ que se irá actualizando en cada ronda. En cada iteración i, la subclave $k_i = k_{63}^i \ldots k_0^i$ estará formada por los 64 bits más a la izquierda de la clave maestra K:

$$k_i = k_{63}^i \ldots k_0^i = \kappa_{79} \ldots \kappa_{16}$$

Después de seleccionar dichos 64 bits, la clave maestra $K = \kappa_{79} \ldots \kappa_0$ se actualiza siguiendo tres pasos. Primero, se desplazan todos los bits 61 posiciones a la izquierda (o 19 posiciones a la derecha). Después, a los 4 bits situados más a la izquierda se les aplica la S-box descrita. Por último, a los bits $\kappa_{19}\kappa_{18}\kappa_{17}\kappa_{16}\kappa_{15}$ se les realiza un XOR con el valor de la iteración i en binario. En resumen, el algoritmo de generación de claves puede describirse como:

1. $\left[\kappa_{79}^{'}\kappa_{78}^{'} \ldots \kappa_1^{'}\kappa_0^{'}\right] = \left[\kappa_{18}\kappa_{17} \ldots \kappa_{20}\kappa_{19}\right]$
2. *a)* $\left[\kappa_{79}^{''}\kappa_{78}^{''}\kappa_{77}^{''}\kappa_{76}^{''}\right] = \mathcal{S}(\left[\kappa_{79}^{'}\kappa_{78}^{'}\kappa_{77}^{'}\kappa_{76}^{'}\right])$
   b) $\left[\kappa_{19}^{''}\kappa_{18}^{''}\kappa_{17}^{''}\kappa_{16}^{''}\kappa_{15}^{''}\right] = \left[\kappa_{19}^{'}\kappa_{18}^{'}\kappa_{17}^{'}\kappa_{16}^{'}\kappa_{15}^{'}\right] \oplus i$
   c) $\kappa_i^{''} = \kappa_i^{'} \ \forall\, i \in \{0, \ldots, 14, 20, \ldots, 75\}$

siendo $[\kappa_{79}\kappa_{78} \ldots \kappa_1\kappa_0]$ la clave maestra de la iteración i, $\left[\kappa_{79}^{'}\kappa_{78}^{'} \ldots \kappa_1^{'}\kappa_0^{'}\right]$ la clave maestra desplazada y $\left[\kappa_{79}^{''}\kappa_{78}^{''} \ldots \kappa_1^{''}\kappa_0^{''}\right]$ la clave maestra de la iteración $i+1$. Para visualizar este proceso, véase la Figura 2.2.

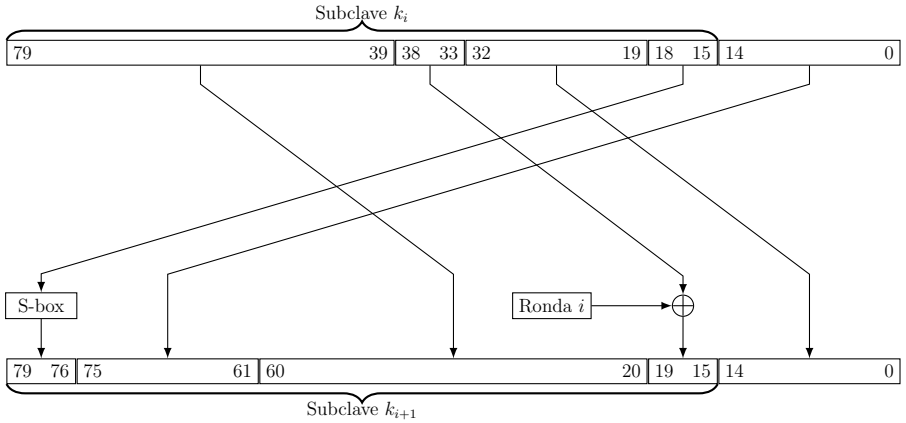

Figura 2.2: Generación de subclaves de PRESENT [17]

Análisis booleano de los componentes del PRESENT

La S-box del PRESENT (véase la Tabla 2.1), al ser una función booleana vectorial $\mathcal{S} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$, se puede descomponer en cuatro funciones booleanas coordenadas $\mathcal{S}_i : \mathbb{F}_2^4 \to \mathbb{F}_2$ con $i \in \{1, 2, 3, 4\}$. En el Cuadro 2.2 se puede consultar la salida de cada una de ellas donde el subíndice 4 indica el bit menos significativo[4].

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\mathcal{S}(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2
$\mathcal{S}_1(x)$	1	0	0	1	1	0	1	1	0	1	1	1	0	0	0	0
$\mathcal{S}_2(x)$	1	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0
$\mathcal{S}_3(x)$	0	0	1	1	0	0	1	0	1	1	1	0	0	1	0	1
$\mathcal{S}_4(x)$	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0

Cuadro 2.2: S-box del cifrado PRESENT descompuesta en funciones coordenadas

Aplicando los códigos escritos en el Apéndice C obtemos los resultados descritos en la Tabla 2.3.

	$\mathcal{S}_1(x)$	$\mathcal{S}_2(x)$	$\mathcal{S}_3(x)$	$\mathcal{S}_4(x)$
Grado algebraico	3	3	3	2
No linealidad	4	4	4	4
Balanceada	Sí	Sí	Sí	Sí
Propagación: SAC	No	No	No	No

Cuadro 2.3: Resultados del análisis booleano de $\mathcal{S}_i(x)$ realizado con *SageMath*

Sin la función concreta para calcular la no-linealidad de *SageMath*, se podría haber computado por la Definición 1.35 de no-linealidad en términos de la transformada de Walsh-Hadamard.

[4]El bit que está más a la derecha. Término estándar

Si se calculan las transformadas de Walsh-Hadamard de cada una de las S-boxes $\mathcal{S}_i(x)$, obtenemos que

$$\begin{aligned}
\mathcal{S}_1 &: (0,0,4,-4,-4,4,0,0,-4,-4,0,-8,8,0,-4,-4)\\
\mathcal{S}_2 &: (0,0,-4,4,-4,-4,0,8,0,0,4,-4,-4,-4,-8,0)\\
\mathcal{S}_3 &: (0,0,4,4,-4,-4,0,0,4,-4,8,0,0,8,4,-4)\\
\mathcal{S}_4 &: (0,0,0,0,0,0,0,0,0,8,0,-8,0,8,0,8)
\end{aligned}$$

Se observa que, efectivamente, la no-linealidad de cada $\mathcal{S}_i(x)$ es 4.

De la misma manera podemos proceder si queremos estudiar si las funciones booleanas son balanceadas por la Proposición 1.37, que hace uso de estas transformadas.

Por otro lado, podemos estudiar también la autocorrelación y si cumple el criterio SAC o $PC(k)$. Como estos dos criterios utilizan la función de autocorrelación y, criptográficamente, son más interesantes que un análisis en solitario de dicha función, comentaremos sólo estos dos criterios.

La función de autocorrelación de cada una de las S-boxes $\mathcal{S}_i(x)$ es

$$\begin{aligned}
\mathcal{S}_1 &: (16,0,0,8,0,-8,-8,0,-8,0,0,-8,0,8,0,0)\\
\mathcal{S}_2 &: (16,0,-8,0,-8,0,0,0,0,-8,0,8,0,8,0,-8)\\
\mathcal{S}_3 &: (16,0,0,-8,0,8,-8,0,-8,0,0,8,0,-8,0,0)\\
\mathcal{S}_4 &: (16,-16,0,0,0,0,0,0,-16,16,0,0,0,0,0,0)
\end{aligned}$$

Al igual que en el Ejemplo 1.40, el orden de estos coeficientes se corresponde con el orden de los números naturales comenzando por $0 = (0000)_2$ hasta $15 = (1111)_2$. Se recuerda que para que las S-boxes $\mathcal{S}_i$ cumplan el criterio SAC (véase Definición 1.39), su función de autocorrelación $r_{\mathcal{S}_i}(a)$ debe ser nula para todos los vectores cuyo peso de Hamming sea 1. En este caso, hay que estudiar en cada uno de estos resultados las posiciones 2, 3, 5 y 9 porque corresponden a los números 1, 2, 4 y 8 que tienen un peso de Hamming de 1 (escritos en binario y considerados como vectores en $\mathbb{F}_2^4$).

Se observa que ninguna de las funciones coordenadas de la S-box del PRESENT cumple el criterio SAC. Las S-boxes $\mathcal{S}_1$ y $\mathcal{S}_3$ se acercan a cumplir el criterio puesto que sólo 1 de los coeficientes es no nulo. Por otro lado, la que más alejada de cumplir el criterio es la $\mathcal{S}_4$ por tener dos coeficientes no nulos y ambos -16, cuyo valor absoluto es el más alto que puede tener la función de autocorrelación.

Los criterios de propagación $PC(k)$ con $k = 2, 3, 4$ ya no se van a cumplir porque no se cumplen para el $PC(1)$ (véase la Definición 1.41).

En este trabajo no se va a profundizar más sobre el análisis del criterio SAC por alejarse del tema principal. Sin embargo, se advierte de que existen algunas relaciones entre el criterio SAC y el grado algebraico [7].

Ahora, respecto a las propiedades booleanas para la S-box como función booleana vectorial $\mathcal{S} : \mathbb{F}_2^4 \to \mathbb{F}_2^4$, obtenemos los resultados de la Tabla 2.4 mediante *SageMath* (véase el Apéndice C).

Se observa que las propiedades analizadas en la S-box concuerdan con las estudiadas en sus funciones coordenadas. Otras propiedades que demuestran el nivel de seguridad del PRESENT se verán en los Capítulos 3 y 4.

	$\mathcal{S}(x)$
Grado algebraico	3
No linealidad	4
Balanceada	Sí

Cuadro 2.4: Resultados del análisis booleano de $\mathcal{S}(x)$ realizado con *SageMath*

2.2. Redes de Feistel

En esta sección se describen de manera esquemática las redes de Feistel. Usaremos las referencias bibliográficas [23, 32].

Una **red de Feistel** es una estructura de cifrados por bloques de r rondas que divide el texto plano P de longitud n en dos partes iguales llamadas (L_0, R_0) para transformarlo en un texto cifrado (L_r, R_r) mediante r funciones de ronda f descrita como:

$$\begin{array}{c} f : K \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n \\ (k_i, L_{i-1}R_{i-1}) \longrightarrow (L_iR_i) \end{array} \text{ siendo } \begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F(R_{i-1}, k_i) \end{cases} \tag{2.2}$$

donde F es una función que utiliza la subclave k_i y es característica de cada cifrado. La complejidad de los cifrados reside en esta función y, normalmente, es una composición de aplicaciones de sustitución, permutación y operaciones bit a bit [32].

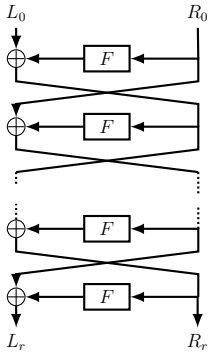

Figura 2.3: Red de Feistel

Observación 2.2. Este proceso es invertible independientemente de la función F aplicada. En otras palabras, la función F puede no ser inyectiva y la red seguirá siendo invertible.

$$\begin{cases} R_{i-1} = L_i \\ L_{i-1} = R_i \oplus F(R_{i-1}, k_i) = R_i \oplus F(L_i, k_i) \end{cases}$$

Por tanto, el proceso de desencriptación es idéntico al de encriptación.

Como se mencionó en la Sección 1.2, las subclaves k_i son generadas por un algoritmo de generación de claves a partir de la clave maestra $K \in \mathcal{K}$. Al igual que en las redes SPN, este algoritmo varía entre cifrados y puede ser tanto propio como con protocolos conocidos.

Observación 2.3. Si se divide el texto plano en más partes, como en cuartos, obtendremos los llamados **cifrados generalizados de Feistel**.

El ejemplo más relevante de redes de Feistel es el ya mencionado Data Encryption Standard (DES). Se puede consultar en [23, 32, 30] su funcionamiento.

Capítulo 3

Criptoanálisis lineal

En este capítulo desarrollaremos el criptoanálisis lineal aplicado a redes SPN. Esto permitirá una mejor compresión del criptoanálisis diferencial. Usaremos las referencias bibliográficas [16, 32, 25, 29, 5, 28]. Sin embargo, en estas referencias tiene una orientación generalista y de divulgación, incluso reconocen que no dan precisión matemática. Por ello, parte de mi labor ha sido adecuar este texto al rigor matemático.

El criptoanálisis lineal es un ataque de tipo KPA[5] desarrollado por Mitsuru Matsui en 1993. Fue aplicado contra el DES de forma teórica ya que, a pesar de usar menos recursos que una ataque por fuerza bruta, no fue posible llevarlo a la práctica por la gran cantidad de ellos que aún suponían. Sin embargo, su trabajo es muy importante para el análisis de seguridad de los cifrados modernos.

En términos generales, el atacante va a intentar aproximar el cifrado mediante expresiones lineales formadas por los bits del texto plano, del texto cifrado y de la clave. Dichas expresiones van a tener una probabilidad de ocurrencia por ser aproximaciones. El objetivo es buscar aquellas que tengan una alta probabilidad para poder hallar la clave.

3.1. Funcionamiento

Primero, se aproximan las componentes no lineales de las redes SPN, es decir, las S-boxes. Después, se concatenan para formar una aproximación lineal del cifrado que usaremos para recuperar la clave.

Sea $\mathcal{S} : \mathbb{F}_2^m \to \mathbb{F}_2^m$ una S-box de una red SPN cualquiera. Supongamos que tenemos una aproximación lineal de ella, que tendrá la forma de la Definición 1.51:

$$\alpha \cdot x \oplus \beta \cdot \mathcal{S}(x) = 0 \tag{3.1}$$

donde $\alpha \in \mathbb{F}_2^m$ es la máscara de entrada y $\beta \in \mathbb{F}_2^m$ es la máscara de salida.

Sobre esta aproximación se estudian los siguientes conceptos:

Definición 3.1. *Se denominan* ***soluciones*** *de la ecuación (3.1) al conjunto*

$$s = \{x \in \mathbb{F}_2^m | \alpha \cdot x \oplus \beta \cdot \mathcal{S}(x) = 0\} \in \mathbb{F}_2^m$$

[5] Con algunas modificaciones puede ser un ataque COA

Si sumamos el cardinal del conjunto y su complementario, $|s|+|s^c| = 2^m$, y recordando la transformada de Walsh-Hadamard para funciones booleanas vectoriales, tenemos que [28]:

$$\mathcal{W}(\mathcal{S})(\alpha,\beta) = 2\,|s| - 2^m \in [-2^m, 2^m]$$

Definición 3.2. *Se denomina* ***probabilidad de ocurrencia de una aproximación lineal*** *del tipo (3.1) a*

$$p = \mathbb{P}_x[\alpha \cdot x \oplus \beta \cdot \mathcal{S}(x) = 0] = \frac{|s|}{2^m} \quad \forall x \in \mathbb{F}_2^m$$

Por la relación anterior entre la transformada de Walsh-Hadamard de la S-box y el número de soluciones de la ecuación (3.1), podemos escribir la probabilidad en términos de la transformada:

$$p = \frac{\mathcal{W}(\mathcal{S})(\alpha,\beta)}{2^{m+1}} + \frac{1}{2}$$

Definición 3.3. *Se llama* ***sesgo de probabilidad lineal*** *a la cantidad en la que la probabilidad p de que se cumpla una expresión lineal se desvía de $\frac{1}{2}$.*

$$\varepsilon = p - \frac{1}{2} = \frac{\mathcal{W}(\mathcal{S})(\alpha,\beta)}{2^{m+1}} \begin{cases} < 0 & |s| < \frac{|\mathbb{F}_2^m|}{2} \\ > 0 & |s| > \frac{|\mathbb{F}_2^m|}{2} \end{cases}$$

Además, $-\frac{1}{2} < \varepsilon < \frac{1}{2}$

Si escogiésemos aleatoriamente bits de $x \in \mathbb{F}_2^n$ en la ecuación (3.1), la probabilidad de que la verifiquen ecuación es del $\frac{1}{2}$. Esto se debe a que en una aproximación lineal, cada bit tiene un 50 % de probabilidad de ser 0 y otro 50 % de ser 1. Por tanto, lo que nos interesa saber es cuándo una expresión lineal es lo más diferente a una expresión aleatoria. Esto se mide con el sesgo de probabilidad lineal ε.

Se pueden diferenciar los siguientes casos:

- Si $\varepsilon = 0$, la expresión lineal no nos da información porque la probabilidad p ha sido de $\frac{1}{2}$.
- Si $\varepsilon > 0$, la expresión lineal es útil porque su probabilidad de ocurrencia p es más alta que $\frac{1}{2}$, es decir, ocurre con mayor frecuencia.
- Si $\varepsilon < 0$, la expresión lineal tiene una probabilidad muy baja de ocurrir. Esto también nos da información acerca de la S-box, ya que significa que la misma expresión pero $\oplus 1$ ocurre con una alta probabilidad (la complementaria). Por tanto, la expresión $\oplus 1$ nos es útil por el mismo motivo que en el anterior caso.

Cuanto mayor es la magnitud del sesgo, $|\varepsilon|$, mejor será la aproximación lineal que tenemos (porque más se alejará la expresión de una aleatoria).

Podemos reunir estos tres conceptos en uno mucho más potente e ilustrativo: la **Tabla de Aproximación Lineal** (*Linear Approximation Table*, LAT). Esta tabla de tamaño $2^m \times 2^m$ recoge todas las aproximaciones lineales que pueden hacerse sobre una S-box.

Además podemos deducir de ella las probabilidades de cada aproximación lineal y su sesgo. Cada entrada de la tabla está definida como

$$LAT[\alpha,\beta] = |s| - 2^{m-1} = 2^m\varepsilon = \frac{\mathcal{W}(\mathcal{S})(\alpha,\beta)}{2}$$

Algunas propiedades relevantes acerca de la LAT son [14]:

- La LAT de $\mathcal{S}^{-1}$ es la traspuesta de la LAT de $\mathcal{S}$.
- La primera fila y columna son todo $0's$ excepto la primera entrada $LAT[0,0] = 2^{m-1}$. La razón de ello es que si la máscara de entrada y salida son nulas, la aproximación de la S-box (3.1) es siempre verdadera.
- Todos los números que recoge la LAT son enteros pares no necesariamente potencias de 2.

Ejemplo 3.4. Vamos a ilustrar estos conceptos sobre la S-box del cifrado PRESENT (véase la Figura 2.1). Escojamos como máscaras $\alpha = 9 = (1001)_2$ y $\beta = 1 = (0001)_2$. Calculamos las soluciones s de (3.1). Utilizaremos la notación $\mathcal{S}(x) = y$.

$$\begin{aligned} s &= \left\{x \in \mathbb{F}_2^4 | \alpha \cdot x \oplus \beta \cdot \mathcal{S}(x) = 0\right\} = \\ &= \left\{x \in \mathbb{F}_2^4 | (1001)(x_1,x_2,x_3,x_4) \oplus (0001)(y_1,y_2,y_3,y_4) = 0\right\} = \\ &= \left\{x \in \mathbb{F}_2^4 | x_1 \oplus x_4 \oplus y_4 = 0\right\} = \\ &= \left\{x \in \mathbb{F}_2^4 | x_1 \oplus x_4 = y_4\right\} \end{aligned} \tag{3.2}$$

En el Cuadro 3.1 están recogidas todas las $x \in \mathbb{F}_2^4$ con su correspondiente $\mathcal{S}(x) = y$ y si verifican o no la ecuación (3.2). Vemos que $|s| = 12$. Como consecuencia, la probabilidad de esta aproximación lineal es de $\frac{12}{16}$ y su sesgo es de $\varepsilon = \frac{1}{4}$. En la LAT del PRESENT esta aproximación estará recogida como:

$$LAT[9,1] = 12 - 2^{4-1} = 4$$

Efectivamente, si consultamos el Cuadro 3.2, veremos que para $(9,1)$ el valor es 4.

Ahora que ya sabemos cómo aproximar una S-box, vamos a ver cómo hacer aproximaciones de varias rondas. Para ello, necesitamos el llamado **Lema de apilamiento**.

Lema 3.5 (**Lema de apilamiento**). *Sean n variables aleatorias booleanas independientes $X_1, \ldots, X_n$ cuyas probabilidades de tomar el valor 0 son $p_i = \mathbb{P}(X_i = 0) = \frac{1}{2} + \varepsilon_i$. Entonces,*

$$\mathbb{P}(X_1 \oplus \ldots \oplus X_n = 0) = \frac{1}{2} + 2^{n-1}\prod_{i=1}^{n}\varepsilon_i = \frac{1}{2} + \varepsilon_{1,2,\ldots,n}$$

Notemos por $\varepsilon_{1,2,\ldots,n}$ al sesgo de la expresión $X_1 \oplus \ldots \oplus X_n = 0$.

Demostración. Supongamos que tenemos dos variables aleatorias booleanas independientes X_1 y X_2 cuyas probabilidades son

$$p_1 = \mathbb{P}(X_1 = 0) = \frac{1}{2} + \varepsilon_1 \quad \text{y} \quad p_2 = \mathbb{P}(X_2 = 0) = \frac{1}{2} + \varepsilon_2$$

$x_1x_2x_3x_4$	$y_1y_2y_3y_4$	$x_1 \oplus x_4 = y_4$
0000	1100	Sí
0001	0101	Sí
0010	0110	Sí
0011	1011	Sí
0100	1001	No
0101	0000	No
0110	1010	Sí
0111	1101	Sí
1000	0011	Sí
1001	1110	Sí
1010	1111	Sí
1011	1000	Sí
1100	0100	No
1101	0111	No
1110	0001	Sí
1111	0010	Sí
$\lvert s \rvert$		12

Cuadro 3.1: Aproximación lineal de la S-box del cifrado PRESENT con $\alpha = 9$ y $\beta = 1$

α/β	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	-4	0	-4	0	0	0	0	0	-4	0	4
2	0	0	2	2	-2	-2	0	0	2	-2	0	4	0	4	-2	2
3	0	0	2	2	2	-2	-4	0	-2	2	-4	0	0	0	-2	-2
4	0	0	-2	2	-2	-2	0	4	-2	-2	0	-4	0	0	-2	2
5	0	0	-2	2	-2	2	0	0	2	2	-4	0	4	0	2	2
6	0	0	0	-4	0	0	-4	0	0	-4	0	0	4	0	0	0
7	0	0	0	4	4	0	0	0	0	-4	0	0	0	0	4	0
8	0	0	2	-2	0	0	-2	2	-2	2	0	0	-2	2	4	4
9	0	4	-2	-2	0	0	2	-2	-2	-2	-4	0	-2	2	0	0
A	0	0	4	0	2	2	2	-2	0	0	0	-4	2	2	-2	2
B	0	-4	0	0	-2	-2	2	-2	-4	0	0	0	2	2	2	-2
C	0	0	0	0	-2	-2	-2	-2	4	0	0	-4	-2	2	2	-2
D	0	4	4	0	-2	-2	2	2	0	0	0	0	2	-2	2	-2
E	0	0	2	2	-4	4	-2	-2	-2	-2	0	0	-2	-2	0	0
F	0	4	-2	2	0	0	-2	-2	-2	2	4	0	2	2	0	0

Cuadro 3.2: Tabla de Aproximación Lineal del cifrado PRESENT

Consideremos $X_1 \oplus X_2 = 0$. Esta expresión será verdadera si ambas variables toman el mismo valor. Por ser variables independientes, la probabilidad de dicha expresión es:

$$\begin{aligned} \mathbb{P}(X_1 \oplus X_2 = 0) &= \mathbb{P}(X_1 = X_2) = \\ &= \mathbb{P}(X_1 = 0)\mathbb{P}(X_2 = 0) + \mathbb{P}(X_1 = 1)\mathbb{P}(X_2 = 1) = \\ &= p_1p_2 + (1 - p_1)(1 - p_2) \end{aligned} \tag{3.3}$$

Como las probabilidades p_1 y p_2 pueden ser expresadas en términos del sesgo, la anterior expresión puede escribirse como:

$$\begin{aligned} \mathbb{P}(X_1 \oplus X_2 = 0) &= \left(\frac{1}{2} + \varepsilon_1\right)\left(\frac{1}{2} + \varepsilon_2\right) + \left(1 - \frac{1}{2} - \varepsilon_1\right)\left(1 - \frac{1}{2} - \varepsilon_2\right) = \\ &= \frac{1}{2} + 2\varepsilon_1\varepsilon_2 \end{aligned} \tag{3.4}$$

Podemos denotar como $\varepsilon_{1,2} = \varepsilon_1\varepsilon_2$.

Siguiendo el mismo razonamiento para n variables aleatorias booleanas independientes, obtenemos el enunciado de la proposición. □

Se deduce que si las variables aleatorias $X_1, \ldots, X_n$ tienen probabilidad $p_i = 0$ o $p_i = 1$ $\forall i$, entonces $\mathbb{P}(X_1 \oplus \ldots \oplus X_n = 0) = 0 \quad$ ó $\quad 1$, respectivamente. Además, si sólo hay un $p_i = \frac{1}{2}$, entonces $\mathbb{P}(X_1 \oplus \ldots \oplus X_n = 0) = \frac{1}{2}$.

En general, podemos representar un cifrado por bloques con clave maestra K donde $\varepsilon_K(P) = C$ mediante una aproximación lineal del mismo como:

$$\alpha \cdot P \oplus \beta \cdot C = \kappa \cdot K$$

donde $\kappa \in \mathbb{F}_2^l$ es la máscara para la clave K.

Al igual que con las aproximaciones de las S-boxes, cuanto mayor es la magnitud del sesgo total, mejor será la aproximación lineal.

Antes de ejemplificar cómo construir una aproximación lineal de un cifrado, vamos a introducir el siguiente concepto.

Definición 3.6. *Se llaman **S-boxes activas** a las distintas S-boxes que se van recorriendo en una aproximación lineal, es decir, tienen un texto de entrada no nulo.*

La decisión de cómo comenzar la aproximación lineal recae sobre el atacante. Así, el número de S-box activas y cuáles utilizaremos y con qué textos de entrada en la primera ronda depende de él. Para el resto de rondas, las S-boxes activas (tanto la cantidad como la posición) y sus textos de entradas dependen de la fase de permutación. Lo único que sigue dependiendo del adversario en estas rondas es qué aproximación de las S-boxes (activas) elegir, que depende también de la LAT.

Ejemplo 3.7. Vamos a realizar una aproximación de un par de rondas. Se utilizará la red SPN de Heys [16] que puede consultarse en el Apéndice B.

Primero, se estudian las aproximaciones de las S-boxes que vamos a utilizar. Para ello, Heys decide que la primera S-box que usaremos será la $\mathcal{S}_{1,2}$ (la primera S-box activa) y su aproximación lineal será la correspondiente a $LAT[B, 4] = 4$ (véase la Figura B.3). Esta aproximación es:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

con un sesgo de $\frac{1}{4}$, y, por tanto, una probabilidad de $\frac{12}{16}$.

Ahora, cuál o cuáles son las S-boxes siguientes y qué texto de entrada tendrán dependen de la fase de permutación. En este caso, la salida $(4)_{16} = (0100)_2$ de $\mathcal{S}_{1,2}$ produce un texto de entrada para la fase de permutación $[0\ 4\ 0\ 0]_{16} = [0000\ 0100\ 0000\ 0000]_2$. Recordamos que la permutación se hace sobre las posiciones del texto completo, por lo que, el único 1 que aparece está en posición 6 que permuta a la posición 6 de nuevo (véase la Figura B.2). Como resultado, en este caso, el texto de entrada a las S-boxes de la segunda ronda es idéntico a la texto de entrada de la fase de la permutación. Ahora, la S-box que debemos aproximar ahora es la $\mathcal{S}_{2,2}$ (siguiente S-box activa). Heys decide que la aproximación lineal sea la dada por $LAT[4, 5] = -4$:

$$X_2 = Y_2 \oplus Y_4$$

cuyo sesgo de $-\frac{1}{4}$, y, por tanto, una probabilidad de $\frac{4}{16}$.

Una vez que tenemos estas expresiones, vamos a hacer una la aproximación de las dos rondas del cifrado. Denotaremos por P_j al j-ésimo bit del texto plano, $U_{i,j}$ al j-ésimo bit del texto de entrada de las S-boxes de la i-ésima ronda, $V_{i,j}$ al j-ésimo bit del texto de salida de las S-boxes de la i-ésima ronda y $k_{i,j}$ al j-ésimo bit de la subclave de la ronda i.

El cifrado comienza con una combinación con claves, por lo que,

$$U_1 = P \oplus K_1$$

donde omitimos el subíndice j porque nos referimos al texto completo. Usando la aproximación de la primera ronda tenemos que:

$$U_{1,5} \oplus U_{1,7} \oplus U_{1,8} = V_{1,6}$$

Equivalentemente,

$$P_5 \oplus K_{1,5} \oplus P_7 \oplus K_{1,7} \oplus P_8 \oplus K_{1,8} = V_{1,6}$$

Esta aproximación lineal tiene una probabilidad de $\frac{12}{16}$, como se explica antes.

Por la aproximación que tenemos de la segunda ronda:

$$U_{2,6} = V_{2,6} \oplus V_{2,8}$$

Por la combinación con claves de la segunda ronda tenemos que:

$$U_{2,6} = V_{1,6} \oplus K_{2,6}$$

Por tanto, tenemos una aproximación $V_{1,6} \oplus K_{2,6} = V_{2,6} \oplus V_{2,8}$ que tiene una probabilidad de $\frac{1}{4}$.

Si la combinamos con la aproximación de la primera ronda, obtenemos una expresión de las dos rondas:

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0$$

Aplicando el Lema de apilamiento, tiene una probabilidad de

$$p = \frac{1}{2} + 2\left(\frac{3}{4} - \frac{1}{2}\right)\left(\frac{1}{4} - \frac{1}{2}\right) = \frac{3}{8}$$

Una vez estudiada cómo elaborar una aproximación lineal de diferentes rondas, vamos a ver cómo recuperar algunos bits de la clave mediante ella. Después, volveremos a ilustrar el método con un ejemplo de la misma referencia [16].

Existen diversos métodos para averiguar bits de las subclaves utilizadas en el cifrado. Matsui inventó dos algoritmos para recuperar bits de la clave, siendo más eficiente el llamado *Algoritmo 2* [22]. Si tenemos una red SPN de r rondas, el objetivo de este método es averiguar bits de la última subclave k_{r+1} teniendo una aproximación lineal de $r-1$ rondas. En este trabajo nos centraremos en este método pero otros que podemos mencionar son aquellos que recuperan bits de la primera subclave o de la última pero con aproximaciones lineales de menos rondas ($r-2$ ó $r-3$ rondas son los más comunes). También se aplica en el criptoanálisis diferencial con las adaptaciones pertinentes.

Por tanto, supongamos que tenemos una aproximación lineal de $r-1$ rondas. Esto significa que tendremos una expresión con bits del texto plano, bits del texto de entrada

a las S-boxes de la última ronda y bits de las distintas subclaves que aparecen al hacer la aproximación de varias rondas. Para visualizarla, consúltese la Figura B.1. En realidad, para poder utilizar la aproximación lineal, la suma de los bits de las distintas subclaves se fija como 0 ó 1 (ya que depende de la clave del cifrado), obteniéndose una aproximación donde sólo haya bits de los dos primeros tipos que hemos descrito anteriormente[6].

Definición 3.8. *Se llama* ***subclave parcial objetivo*** *a los bits de la última subclave k_{r+1} del cifrado que están en contacto con las S-boxes de la última ronda.*

Cada subclave parcial objetivo que probemos tiene un contador T_i que se incrementa cuando la aproximación es verdadera con la clave probada. Si tenemos n bits de la subclave parcial objetivo, habrá n^2 contadores T_i.

Observación 3.9. Por abuso de notación, denotaremos por k_{r+1} tanto a la última subclave del cifrado como a la subclave parcial objetivo (son la misma pero ésta última sólo algunos bits como ha sido mencionado).

El método comienza teniendo una cantidad N de parejas de texto plano/texto cifrado. El origen de estas parejas suele ser la generación de N textos planos de forma aleatoria por parte del atacante y, posteriormente, obtiene los correspondientes textos cifrados introduciéndolos en el cifrado.

Por cada pareja, probamos todas las posibles subclaves parcial objetivo k'_{r+1}. Esto es, para una pareja (P, C) y por cada una de las subclaves candidatas k'_{r+1}, desencriptamos parcialmente $C = z_r$ y obtenemos el texto de la iteración anterior $z_{r-1} = z_r \oplus k'_{r+1}$. Sin embargo, en la aproximación lineal del cifrado no son los bits de este texto los que aparecen sino los bits de $\mathcal{S}^{-1}(z_{r-1})$, como se mencionó anteriormente. Introduciendo estos bits, observamos si la aproximación se cumple. Si es así, el contador T_i de k'_{r+1} aumenta.

Después de probar las subclaves candidatas en todas las parejas, calculamos la desviación de cada subclave respecto a la mitad del número de parejas probadas:

$$sesgo = \frac{T_i}{N} - \frac{1}{2}$$

La subclave correcta será aquella cuyo *sesgo* sea más grande. En otras palabras, aquella que difiera en mayor medida de la mitad del número de parejas de texto introducidas. La razón es que la probabilidad de la aproximación lineal con esta subclave será lo más lejana al $\frac{1}{2}$ posible.

El resto de bits de la última subclave k_{r+1} se recomienda obtenerlos mediante un ataque por fuerza bruta o aplicando de nuevo un criptoanálisis lineal con otras aproximaciones.

Obtener bits de la última subclave es útil para obtener la clave maestra de cifrados que tenga su propio algoritmo de generación de subclaves. Cuando hay otros protocolos de generación de subclaves, se utilizan otros métodos de aproximación más útiles.

En este método se hace la siguiente suposición.

Hipótesis (*Hipótesis de la aleatorización de la clave incorrecta*)**.** Si utilizamos una clave incorrecta cuando desencriptamos parcialmente el texto cifrado, la probabilidad de la aproximación lineal será cercana al $\frac{1}{2}$ y, por tanto, obtendremos un sesgo $\varepsilon \simeq 0$. Visto de otro modo, la distribución de los contadores T_i de las claves erróneas tienen una distribución más uniforme.

[6]Consultar ejemplo para mayor aclaración

Gracias a esta hipótesis, algunos autores han podido desarrollar fórmulas estadísticas de la probabilidad de éxito del ataque (véase la Sección 3.2.1).

Ejemplo 3.10. Ilustraremos este método con el ejemplo extraído de [16] donde ataca una red SPN de 4 rondas, construyendo una aproximación lineal de 3 rondas para obtener los bits de la subclave k_5. Puede consultarse en dicha referencia cómo hacer la aproximación lineal pero en este ejemplo partiremos de ella ya construida. En el Ejemplo 3.7 explicamos las dos primeras rondas.

Denotando $k_{i,j}$ al j-ésimo bit de la subclave de la ronda i, $U_{4,j}$ al j-ésimo bit del texto de entrada de las S-boxes de la cuarta ronda y P_j al j-ésimo bit del texto plano, esta aproximación tiene la forma:

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \sum k = 0$$

$$\sum k = k_{1,5} \oplus k_{1,7} \oplus k_{1,8} \oplus k_{2,6} \oplus k_{3,6} \oplus k_{3,14} \oplus k_{4,6} \oplus k_{4,8} \oplus k_{4,14} \oplus k_{4,16}$$

Según los cálculos de Heys y aplicando el Lema de apilamiento, esta expresión tiene una probabilidad de $p = \frac{15}{32}$. Como la suma de los bits de las subclaves de las distintas rondas es fija (pero no la conocemos, puede ser 0 o 1), entonces tenemos que (con la misma notación para los subíndices):

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0$$

Esta aproximación lineal tiene una probabilidad de $p = \frac{15}{32}$ o $p = 1 - \frac{15}{32} = \frac{17}{32}$, dependiendo de si $\sum k$ es 0 o 1. En cualquier caso, la magnitud del sesgo de esta aproximación es de $|\varepsilon| = \frac{1}{32}$. Véase la Figura B.1.

Podemos ver en la Figura B.1 que las S-boxes activas son $\mathcal{S}_{1,2}$, $\mathcal{S}_{2,2}$, $\mathcal{S}_{3,2}$, $\mathcal{S}_{3,4}$.

La subclave parcial objetivo es $k_5' = [0000\ k_{5,5} \dots k_{5,8}\ 0000\ k_{5,13} \dots k_{5,16}]_2$ según nos indican los subíndices de $U_{4,j}$. Por tanto, hay que probar para N parejas, $16^2 = 256$ subclaves que son: $[0\ 1\ 0\ 1], [0\ 1\ 0\ 2], \dots, [0\ E\ 0\ F], [0\ F\ 0\ F]$[7].

Heys halla, después de probar $N = 10000$ parejas, que el contador de la clave [0 2 0 4] fue de $T_{21} = 5336$. Esto resulta en un $sesgo = 0{,}0336 \simeq |\varepsilon| = \frac{1}{32}$.

3.2. Aspectos del ataque

Si recordamos el concepto de S-boxes activas (véase Definición 3.6), se ha demostrado que la probabilidad p de una expresión lineal depende de la cantidad de S-boxes que tenga y el sesgo de probabilidad lineal de cada una, ε_i. De hecho, cuantas menos S-boxes activas haya, mayor será la magnitud del sesgo de la aproximación global $|\varepsilon|$ [16].

Subclaves erróneas

En la práctica, este algoritmo puede conducir a una subclave errónea ya que podría haber otras con *sesgos* elevados y no ser las correctas. Las principales causas de esto son:

1. Las propiedades particulares de la S-box que influyen en la desencriptación parcial.

[7] Notación hexadecimal

2. Aunque generalmente funciona bien, el Lema de apilamiento puede causar cierta imprecisión al asumir la independencia de las aproximaciones entre rondas.

Acerca de las aproximaciones lineales

Una aproximación lineal (α, β) de una red SPN de r rondas puede ser descompuesta en r aproximaciones lineales que corresponden a los pasos intermedios

$$(\alpha, \beta) = [(\alpha, v_1), (v_1, v_2), \ldots, (v_{r-1}, \beta)]$$

donde $v_i \in \mathbb{F}_2^n$ representan dichos pasos. Si fijamos pasos (distintos entre sí), cada aproximación lineal recibe el nombre de **secuencia lineal** (*linear trail*). Es decir, tenemos diversas formas de obtener (α, β). La colección de todas las secuencias lineales de (α, β) se llama **casco lineal** (*linear hull*). Diferentes secuencias lineales (α, β) tienen diferentes probabilidades. La secuencia lineal que tenga la mayor probabilidad recibirá el nombre de **secuencia dominante** y será la que debamos utilizar. Aunque es común su existencia también puede darse que todas las secuencias lineales de un casco tengan bajas probabilidades pero su combinación resulte en una expresión lineal con un sesgo elevado.

La existencia del casco lineal influye a la hora de evaluar la seguridad de un cifrado contra el criptoanálisis lineal. No por tener secuencias lineales con bajas probabilidades significará que el cifrado estará seguro contra este tipo de ataque. En resumen, hay que tomar en consideración que las posibles combinaciones de las secuencias produzcan un riesgo en el cifrado.

Datos necesarios

Matsui demostró que el número de textos planos conocidos que se necesitan para atacar se puede aproximar como $\frac{1}{\varepsilon^2}$ siendo ε el sesgo de la aproximación lineal del cifrado.

En el Ejemplo 3.10, Heys [16] eligió probar 10000 textos planos para realizar el criptoanálisis pero, según Matsui, dado que $\varepsilon = \frac{15}{32}$, le hubiera bastado $\frac{1}{\varepsilon^2} = 1024$.

Resistencia de los cifrados

En la actualidad, los cifrados se protegen del criptoanálisis lineal tomando medidas como usar S-boxes con sesgos pequeños[8] y estructuras que aumenten el número de S-boxes activas para obtener expresiones lineales con probabilidades muy bajas. Sin embargo, no podemos pensar que la seguridad resida en evitar expresiones lineales con probabilidades altas. Existen dos motivos principalmente: las aproximaciones de las S-boxes, en realidad, no son independientes (y esto influye en la probabilidad) y la existencia del casco lineal.

3.2.1. Probabilidad de éxito

En esta sección indicaremos brevemente cómo calcular la probabilidad de éxito de este ataque y los recursos en términos de datos para ello.

Como se ha mencionado anteriormente, uno de los inconvenientes de este algoritmo es que pueda haber varias subclaves con un *sesgo* alto. Para solucionarlo, las subclaves

[8] El sesgo, por el Lema de apilamiento, depende de cada aproximación de la S-box

se clasifican en una lista de mayor a menor sesgo y se van probando hasta cierto sesgo. Si queremos recuperar una subclave de m bits y la clave correcta k_0 está en el puesto r entre las 2^m posibles, entonces el ataque se reduce en un factor $2^{m-log(r)}$ en términos de recursos sobre el ataque por fuerza bruta [29]. Al número de bits $m-log(r)$ lo llamaremos **ventaja dada por el ataque**.

Diversos autores han dado fórmulas para la probabilidad del éxito del ataque lineal [29, 5]. En el caso de Selçuk, desarrolló una fórmula que aproxima la probabilidad de éxito P_S teniendo como datos la cantidad de textos planos N y el sesgo de la aproximación lineal ε. A continuación se enuncia dicha fórmula que puede encontrarse demostrada en [29] de manera teórica y corroborada en la práctica.

Proposición 3.11. *[29] Sea un cifrado por bloques de longitud n y tamaño de la clave l. Sea una aproximación lineal con un sesgo ε y una cantidad de textos planos $N \leq 2^n$. Si P_S es la probabilidad de éxito de este método con una ventaja de a-bits sobre el ataque por fuerza bruta, entonces:*

$$P_S = \Phi\left(2\sqrt{N}|\varepsilon| - \sqrt{1 + \frac{N}{2^n}}\Phi^{-1}(1 - 2^{-a-1})\right)$$

donde Φ es la función de distribución acumulada de la distribución normal estándar.

Como corolario de esta proposición, Selçuk también aproximó el número de textos planos necesarios para conseguir una ventaja de a-bits con una probabilidad de éxito P_S y sesgo ε.

Corolario 3.12. *[29] Bajo las mismas hipótesis que la Proposición anterior, el número de textos planos necesarios en un ataque lineal para obtener una ventaja de a-bits con una probabilidad de éxito P_S es de*

$$N = \left(\frac{\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-a-1})}{2}\right)^2 \cdot \varepsilon^{-2}$$

3.3. PRESENT

Los autores del cifrado PRESENT realizaron un análisis de seguridad del mismo frente algunos criptoanálisis conocidos [6]. En el caso del criptoanálisis lineal, demostraron una cota superior para el sesgo de una aproximación lineal de 4 rondas.

Proposición 3.13. *[6] Sea ε_{4R} el sesgo máximo de una aproximación lineal de 4 rondas. Entonces,*

$$\varepsilon_{4R} \leq 2^{-7}$$

La demostración del teorema puede consultarse en el Apéndice F. Esta cota superior limita obtener una buena aproximación lineal, en particular, de un número de rondas múltiplo de 4.

Por otro lado, en [25] estudian cómo la debilidad de la fase de permutación del PRESENT hace que sean posibles las relaciones lineales iterativas para cualquier número

de rondas. Además, afirman que el sesgo de una aproximación lineal de 3 rondas también está acotado superiormente por 2^{-7}.

Uno de los mejores criptoanálisis lineales conocidos contra el PRESENT se puede consultar en la referencia [15] donde atacan 26 y 27 rondas contra el PRESENT con longitud de clave 80 y el primer ataque a 28 rondas del PRESENT con longitud de clave 128. En la siguiente imagen se resumen sus ataques siendo κ la notación para la longitud de la clave. La cantidad de datos está medida en número de textos planos y el tiempo en encriptaciones.

Rounds	**Appr.**	**Capacity**	**Data**	**Time** ($\kappa = 80$)	**Time** ($\kappa = 128$)	**Memory**	P_S	**Source**
26	2995†	$2^{-55.58}$	$2^{63.8}$	2^{70}	2^{118}	2^{34}	0.95	[Ch10]
	135	$2^{-55.47}$	2^{62}	2^{72}	2^{120}	2^{48}	0.95	[BTV18]
	128	$2^{-54.81}$	$2^{61.9}$	2^{65}	2^{113}	2^{44}	0.95	This
	128	$2^{-54.81}$	$2^{60.8}$	2^{72}	2^{120}	2^{44}	0.95	This
27	405†	$2^{-55.33}$	$2^{63.8}$‡	2^{77}	2^{125}	2^{70}	0.95	[ZZ15]
	135	$2^{-58.06}$	$2^{63.8}$	$2^{77.5}$	$2^{125.5}$	2^{48}	0.95	[BTV18]
	128	$2^{-56.71}$	$2^{63.4}$	2^{72}	2^{120}	2^{44}	0.95	This
28	448	$2^{-56.98}$	2^{64}	-	2^{122}	2^{89}	0.95	This

†: Multidimensional linear cryptanalysis is used.

‡: Only one fourth of the known plaintexts is effectively used for each key guess.

Cuadro 3.3: Criptoanálisis lineal aplicado al PRESENT por [15]. Los resultados del artículo están indicados por *This*. El resto pertenecen a otros trabajos previos

Capítulo 4

Criptoanálisis diferencial

En este capítulo desarrollaremos el criptoanálisis diferencial aplicado a redes SPN. Usaremos las referencias bibliográficas [16, 4, 30, 29, 5, 28].

El criptoanálisis diferencial es un ataque de tipo CPA publicado por Eli Biham y Adi Shamir en 1990. Fue aplicado primero contra una versión reducida del DES y posteriormente contra el DES completo, siendo más rápido que el ataque por fuerza bruta.

4.1. Funcionamiento

En la introducción de este trabajo se dio una pequeña idea en términos generales. Para ahondar más en este ataque necesitamos la siguiente definición.

Definición 4.1. *Sean $x^{'}$, $x^{''} \in \mathbb{F}_2$ cualesquiera. Se define la* ***diferencia entre $x^{'}$ y $x^{''}$*** *como*

$$\Delta x := x^{'} \oplus x^{''}$$

Definición 4.2. *Sean $X^{'} = (X_1^{'}, \ldots, X_n^{'})$, $X^{''} = (X_1^{''}, \ldots, X_n^{''}) \in \mathbb{F}_2^n$ cualesquiera. Se define la* ***diferencia entre $X^{'}$ y $X^{''}$*** *como*

$$\Delta X := X^{'} \oplus X^{''} = (\Delta X_1, \ldots, \Delta X_n) \in \mathbb{F}_2^n$$

donde ΔX_i es la diferencia entre $X_i^{'}$ y $X_i^{''}$ $\forall\, i \in \{1, \ldots, n\}$.

La notación utilizada ΔX es la estándar en la literatura y puede considerarse un abuso de notación. En este trabajo también se hará uso de ella.

Observación 4.3. Sean $X^{'}, X^{''}, Y^{'}, Y^{''} \in \mathbb{F}_2^n$ y consideremos sus diferencias ΔX y ΔY, respectivamente. Se cumple que:

- Si π es una permutación, entonces $\pi(\Delta X) = \pi(X) \oplus \pi(X^{'})$
- $\Delta X \oplus \Delta Y = (X \oplus Y) \oplus (X^{'} \oplus Y^{'}) \equiv \Delta(X \oplus Y)$

Proposición 4.4. *Sea un cifrado por bloques y consideremos dos textos $X^{'}, X^{''}$ y una subclave k. La combinación $X^{'} \oplus k$ y $X^{''} \oplus k$ no afecta a la diferencia ΔX.*

Demostración. Sea X', X'' dos textos cualesquiera, k una subclave e Y', Y'' el resultado de combinar los textos con la clave, es decir, $Y' = X' \oplus k$, $Y'' = X'' \oplus k$. Tenemos que ver que $\Delta X = \Delta Y$.

$$\Delta Y = Y' \oplus Y'' = (X' \oplus k) \oplus (X'' \oplus k) = X' \oplus X'' = \Delta X$$

ya que $k \oplus k = 0$ y la operación XOR es conmutativa y asociativa. □

En un cifrado por bloques de tamaño n idealmente aleatorio la probabilidad de que dada una diferencia en el texto de entrada ΔP provoque una diferencia concreta ΔC en los textos de salida es de $\frac{1}{2^n}$. La razón es que la probabilidad de que en cada posición haya un 0 o un 1 es de $\frac{1}{2}$ para ambas opciones y, como hay n bits y son sucesos independientes entre bits, tenemos que $\frac{1}{2} \cdot \overset{n}{\ldots} \cdot \frac{1}{2} = \frac{1}{2^n}$.

Definición 4.5. *Se llama **diferencial** al par $(\Delta X, \Delta Y) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ donde ΔX es una diferencia en los textos de entrada que provoca la diferencia ΔY en los textos de salida.*

La diferencia de entrada ΔX no siempre provocará la misma diferencia de salida ΔY como veremos más adelante. Dicho de otro modo, puede haber distintas parejas $(\Delta X, \Delta Y)$ con el mismo ΔX pero distinto ΔY. Es por ello, que el diferencial tiene una probabilidad p asociada. Esta probabilidad nos dice cuánto de frecuente es que dado ΔX resulte en ΔY. Habrá diferenciales con probabilidad nula, es decir, que nunca se darán ya que dada la diferencia ΔX, jamás provocará la diferencia ΔY.

El criptoanálisis diferencial es un método que intenta encontrar diferenciales con probabilidades p muy altas y, con ello, obtener la clave. En general, si tenemos un cifrado por bloques de r rondas, vamos a estudiar un diferencial donde tengamos una diferencia en los textos de entrada ΔP y una diferencia en los textos de salida de la ronda $r-1$, $\Delta C'$. Con este diferencial $(\Delta P, \Delta C')$, procederemos de la misma forma que en el criptoanálisis lineal, desencriptando parcialmente y averiguando bits de la última subclave k_{r+1} del cifrado. Para construirlo, hay que explorar las distintas características diferenciales, que definimos a continuación.

Definición 4.6. *Se define la **característica diferencial de n rondas** $\Delta = (\Delta X_0, \ldots, \Delta X_n)$ a la secuencia de $n+1$ diferencias de entrada a la fase de sustitución de cada ronda del cifrado. ΔX_0 es la diferencia del texto plano de origen ΔP mientras que ΔX_n es la diferencia de entrada a la ronda $n+1$.*

Las diferencias de entrada y salida mencionadas se refieren al texto completo del cifrado. Sin embargo, recordamos que ese texto se divide en bloques para entrar en las distintas S-boxes del cifrado. La notación utilizada en los conceptos de diferencia, diferencial y característica diferencial será la misma tanto si estamos hablando de S-boxes como de textos completos.

De manera similar al criptoanálisis lineal, introducimos el siguiente concepto.

Definición 4.7. *Se definen las **S-boxes activas** como las S-boxes donde la característica diferencial indica que hay un diferencial no nulo en la entrada de las mismas.*

Primero, estudiaremos las propiedades diferenciales de una S-box cualquiera para buscar diferenciales con una alta probabilidad. Después, combinamos estos diferenciales de

una ronda a otra, con el fin de obtener una característica diferencial con una probabilidad alta.

Sea $\mathcal{S} : \mathbb{F}_2^m \to \mathbb{F}_2^m$ una S-box de una red SPN cualquiera. Las propiedades diferenciales que tiene $\mathcal{S}$ pueden ser descritas de dos formas (que detallaremos seguidamente): sabiendo la diferencia de entrada ΔX y el valor de un texto $X^{'}$, conocemos la diferencia de salida ΔY y sabiendo la diferencia de entrada y salida $(\Delta X, \Delta Y)$, conocemos algo del valor $X^{'}$.

En el primer caso mediante la derivada direccional de $\mathcal{S}$, definida en la Definición 1.53:

$$\mathcal{S}_{\Delta X}(X^{'}) = \mathcal{S}(X^{'} \oplus \Delta X) \oplus \mathcal{S}(X^{'}) = \Delta Y$$

En el segundo caso, los valores $X^{'}$ que verifican el diferencial reciben el nombre de **soluciones del diferencial** $(\Delta X, \Delta Y)$:

$$s = \left\{ X^{'} \in \mathbb{F}_2^n \mid \mathcal{S}_{\Delta X}(X^{'}) = \Delta Y \right\}$$

Además, cuando consideremos parejas de textos de entrada $(X^{'}, X^{''})$ tales que $\Delta X = X^{'} \oplus X^{''}$, al ser una operación conmutativa, podemos considerar sólo los 2^m valores que puede tomar $X^{'}$ ya que $X^{''} = X^{'} \oplus \Delta X$.

Ejemplo 4.8. Consideramos la S-box $\mathcal{S}$ del cifrado PRESENT (véase la Figura 2.1), la diferencia de entrada $\Delta X = (B)_{16} = (1011)_2$ y un texto $X = (0)_{16} = (0000)_2$. Entonces,

$$(X, X^{'} = X \oplus \Delta X) = (0000, 0000 \oplus 1011)_2 = (0000, 1011)_2 \equiv (0, B)_{16}$$

Luego,

$$(Y, Y^{'}) = (\mathcal{S}(X), \mathcal{S}(X^{'})) = (C, 8)_{16} = (1100, 1000)_2 \Rightarrow \Delta Y = (1100)_2 \oplus (1000)_2 = (0100)_2$$

En la siguiente tabla se recogen todas las diferencias de salida que dan lugar la pareja $(X, X^{'} = X \oplus \Delta X)$ donde $\Delta X = (B)_{16}$. La primera fila corresponde al ejemplo descrito.

X	$X^{'}$	Y	$Y^{'}$	ΔY
0000	1011	1100	1000	0100
0001	1010	0101	1111	1010
0010	1001	0110	1110	1000
0011	1000	1011	0011	1000
0100	1111	1001	0010	1011
0101	1110	0000	0001	0001
0110	1101	1010	0111	1101
0111	1100	1101	0100	1001
1000	0011	0011	1011	1000
1001	0010	1110	0110	1000
1010	0001	1111	0101	1010
1011	0000	1000	1100	0100
1100	0111	0100	1101	1001
1101	0110	0111	1010	1101
1110	0101	0001	0000	0001
1111	0100	0010	1001	1011

Cuadro 4.1: Diferencias ΔY dada $\Delta X = (B)_{16}$

Por ejemplo, se deduce que el número de ocurrencias de $\Delta Y = (8)_{16}$ dado $\Delta X = (B)_{16}$ es 4, por tanto, la probabilidad es de $\frac{4}{16}$.

Todas las soluciones s de una S-box $\mathcal{S}$ están recogidas en una tabla llamada **Tabla de Distribución de Diferencias** (*Difference distribution table*, DDT). Cada entrada de la tabla está definida como

$$DDT[(\Delta X, \Delta Y)] = |s| \quad \forall \Delta X, \Delta Y$$

Una entrada de esta tabla se puede leer como: hay $DDT[(\Delta X, \Delta Y)]$ pares de parejas de texto de entrada (X', X'') con una diferencia ΔX que dan lugar a la diferencia ΔY.

$\Delta X/\Delta Y$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

Cuadro 4.2: Tabla de Distribución de Diferencias del cifrado PRESENT

Por tanto, podemos calcular todas las probabilidades de que una diferencia de entrada ΔX dé lugar a una diferencia de salida ΔY:

$$\mathcal{P}\left[\mathcal{S}_{\Delta X}(X) = \Delta Y\right] = \frac{DDT[(\Delta X, \Delta Y)]}{2^m}$$

Ejemplo 4.9. En el Ejemplo 4.8 vimos que el diferencial $(4, B)_{16}$ ocurre 4 veces. Podemos ver en la $DDT[(4, B)]$ (véase la Figura 4.2) que, efectivamente, es 4.

Definición 4.10. *Se llama* ***uniformidad diferencial de una S-box*** $\mathcal{S}$ *al máximo valor que aparece en la tabla exceptuando la entrada de* $(\Delta X = 0, \Delta Y = 0)$

$$du_{\mathcal{S}} = \max DDT[(\Delta X, \Delta Y)] \quad con \quad \Delta X \neq 0$$

La uniformidad diferencial es determinante a la hora de evaluar la resistencia de un cifrado frente a este criptoanálisis ya que indica la máxima probabilidad de que dada una diferencia de entrada dé lugar a cierta diferencia de salida.

Observación 4.11. Se deduce que $\mathcal{P}\left[\mathcal{S}_{\Delta X}(X) = \Delta Y\right] \leq \dfrac{du_{\mathcal{S}}}{2^m}$

Ejemplo 4.12. La uniformidad diferencial de la S-box del cifrado PRESENT es $du_{\mathcal{S}} = 4$. Esto significa que un diferencial de una ronda tendrá como máximo una probabilidad de $\frac{4}{16} = 0{,}25$.

Algunas propiedades relevantes acerca de la DDT son [16, 30, 14]:

- Todos los elementos de la tabla son pares. Esto se debe a que un par de textos de entrada (X', X'') tiene el mismo ΔX que (X'', X'). Sus diferencias de salida también coinciden independientemente del orden.
- La suma de todos los elementos de una fila (o de una columna) debe ser 2^m. En el Ejemplo 4.8 es $2^4 = 16$.
- La primera entrada, que corresponde a $(\Delta X, \Delta Y) = (0, 0)$, es siempre igual a 2^m porque si los dos textos de entrada son iguales ($\Delta X = 0$), entonces los textos de salida también son iguales ($\Delta Y = 0$).
- Si tuviésemos una S-box ideal, en el sentido de que sus diferenciales no dan información, entonces todos los elementos serían 1 y la probabilidad de ocurrencia de cada uno sería $\frac{1}{2^m}$. Por las propiedades enumeradas anteriormente, esto no es posible.
- Como la combinación con claves no afecta a las diferencias (Proposición 4.4), la DDT de una S-box es igual que la DDT de una S-box que tiene en cuenta la clave.

Como se mencionó en el Capítulo 2, las S-boxes eran el único componente no lineal de las redes SPN y, por ello, son estudiadas para construir el ataque. Sin embargo, las otras partes del cifrado, consideradas como lineales o afines, también son estudiadas desde el punto de vista del criptoanálisis diferencial. El principal objetivo es examinar el número de S-boxes activas que hay a lo largo del cifrado. Sin embargo, este análisis se hace al diseñar el cifrado para que sea resistente contra el criptoanálisis diferencial pero no se realiza cuando estamos desarrollando el ataque. El número de S-boxes activas a lo largo del cifrado tiene su relevancia, como se explicará en secciones siguientes. En el trabajo, la inclusión de este concepto se ha hecho para resaltar la relevancia de estas partes, en nuestro caso, la fase de permutación[9].

Proposición 4.13. *La diferencia de entrada, ΔX, en una función booleana afín f siempre dará un misma diferencia de salida.*

Demostración. Si $f : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2$ es una función booleana afín, entonces $f(x) = l_a(x) \oplus a_0$ con $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ y $a_0 \in \mathbb{F}_2$ (véase la Observación 1.26).

Recordamos la definición de derivada direccional $f_{\Delta X}(x)$ de f en ΔX (véase la Definición 1.52),

$$f_{\Delta X}(x) = f(x \oplus \Delta X) \oplus f(x) = l(x \oplus \Delta X) \oplus a \oplus l(x) \oplus a = l(x) \oplus l(\Delta X) \oplus l(x) = l(\Delta X)$$

El resultado $l(\Delta X)$ es un valor único, en el sentido de que, siempre que utilicemos la misma diferencia de entrada ΔX, obtendremos la misma diferencia de salida $l(\Delta X)$. Por tanto, ese valor es el único que ocurrirá, es decir, tiene una probabilidad 1 de ocurrir. El resto de diferencias de salida que se puedan contemplar tienen una probabilidad nula de ocurrencia. □

Por ello, siempre es predecible la salida de este tipo de funciones. La pregunta que surge es cuándo una fase lineal es buena, en el sentido de, cuándo una fase lineal maximiza el número de S-boxes activas. Para ello, utilizamos el concepto del número de ramas [11].

[9]Por ser una permutación de bits

Definición 4.14. *Se llama* ***número de ramas*** *(Branch number, $\mathcal{B}$) al mínimo número de S-boxes activas que hay en dos rondas consecutivas.*

Observación 4.15. Como mínimo, $\mathcal{B} = 2$. Esto ocurre porque, si hay una diferencia de entrada no nula, entonces hay, al menos, una S-box activa en una ronda y otra S-box activa en la siguiente [14].

Observación 4.16. En el mejor de los casos es

$$\mathcal{B}^* = 1 + \text{número de S-boxes por ronda}$$

Esto significa que si, por ejemplo, sólo se tiene una S-box activa en una ronda y el número de ramas es $\mathcal{B}^*$, entonces en la siguiente ronda todas las S-boxes son activas. De esta forma, aumentamos la dispersión de los bits y el número de aproximaciones lineales, que conlleva una disminución de la probabilidad de la aproximación total [11].

Para conseguir un buen número de ramas $\mathcal{B}$ no sólo hacen falta permutaciones, sino también otro tipos de fases que no trataremos en este trabajo. Se menciona al AES que tiene una fase en la que se mezclan distintos bits y no es ni una permutación ni una combinación con subclaves.

Ejemplo 4.17. El número de ramas del cifrado PRESENT es $\mathcal{B} = 3$. Por ello, la dispersión de los bits de este cifrado es pobre. A pesar de esto y, en este caso, no es un punto débil tan relevante como se espera por el número de rondas y las propiedades de la S-box utilizada en este cifrado.

Una vez que tenemos información de las propiedades de los diferenciales de nuestras S-boxes en una red SPN, vamos a usarlas para hacer una característica diferencial del cifrado. Esto es, vamos a concatenar diferenciales a lo largo de las rondas. El objetivo será crear una característica diferencial Δ de las primeras $r-1$ rondas de un cifrado de r rondas.

Sea una red SPN de r rondas con un tamaño de bloque n y $\mathcal{S} : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2^m$ una aplicación S-box que es igual para todo el cifrado. En la primera ronda, elegiremos el diferencial $(\Delta X_1, \Delta Y_1)$ que mayor probabilidad tenga mediante la DDT y una S-box $\mathcal{S}_{1,j}$ donde $j \in \left\{0, \ldots, \frac{n}{m}\right\}$.

En la segunda ronda, debemos tener en cuenta que los bits han sido permutados y mezclados con la clave (esto no influye en los diferenciales por la Proposición 4.4) y la permutación se hace por posiciones. Por tanto, hay que observar dónde se localizan los bits no nulos para saber cuál es la diferencia de entrada ΔX_2 que vamos a tener y la S-box $\mathcal{S}_{2,j}$ que se va a utilizar. De nuevo, se observa qué diferencia de salida es más probable consultando la DDT en la fila de ΔX_2 y así sucesivamente.

En resumen, en la ronda i, podremos determinar las S-boxes activas y sus correspondientes entradas al analizar las S-boxes activas, sus salidas y la permutación de la ronda $i-1$.

En cualquier ronda, el resto de S-boxes que no se mencionan tienen un diferencial de entrada nulo, por tanto, un diferencial de salida nulo. Este proceso se hace con las $r-1$ rondas del cifrado.

La probabilidad de que se produzca la característica diferencial completa p_D se calcula asumiendo independencia entre los diferenciales de las S-boxes activas:

$$p_D = \prod_{i=1}^{n_{\mathcal{S}}} p_i$$

donde $n_{\mathcal{S}}$ es el número de S-boxes activas y p_i las probabilidades de cada diferencial en la i-ésima S-box activa.

Observación 4.18. No todos los pares de texto que introduzcamos van a cumplir la característica diferencial aunque cumplan la diferencia de entrada.

Definición 4.19. *Se denominan **pares correctos** (respecto de una característica diferencial Δ) a los pares de texto plano (P, P') que satisfacen Δ. Se denominan **pares incorrectos** a los pares de texto plano (P, P') que no satisfacen Δ.*

Si tenemos N parejas de texto plano para realizar el ataque, entonces una fracción p_D de las parejas van a ser pares correctos. Denotamos por μ al número esperado de parejas correctas.

$$\mu = p_D \cdot N$$

A continuación ilustramos este proceso con un ejemplo extraído de [16] donde el autor, Heys, construye una característica diferencial de 3 rondas que, posteriormente, utilizará para atacar la red SPN de 4 rondas que hemos utilizado como ejemplo en el Capítulo 3.

Ejemplo 4.20. La característica diferencial que construye comienza con el diferencial $(\Delta X, \Delta Y) = (B, 2)_{16}$ que tiene una probabilidad $p = \frac{8}{16}$, la más alta, según la DDT de la S-box que utiliza (véase la Figura B.4) y la S-box activa $\mathcal{S}_{1,2}$. La diferencia $\Delta X = (B)_{16}$ ha sido elegida por el autor, basándose en que la uniformidad diferencial de la S-box de su cifrado es $du_{\mathcal{S}} = 8$ y corresponde a $\Delta Y = (2)_{16}$. Por otro lado, la elección de la S-box activa ha sido arbitraria.

Según su S-box y su fase de permutación (véase la Figura B.2), la característica para las 3 rondas es:

$\mathcal{S}_{1,2}$:	$\Delta X = (B)_{16}$	$\rightarrow$	$\Delta Y = (2)_{16}$	con $p_1 = \frac{8}{16}$
π_1:	$[0000\ 0010\ 0000\ 0000]_2$	$\rightarrow$	$[0000\ 0000\ 0100\ 0000]_2$	
$\mathcal{S}_{2,3}$:	$\Delta X = (4)_{16}$	$\rightarrow$	$\Delta Y = (6)_{16}$	con $p_2 = \frac{6}{16}$
π_2:	$[0000\ 0000\ 0110\ 0000]_2$	$\rightarrow$	$[0000\ 0010\ 0010\ 0000]_2$	
$\mathcal{S}_{3,2}$:	$\Delta X = (2)_{16}$	$\rightarrow$	$\Delta Y = (5)_{16}$	con $p_3 = \frac{6}{16}$
$\mathcal{S}_{3,3}$:	$\Delta X = (2)_{16}$	$\rightarrow$	$\Delta Y = (5)_{16}$	con $p_4 = \frac{6}{16}$
π_3:	$[0000\ 0101\ 0101\ 0000]_2$	$\rightarrow$	$[0000\ 0110\ 0000\ 0110]_2$	

Veamos cómo se halla la segunda ronda. La salida de la primera ronda es $\Delta Y = (2)_{16} = (0010)_2$, es decir, el texto de salida es $[0000\ 0010\ 0000\ 0000]_2 = [0\ 2\ 0\ 0]_{16}$. Por la permutación de la red, se convierte en $[0000\ 0000\ 0100\ 0000]_2 = [0\ 0\ 4\ 0]_{16}$. Entonces, en la segunda ronda la S-box activa es $\mathcal{S}_{2,3}$ cuya diferencia de entrada es $\Delta X = (4)_{16}$. Heys consulta su DDT (véase la Figura B.4) y determina que la diferencia de salida más probable será $\Delta Y = (6)_{16}$, ya que $DDT[(4, 6)] = 6$ (la más alta). Este mismo proceso se

repite hasta el número de rondas que se desee aproximar (en este ejemplo ha sido hasta la ronda 3).

Al final, la característica diferencial construida tiene una probabilidad de

$$p_D = \frac{8}{16} \cdot \frac{6}{16} \cdot \frac{6}{16} \cdot \frac{6}{16} = \frac{27}{1024}$$

En resumen, el diferencial de las 3 rondas es

$$(\Delta X, \Delta Y) = (0B00, 0606)_{16}$$

Y la característica diferencial

$$\Delta = (0B00, 0040, 0220, 0606)_{16}$$

Una vez que tenemos la característica diferencial Δ de $r-1$ rondas con una probabilidad suficientemente alta p_D, vamos a utilizarla para recuperar bits de la clave. Después, volveremos a ilustrar el método con un ejemplo de la misma referencia [16].

Biham y Shamir procedieron igual que Matsui con el *Algoritmo 2* [4]: desencriptar parcialmente la última ronda para determinar si el par ha sido correcto o no con una probabilidad p_D. Por tanto, si tenemos una red SPN de r rondas, el objetivo del método es averiguar bits de la última subclave k_{r+1} teniendo una característica diferencial de $r-1$ rondas. En realidad, este escenario es ideal. En la práctica, muchos cifrados no van permitir hacer tal característica diferencial con una probabilidad suficientemente alta. Suele hacerse en versiones reducidas del cifrado que, de igual forma, debilitan la seguridad del mismo aunque no sea en su versión completa. En este trabajo nos centraremos en dicho escenario ideal para ilustrar el proceso.

Definición 4.21. *Se llama* ***subclave parcial objetivo*** *a los bits de la última subclave k_{r+1} del cifrado que corresponden a las últimas S-boxes activas.*

Cada subclave parcial objetivo tiene un contador T_i que aumenta cada vez que un par de textos cumplen la característica. La tabla que recoge los contadores de cada subclave tiene n^2 entradas siendo n el número de bits que vamos a intentar averiguar.

Observación 4.22. Por abuso de notación, denotaremos por k_{r+1} tanto a la última subclave del cifrado como a la subclave parcial objetivo (son la misma pero ésta última sólo algunos bits como se mencionó).

Supongamos que se ha construido una característica diferencial $\Delta = (\Delta X_0 = \Delta P, \ldots, \Delta X_{r-1})$ de $r-1$ rondas con una probabilidad p_D suficientemente alta. El método comienza teniendo una cantidad N de pares de texto plano (P', P'') que cumplen la diferencia de entrada ΔP marcada por la característica Δ. El origen de los textos planos P' suele ser la generación aleatoria por parte del atacante y los textos P'' son calculados como $P'' = P' \oplus \Delta P$.

El cifrado es conocido, en el sentido de que, se conoce todo el funcionamiento menos la clave. Funciona como una caja negra: podemos introducir textos de entrada y obtener textos de salida. Por tanto, podemos procesar estas parejas de texto plano para obtener también N parejas de texto cifrado (C', C''). Por cada pareja de texto cifrado (C', C''), desencriptamos parcialmente con todas las posibles subclaves parciales objetivo k'_{r+1}.

Esta desencriptación se realiza como sigue: con $C' = z_r = z_{r-1} \oplus k'_r$ y la subclave candidata k'_{r+1}, obtenemos la salida de la última ronda de las S-boxes $z_{r-1}^{C'} = C' \oplus k'_{r+1}$. Después, calculamos $\mathcal{S}^{-1}(z_{r-1}^{C'})$ que es el texto de entrada a la S-boxes de la última ronda. Hacemos el mismo procedimiento con la misma subclave candidata pero con el texto cifrado C''. Al final, tenemos los dos textos de entrada a las S-boxes de la última ronda, $\mathcal{S}^{-1}(z_{r-1}^{C'})$ y $\mathcal{S}^{-1}(z_{r-1}^{C''})$ a los que hacemos la diferencia. Si coinciden con la diferencia última de la característica Δ, entonces aumentamos el contador T_j de k'_{r+1} (j el subíndice correspondiente a la subclave candidata k'_{r+1}).

Después de probar todas las subclaves candidatas en todas las parejas, la subclave parcial objetivo con el contador T_i más grande será considerada la subclave correcta. La justificación de esto se basa en que dicha subclave es la que provoca con mayor frecuencia que la diferencia de entrada en la última ronda sea la indicada por la característica Δ.(recordemos que Δ tiene una probabilidad p_D alta de ocurrencia). Dicho de otro modo, la subclave k'_{r+1} con el contador T_i más grande es la que conlleva que ocurra con más frecuencia pares correctos.

Por otro lado, si tenemos un subclave incorrecta, se asume que la diferencia de entrada en la última ronda de S-boxes son bits aleatorios. Esto conlleva que la diferencia tiene una probabilidad muy baja de coincidir con la de la característica.

En la práctica, no es necesario la desencriptación parcial de todas las parejas de textos cifrados (C', C''). Como sabemos cuáles son las S-boxes activas de la última ronda, la diferencia en el resto debe ser 0. Si tenemos una pareja de texto cifrado que afecta a una S-box distinta de las activas, la descartamos [4]. A este proceso le llamamos **filtrado**.

Después de probar las N parejas, calculamos la probabilidad de que ocurran pares correctos para cada subclave candidata k'_{r+1}:

$$prob = \frac{T_i}{N}$$

donde T_i es el contador de cada subclave candidata k'_{r+1}.

Con ello y, en un caso ideal o en el mejor de ellos, podemos comprobar que la probabilidad p_D calculada de la característica diferencial Δ es aproximadamente igual que la probabilidad *prob* obtenida experimentalmente de la subclave que tiene el contador T_i más alto. En la siguiente sección veremos que pueden aparecer contadores (y, por ello, probabilidades) altas para subclaves parciales objetivo incorrectas.

Ejemplo 4.23. Explicaremos este método con, de nuevo, el ejemplo extraído de [16] donde, recordamos, ataca una red SPN de 4 rondas. Heys utiliza la característica diferencial descrita en el Ejemplo 4.20.

$$\Delta = (0B00, 0040, 0220, 0606)_{16}$$

Según esta característica, como la diferencia de entrada a las S-boxes de la última ronda (la cuarta) es $\Delta_3 = (0606)_{16}$, las S-boxes activas son $\mathcal{S}_{4,2}$ y $\mathcal{S}_{4,4}$. Por tanto, las subclave parcial objetivo es $k'_5 = [0000\ k_{5,5} \dots k_{5,8}\ 0000\ k_{5,13} \dots k_{5,16}]_2$. Por tanto, para N parejas hay que probar $[0\ 1\ 0\ 1]_{16}, [0\ 1\ 0\ 2]_{16}, \dots, [0\ E\ 0\ F]_{16}, [0\ F\ 0\ F]_{16}$, que son $16^2 = 256$ subclaves.

El contador T_i de cada subclave candidata k'_5 aumenta cuando, en la desencriptación parcial, obtengamos la misma diferencia que $\Delta_3 = [0\ 6\ 0\ 6]_{16}$. La tabla que recoge a cada

subclave y cada contador tiene un tamaño de 256 entradas por ser 8 los bits que vamos a buscar.

Heys prueba $N = 5000$ parejas de texto plano (10000 encriptaciones) cuya diferencia es $\Delta P = [0000\ 1011\ 0000\ 0000]_2 = [0\ B\ 0\ 0]_{16}$. Siguiendo el proceso descrito anteriormente, Heys halla que los bits de la subclave que se da por correcta k_5 son $[0000\ k_{5,5} \dots k_{5,8}\ 0000\ k_{5,13} \dots k_{5,16}]_2 = [0\ 2\ 0\ 4]_{16}$ porque su contador es $T_{20} = 122$. Por tanto, la probabilidad de que se den pares correctos con esta subclave es de $prob = 0{,}0244 \simeq p_D = \frac{27}{1024} = 0{,}0264$.

4.2. Aspectos del ataque

La probabilidad de la característica diferencial depende de lo alta que sea la probabilidad de los diferenciales de las S-boxes activas así como la cantidad de dichas S-boxes. Cuantas menos S-boxes activas haya, mayor será la probabilidad de la característica.

Subclaves erróneas

En la práctica, este proceso nos puede indicar también subclaves erróneas porque podrían aparecer otras subclaves candidatas con probabilidades *prob* altas y no ser las correctas. Los factores principales que contribuyen a esto son:

1. Las propiedades particulares de la S-box que influyen en la desencriptación parcial.
2. Asumir independencia entre las rondas del cifrado para determinar la probabilidad de la característica conlleva cierta imprecisión. Generalmente, para la mayoría de cifrados, no es problemático pero siempre hay un pequeño margen de error al asumirlo.
3. Pueden existir distintas características diferenciales $\Delta \neq \Delta'$ que dan lugar al mismo diferencial $(\Delta X, \Delta Y)$. Este concepto es similar al del casco lineal y se profundizará más en las próximas secciones.

Acerca de las características diferenciales

La probabilidad de la característica diferencial depende de lo alta que sea la probabilidad de los diferenciales de las S-boxes activas así como la cantidad de dichas S-boxes. Cuantas menos S-boxes activas haya, mayor será la probabilidad de la característica [16].

Observación 4.24. Existe un tipo de característica especial llamada **característica diferencial iterativa**. Si tenemos una característica $\Delta = (\Delta X_0, \dots, \Delta X_n)$ con $n \in \mathbb{N}$ y donde $\Delta X_0 = \Delta X_n$, podemos concatenarla consigo misma para obtener una característica Δ' que cubra un mayor número de rondas. En particular, si Δ tiene una probabilidad p_D, entonces Δ' tiene una probabilidad $p'_D = (p_D)^i$ siendo i el número de veces que concatenamos Δ.

En la Figura 4.3 podemos ver una característica diferencial iterativa del PRESENT que ha construido Wang [33]. Tiene una probabilidad $p_D = 2^{-4} \cdot 2^{-4} \cdot 2^{-4} \cdot 2^{-6} = 2^{-18}$. En su artículo, a partir de ella, construye otras características diferenciales que abarcan más rondas.

Rounds		Differences	Pr
I		$x_0 = 4, x_3 = 4$	1
R1	S	$x_0 = 5, x_3 = 5$	$\frac{1}{2^4}$
R1	P	$x_0 = 9, x_8 = 9$	1
R2	S	$x_0 = 4, x_8 = 4$	$\frac{1}{2^4}$
R2	P	$x_8 = 1, x_{10} = 1$	1
R3	S	$x_8 = 9, x_{10} = 9$	$\frac{1}{2^4}$
R3	P	$x_2 = 5, x_{14} = 5$	1
R4	S	$x_2 = 1, x_{14} = 1$	$\frac{1}{2^6}$
R4	P	$x_0 = 4, x_3 = 4$	1

Cuadro 4.3: Característica diferencial iterativa del cifrado PRESENT desarrollada por Meiqin Wang [33]

Relación señal-ruido

Sabemos que cada par de textos planos sugiere varias subclaves, es decir, cuando se prueban todas las subclaves posibles con un par de textos, los contadores de varias subclaves aumentan. Un par correcto sugiere sólo una subclave correcta y algunas erróneas. Un par incorrecto sugiere sólo subclaves erróneas. Para medir cuántas veces es más frecuente que se sugiera la subclave correcta en relación a el resto de las subclaves se utiliza la **relación señal-ruido**.

La señal se refiere a todas las veces que obtengo el diferencial intermedio predicho por la característica habiendo introducido la subclave correcta. El ruido se refiere a todas las veces que obtengo también dicho diferencial intermedio pero habiendo introducido una subclave errónea.

Definición 4.25. *Se define la* ***relación señal-ruido*** *(Signal-to-noise ratio, SNR) como el cociente entre el número esperado de pares correctos y la media de las subclaves incorrectas sugeridas.*

La media de las subclaves incorrectas sugeridas es el número medio de textos planos que usa el algoritmo para deducir incorrectamente los bits de la subclave parcial objetivo.

La relación viene dada por la siguiente fórmula [24]:

$$SNR = \frac{N \cdot p_{correcto}}{N \cdot p_{incorrecto}} = \frac{p}{\alpha \cdot \beta \cdot 2^{-k}}$$

donde

- N es el número de parejas de texto plano utilizados
- p es la probabilidad de la característica diferencial
- α es la media de las subclaves posibles sugeridas para cada pareja
- β es la proporción de parejas que filtramos frente a todas las parejas que tenemos
- k es el número de bits que tenemos que averiguar de la última subclave

Se deduce de la fórmula que la relación no depende de la cantidad de textos utilizada. La relación nos orienta sobre el número de parejas que necesitamos:

- Si esta relación es alta ($SNR > 2$), significa que se obtiene la diferencia predicha por la subclave correcta de forma más frecuente que por subclaves erróneas. Por tanto, se necesitarán pocos textos.
- Si la relación es baja ($1 < SNR \leq 2$), se tiene que tomar más pares para que la subclave correcta se distinga de entre las incorrectas.
- Si la relación es muy baja ($SNR \leq 1$), significa que aunque se tomen muchas parejas, el contador de la subclave correcta será indistinguible de las incorrectas.

Ejemplo 4.26. En el caso del DES, Biham y Shamir probaron experimentalmente que si SNR estaba entre 1 y 2, entonces sólo hacían falta entre 20 y 40 pares. Además, si SNR era alta, realmente sólo se necesitaban 3 ó 4 [4].

Datos necesarios

Como hemos mencionado en el apartado anterior, la relación señal-ruido nos ayuda a estimar cuántas parejas de texto plano debo tener para atacar el cifrado mediante criptoanálisis diferencial de forma exitosa.

Si p es la probabilidad de la característica diferencial, es decir, la probabilidad de que ocurra un par correcto, entonces necesitaremos $\frac{1}{p}$ el valor esperado de parejas de texto plano para que al menos el diferencial suceda una vez. Por tanto, es lógico que para que aparezca más de una vez pares correctos y así poder sugerir la clave correcta, utilicemos un múltiplo de $\frac{1}{p}$. En general, podremos aproximar el número de datos que necesitamos para este ataque mediante la siguiente fórmula:

$$N \simeq \frac{c}{p}$$

donde c es una constante que depende del cifrado [4, 16].

Este valor se puede ver como una cota inferior en términos de datos pero también de tiempo de procesamiento, ya que tendremos el doble de encriptaciones, $2N$. Esto se debe a que estamos encriptando parejas de texto plano que se obtienen como se describe en la sección anterior.

Acerca de las características diferenciales

Un diferencial $(\Delta X, \Delta Y)$ de una red SPN de r rondas donde ΔY corresponde a la diferencia de salida de una ronda s ($s < r$) puede tener distintas probabilidades p asociadas. La razón es que puede ser compuesto por diferentes características diferenciales. Dicho de otro modo, pueden existir características diferenciales diferentes (y por ello con probabilidades p_D distintas) pero que dan lugar al mismo diferencial $(\Delta X, \Delta Y)$.

El conjunto de todas las características diferenciales que dan lugar a un mismo diferencial no tiene una denominación concreta en la literatura aunque este fenómeno está registrado en algunos artículos como análogo al casco lineal. En este trabajo lo denominaremos **casco diferencial**.

Para asegurarnos de que un cifrado resiste a este criptoanálisis, la probabilidad de cualquier diferencial que podamos obtener debe estar por debajo de un umbral, así como la probabilidad de las características diferenciales.

En un casco diferencial, la característica diferencial que domina es aquella que tiene la mayor probabilidad entre todas las que dan lugar al mismo diferencial. Dicha probabilidad es la que se utiliza como probabilidad del diferencial. En la práctica, es muy díficil hallar con exactitud este casco.

Resistencia de los cifrados

La investigación sobre cómo deben ser los cifrados para resistir frente al criptoanálisis diferencial se ha centrado en las propiedades diferenciales de las S-boxes y la fase de permutación que ayuda a maximizar el número de S-boxes activas. El mejor ejemplo de ello es el AES.

4.2.1. Probabilidad de éxito

En esta subsección indicaremos brevemente cómo calcular la probabilidad de éxito de este ataque y los recursos en términos de datos para ello.

En el criptoanálisis diferencial también clasificamos las subclaves en una lista de mayor a menor con los contadores T_i de cada subclave. Volvemos a recordar que si queremos recuperar una subclave de m bits y la subclave correcta k_0 está en el puesto r entre las 2^m posibles, entonces el ataque se reduce en un factor $2^{m-log(r)}$ en términos de recursos sobre el ataque por fuerza bruta [29]. Al número de bits $m-log(r)$ lo llamaremos **ventaja dada por el ataque**.

Al igual que con el criptoanálisis lineal, algunos autores han dado fórmulas para la probabilidad del éxito del ataque [29, 5]. Desarrollaremos el trabajo de Selçuk, que demostró una fórmula que aproxima la probabilidad de éxito P_S teniendo como datos la cantidad de textos planos N y la relación señal-ruido SNR (véase la Definición 4.25). A continuación se enuncia y demuestra dicha fórmula que puede encontrarse corroborada empíricamente en [29]. Los resultados utilizados en la siguiente demostración están recogidos en el Apéndice E.

Proposición 4.27. *[29] Sea un ataque por criptoanálisis diferencial a un cifrado por bloques para recuperar m bits de la subclave. Sean N parejas de texto plano y cifrado, una característica diferencial con probabilidad p_D de este cifrado y una relación señal-ruido SNR. Supongamos que los contadores T_i de las subclaves son independientes y que están distribuidos idénticamente para todas las subclaves erróneas. Si P_S es la probabilidad de éxito de este ataque con una ventaja de a-bits sobre el ataque por fuerza bruta, entonces, para m y N suficientemente grandes:*

$$P_S = \Phi\left(\frac{\sqrt{Np_D\,SNR} - \Phi^{-1}(1-2^{-a})}{\sqrt{SNR+1}}\right)$$

donde Φ es la función de distribución acumulada de la distribución normal estándar.

Demostración. Primero, vamos a fijar y recordar notación para la demostración:

- m es el número de bits de las subclaves (o subclave) que queremos recuperar.

- N es el número de pares totales analizados
- k_0 es la subclave correcta
- k_i $1 \leq i \leq 2^m - 1$ son las claves incorrectas
- p_i $1 \leq i \leq 2^m - 1$ es la probabilidad de que las k_i sean sugeridas por un par de textos planos
- T_i $1 \leq i \leq 2^m - 1$ es el contador del número de veces que k_i es sugerida
- W_i $1 \leq i \leq 2^m - 1$ son los contadores T_i pero colocados en orden creciente
- p_D es la probabilidad de la característica diferencial Δ utilizada
- p_g es la probabilidad promedio de que alguna subclave dada sea sugerida por un par aleatorio con la diferencia inicial dada por la característica Δ
- $SNR = \frac{p_D}{p_g}$ es la relación señal-ruido que, por la notación utilizada, se puede escribir como dicho cociente.

Por hipótesis, los T_i son independientes y se distribuyen idénticamente $\forall\ i \neq 0$. Esto último significa que todas las claves erróneas tienen la misma probabilidad de ser sugeridas por un par aleatorio, p_i es igual $\forall\ i \neq 0$. Denotaremos por p_l a esta probabilidad.

Los contadores T_i siguen una distribución binomial.

$$\begin{aligned} T_0 &\sim \mathcal{B}(N, p_0) \\ T_i &\sim \mathcal{B}(N, p_l) \quad \forall i \neq 0 \end{aligned}$$

Denotaremos por F_0 y F_l a las funciones de distribución y f_0 y f_l a las funciones de densidad, respectivamente. Como N es grande, las distribuciones binomiales se pueden aproximar a distribuciones normales, $\mathcal{N}(\mu_0, \sigma_0^2)$ y $\mathcal{N}(\mu_l, \sigma_l^2)$ respectivamente, donde los parámetros son

$$\begin{aligned} \mu_0 = Np_0 \quad \sigma_0^2 = Np_0(1-p_0) \approx Np_0 \\ \mu_l = Np_l \quad \sigma_l^2 = Np_l(1-p_l) \approx Np_l \end{aligned}$$

Como hemos explicado anteriormente, en un ataque diferencial, la subclave correcta es sugerida tanto por pares correctos como por pares incorrectos, mientras que las subclaves incorrectas son sugeridas siempre por pares incorrectos. En ese caso,

$$\begin{aligned} p_0 = p_D + (1-p_D)p_g \approx p_D + p_g \\ p_l = p_g \end{aligned}$$

En un ataque con una ventaja de a-bits, el éxito se define como tener la subclave correcta k_0 clasificada en las primeras 2^{m-a} subclaves de entre las 2^m subclaves candidatas. En términos de contadores, el contador de la subclave correcta T_0 es mayor que los contadores (ordenados) W_i de las subclaves que están fuera de las primeras 2^{m-a}:

$$T_0 > W_{2^m - 2^{m-a}}$$

Por comodidad, denotaremos por $\bar{r} = 2^m - 2^{m-a} = 2^m(1 - 2^{-a})$.

El Teorema E.3 afirma que un cuantil muestral de orden q de una muestra de n elementos sigue una distribución normal con una expresión explícita para la media y la

desviación típica de dicho cuantil. En este caso, consideramos que los contadores W_i son un cuantil muestral de orden $q = 1 - 2^{-a}$. Por tanto, por el Teorema E.3, $W_{\bar{r}}$ sigue, aproximadamente, una distribución normal $\mathcal{N}(\mu_q, \sigma_q^2)$ siendo F_q y f_q las funciones de distribución y de densidad de esta distribución, respectivamente y donde

$$\mu_q = F_l^{-1}(q) = F_l^{-1}(1-2^{-a}) \overset{*}{=} \mu_l + \sigma_l \Phi^{-1}(1-2^{-a})$$
$$\sigma_q = \frac{1}{f_l(\mu_q)}\sqrt{\frac{q(1-q)}{n}} = \frac{1}{f_l(\mu_{1-2^{-a}})} 2^{-\frac{m+a}{2}} \overset{*}{=} \frac{\sigma_l}{\phi(\Phi^{-1}(1-2^{-a}))} 2^{-\frac{m+a}{2}}$$

donde la última igualdad en cada línea se verifica por ser F_l y f_l a las funciones de distribución y de densidad normales.

Luego, la probabilidad de éxito P_S es

$$P_S = \int_0^\infty \int_{-\infty}^{x} f_q(y)dy f_0(x)dx$$

Según Selçuk, para $a, m \geq 8$, tenemos que $\mu_q > 5\sigma_q$ y, por tanto, la probabilidad de $T_0 - W_{\bar{r}} < 0$ es despreciable [29]. Como T_0 y $W_{\bar{r}}$ siguen una distribución normal, entonces $T_0 - W_{\bar{r}}$ sigue también una distribución normal $\mathcal{N}(\mu_j, \sigma_j^2)$ donde $\mu_j = \mu_0 - \mu_q$ y $\sigma_j = \sqrt{\sigma_0^2 + \sigma_q^2}$

Por tanto,

$$P_S = P(T_0 - W_{\bar{r}} > 0) = \int_0^\infty f_j(x)dx = \int_{-\frac{\mu_0-\mu_q}{\sqrt{\sigma_0^2+\sigma_q^2}}}^{\infty} \phi_{\mu_j,\sigma_j}(x)dx$$

donde ϕ_{μ_j,σ_j} es la función de densidad de la distribución normal $\mathcal{N}(\mu_j, \sigma_j^2)$. Se puede consultar en el Apéndice D.

Para $\sigma_q^2 \ll \sigma_0^2$, tenemos

$$P_S = \int_{-\frac{\mu_0-\mu_q}{\sigma_0}}^{\infty} \phi(x)dx \tag{4.1}$$

Si desarrollamos el límite inferior de la integral, podremos ponerla en función de la relación señal-ruido y obtendremos la fórmula del enunciado del teorema.

$$\begin{aligned}
-\frac{\mu_0 - \mu_q}{\sigma_0} &= \frac{-Np_0 + Np_l + \sqrt{Np_l}\Phi^{-1}(1-2^{-a})}{\sqrt{Np_0}} \\
&= \frac{-Np_D + \sqrt{Np_g}\ \Phi^{-1}(1-2^{-a})}{\sqrt{N(p_D+p_g)}} \\
&= -\sqrt{Np_D}\sqrt{\frac{p_D}{p_D+p_g}} + \sqrt{\frac{p_g}{p_D+p_g}}\Phi^{-1}(1-2^{-a}) \\
&= -\sqrt{Np_D}\sqrt{\frac{SNR}{SNR+1}} + \sqrt{\frac{1}{SNR+1}}\ \Phi^{-1}(1-2^{-a}) \\
&= \sqrt{\frac{-Np_D\ SNR}{SNR+1}} + \frac{\Phi^{-1}(1-2^{-a})}{\sqrt{SNR+1}} \\
&= \frac{-\sqrt{Np_D\ SNR} + \Phi^{-1}(1-2^{-a})}{\sqrt{SNR+1}}
\end{aligned}$$

Sustituyendo en la anterior ecuación (4.1):

$$P_S = \Phi\left(\frac{\sqrt{Np_D\ SNR} - \Phi^{-1}(1-2^{-a})}{\sqrt{SNR+1}}\right)$$

□

Como corolario de esta proposición, Selçuk también aproximó el número de textos planos y cifrados necesarios para conseguir una ventaja de a-bits dada una probabilidad de éxito P_S, despejando N de la fórmula anterior.

Corolario 4.28. *[29] Bajo las mismas hipótesis que la Proposición 4.27, el número de textos planos necesarios en un ataque diferencial para obtener una ventaja de a-bits con una probabilidad de éxito P_S es de*

$$N = \frac{(\sqrt{SNR+1}\cdot\phi^{-1}(P_S) + \phi^{-1}(1-2^{-a}))^2}{SNR}p^{-1}$$

4.3. PRESENT

Al igual que con el criptoanálisis lineal, los autores del cifrado PRESENT realizaron un análisis de seguridad del mismo frente al criptoanálisis diferencial, entre otros [6]. En este caso, demostraron una cota inferior del número de S-boxes activas presentes en una característica diferencial.

Proposición 4.29. *[6] Cualquier característica diferencial de* 5 *rondas del cifrado PRESENT tiene como mínimo* 10 *S-boxes activas.*

Demostración. Primero, agrupamos las 16 S-boxes $\mathcal{S}$ de la fase de sustitución σ del cifrado PRESENT en grupos de 4 S-boxes.

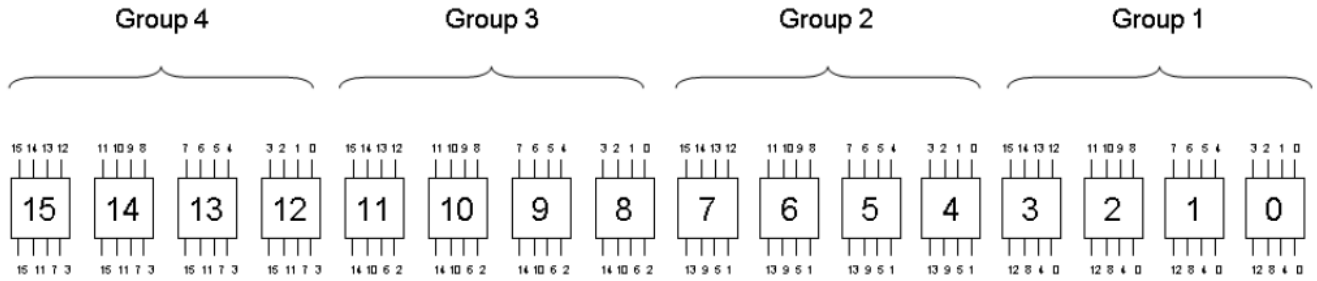

Figura 4.1: Agrupación de las 16 S-boxes del PRESENT. Los números de entrada a ellas indican de qué S-box provienen los bits y los números de salida, a qué S-box se dirigen [6]

Si estudiamos la fase de permutación π, podemos deducir que

Observación 1 Los bits de entrada a una S-box proceden de 4 S-boxes distintas del mismo grupo.

Ejemplo 4.30. Si escogemos la S-box $\mathcal{S}_{15}$, vamos a ver que sus bits de entrada provienen de las 4 S-boxes de su grupo, el grupo 4. Las posiciones de los bits de entrada a esta S-box son $(63\ 62\ 61\ 60)_2$. Si estudiamos de dónde vienen estos bits según la permutación π (véase el Cuadro B.2), vemos que:

$$\begin{array}{ll} \pi(j) = 63 & \rightarrow j = 63 \quad \text{posición que está en la S-box } \mathcal{S}_{15} \\ \pi(j) = 62 & \rightarrow j = 59 \quad \text{posición que está en la S-box } \mathcal{S}_{14} \\ \pi(j) = 61 & \rightarrow j = 55 \quad \text{posición que está en la S-box } \mathcal{S}_{13} \\ \pi(j) = 60 & \rightarrow j = 51 \quad \text{posición que está en la S-box } \mathcal{S}_{12} \end{array}$$

Como podemos ver, $\mathcal{S}_{15}, \mathcal{S}_{14}, \mathcal{S}_{13}, \mathcal{S}_{12}$ son las 4 S-boxes distintas que pertenecen al grupo 4. Análogamente para el resto de S-boxes.

Observación 2 Los bits de entrada a un grupo de S-boxes provienen de las 16 S-boxes.

Ejemplo 4.31. Esta observación se puede consultar en la Imagen 4.1. Si escogemos el grupo 4, la entrada a sus S-boxes $\mathcal{S}_{15}, \mathcal{S}_{14}, \mathcal{S}_{13}, \mathcal{S}_{12}$ es:

$$(63\ 62\ 61\ 60\ 59\ 58\ 57\ 56\ 55\ 54\ 53\ 52\ 51\ 50\ 49\ 48)_2$$

Las posiciones de los bits de donde provienen son:

$\mathcal{S}$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
j	63	59	55	51	47	43	39	35	31	27	23	19	15	11	7	3
$\pi(j)$	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Como se puede comprobar, provienen de las 16 S-boxes. Análogamente para el resto de grupos.

Observación 3 Cada uno de los 4 bits de salida de una S-box se convierten en la entrada de 4 S-boxes distintas, cada una de ellas perteneciendo a un grupo distinto.

Ejemplo 4.32. Esta observación se puede consultar en la Imagen 4.1. Escogiendo la S-box $\mathcal{S}_{15}$, sus bits de salida son $(63\ 62\ 61\ 60)_2$.

$$\begin{array}{ll} j = 63 & \rightarrow \pi(j) = 63 \quad \text{posición que está en la S-box } \mathcal{S}_{15} \quad \text{en el grupo 4} \\ j = 62 & \rightarrow \pi(j) = 47 \quad \text{posición que está en la S-box } \mathcal{S}_{11} \quad \text{en el grupo 3} \\ j = 61 & \rightarrow \pi(j) = 31 \quad \text{posición que está en la S-box } \mathcal{S}_{7} \quad \text{en el grupo 2} \\ j = 60 & \rightarrow \pi(j) = 15 \quad \text{posición que está en la S-box } \mathcal{S}_{3} \quad \text{en el grupo 1} \end{array}$$

Por tanto, cada uno de los bits de salida se ha convertido en la entrada de S-boxes activas pertenecientes a grupos distintos.

Observación 4 Los bits de salida de las S-boxes que pertenecen a grupos distintos son bits de entrada de S-boxes distintas.

Ejemplo 4.33. Esta observación se puede consultar en la Imagen 4.1. Si lo vemos con los bits de salida del grupo 4 y 3, las S-boxes del grupo 4 son $\mathcal{S}_{15}, \mathcal{S}_{11}, \mathcal{S}_7, \mathcal{S}_3$ y las del grupo 3 son $\mathcal{S}_{14}, \mathcal{S}_{10}, \mathcal{S}_6, \mathcal{S}_2$. Todas las S-boxes son distintas.

Recordemos que el PRESENT tiene 31 rondas, que vamos a enumerar del 1 al 31. Consideremos 5 rondas consecutivas que van de $i-2$ a $i+2$ para $i \in \{3, \ldots, 29\}$ y sea D_j el número de S-boxes activas en la ronda j.

Si tenemos más de 2 S-boxes activas por ronda, $D_j \geq 2$ con $i-2 \leq j \leq i+2$, entonces el teorema se obtiene trivialmente. Por tanto, supongamos que el número de S-box activas de una ronda j es 1, es decir, $D_j = 1$. Podemos distinguir varios casos:

Caso $D_i = 1$ La S-box del PRESENT es tal que la diferencia de entrada de un sólo bit produce una diferencia de salida de dos bits (criterio de diseño [6]). Por ello, $D_{i-1} + D_{i+1} \geq 3$ porque el número de S-box activas de la ronda anterior $i-1$ debe ser al menos 1 y en la ronda siguiente $i+1$ de 2 por el criterio de diseño o viceversa.

Por la Observación 1, todas las S-boxes activas de la ronda $i-1$ (la anterior) son del mismo grupo y cada una de ellas tiene un sólo bit de diferencia en la salida. Por la Observación 2, tenemos que $D_{i-2} \geq 2D_{i-1}$ ya que los bits de entrada de las S-boxes activas de la ronda $i-1$ son del mismo grupo y, por tanto, provienen de las 16 S-boxes. Consecuentemente, por la Observación 3, todas las S-boxes activas de la ronda $i+1$ pertenecen a distintos grupos y tienen un sólo bit en la diferencia de entrada. Por la Observación 4, tenemos que $D_{i+2} \geq 2D_{i+1}$. Recopilando las expresiones:

$$\begin{array}{c} D_i = 1 \\ D_{i-1} + D_{i+1} \geq 3 \\ D_{i-2} \geq 2D_{i-1} \\ D_{i+2} \geq 2D_{i+1} \end{array} \quad \text{por tanto} \quad D_{i-2} + D_{i+2} \geq 2D_{i-1} + 2D_{i+1} = 2(D_{i-1} + D_{i+1}) \geq 2 \cdot 3$$

Sumando todas las S-boxes activas:

$$\sum_{j=i-2}^{i+2} D_j \geq 1 + 3 + 2 \cdot 3 = 10$$

Caso $D_{i-1} = 1$ Si $D_i = 1$, estamos en el caso anterior. Por tanto, supongamos que $D_i \geq 2$. Como $D_{i-1} = 1$, en la ronda $i-2$ habría al menos una S-box activa, esto es, $D_{i-2} \geq 1$. Por la Observación 3, todas las S-boxes activas de la ronda i pertenecen a distintos grupos y sólo tienen un bit en la diferencia de entrada. Como consecuencia y por la Observación 4, $D_{i+1} \geq 2D_i \geq 4$ ya que, además, un bit de diferencia en la entrada causa dos. Además, todas las S-boxes activas de la ronda $i+1$ tienen un sólo bit en la diferencia de entrada y están distribuidas de tal forma que al menos dos grupos de S-boxes tienen al menos una S-box activa. Esto significa que $D_{i+2} \geq 4$ y, por tanto, que

$$\sum_{j=i-2}^{i+2} D_j \geq 1 + 1 + 2 + 4 + 4 = 12$$

Caso $D_{i+1} = 1$ Si $D_i = 1$, estamos en el primer caso de nuevo. Por tanto, supongamos que $D_i \geq 2$. Por la Observación 1, todas las S-boxes activas de la ronda i son del mismo grupo y cada una de ellas sólo tiene un bit en la diferencia de salida. Por ello y por la Observación 2, $D_{i-1} \geq 2D_i \geq 4$. Además, todas las S-boxes activas de

la ronda $i-1$ tienen sólo un bit en la diferencia de salida y están distribuidas de tal forma que al menos dos grupos contienen al menos 2 S-boxes activas. Por tanto, tenemos que $D_{i-2} \geq 4$. De la ronda $i-2$ podemos suponer que al menos habrá una S-box activa por haber una en la ronda anterior. Por tanto, que

$$\sum_{j=i-2}^{i+2} D_j \geq 4+4+2+1+1 = 12$$

Caso $D_{i+2} = 1$ o $D_{i-2} = 1$ Se procede de manera semejante al segundo y tercer caso.

□

Como se mencionó en la anterior sección, el número de S-boxes activas afecta a la probabilidad de la característica. Cuanto mayor número de ellas, peor para el atacante ya que la probabilidad disminuye considerablemente. Por ello, los autores del PRESENT consideraron una buena cota la enunciada por el teorema.

Sin embargo, posteriores criptoanálisis diferenciales de diversos autores han mostrado características diferenciales que, a pesar de tener más de 10 S-boxes activas, ponen en cierto riesgo al cifrado [33, 1]. No lo suficiente para considerar el cifrado roto pero sí se consideran hallazgos relevantes en materia de seguridad.

El criptoanálisis diferencial más relevante hasta la fecha es el realizado por Meiqin Wang, donde construye 24 características diferenciales de 14 rondas cada una, entre otras [33]. La construcción se basa en características diferenciales iterativas y, en particular, describe la más satisfactoria, de 4 rondas (véase la Figura 4.3). En la Figura 4.4 Wang recopila las mejores características que ha hallado. Podemos observar que cuando llegó a la correspondiente a la ronda 15 paró porque era menor que 2^{-64}, siendo 64 la longitud del bloque del cifrado PRESENT. Se recuerda que esto es menos eficiente que el ataque por fuerza bruta (véase la Sección 1.5).

En particular, Wang ataca 16 rondas del PRESENT con la mejor característica diferencial de 14 rondas que ha construido con una probabilidad $p_D = 2^{-62}$ para recuperar 32 bits de las subclaves $k_1 6$ (8 bits) y $k_1 7$ (32 bits).

$$\Delta = (0700000000000700, \ldots, 0000000900000009)_{16}$$

Utiliza 2^{64} textos planos elegidos, 2^{65} accesos a la memoria (tiempo de procesamiento) y 2^{32} contadores de 6-bits[10] (memoria utilizada).

La relación señal-ruido, calculada por Wang, de su ataque es $SNR = 17{,}63$ [33]. Tal y como vimos en la Sección 4.2, al ser una relación alta se necesitarán *pocos* textos planos. Entiéndase como *pocos* textos a una cantidad mucho menor que en otros ataques o si tuviéramos este mismo ataque con peor característica diferencial.

La probabilidad de éxito, calculada por Wang, mediante la fórmula descrita en la Sección 4.2.1, de su ataque es [33]

$$P_S = \int_{-\frac{\sqrt{\mu SNR} - \phi^{-1}(1-2^{-a})}{\sqrt{SNR+1}}}^{\infty} \phi(x)dx = 0{,}999999939$$

[10]Estos contadores son dispositivos electrónicos que se emplean en el ámbito de la electrónica para contar y almacenar valores en un rango de 0 a 63 utilizando 6 bits de información

Table 6. Probability of the Best Characteristics

Rounds	Differential Probability	Number of Active S-box
5	2^{-20}	10
6	2^{-24}	12
7	2^{-28}	14
8	2^{-32}	16
9	2^{-36}	18
10	2^{-42}	20
11	2^{-46}	22
12	2^{-52}	24
13	2^{-56}	26
14	2^{-62}	28
15	2^{-66}	30

Cuadro 4.4: Probabilidad y número de S-boxes activas de las mejores características diferenciales de Wang [33]

donde $a = 32$ los bits de la subclave parcial objetivo y μ el número de pares correctos que pueden ser obtenidos

$$\mu = p \cdot N = 2^{-62} \cdot 2^{63} \cdot 24 = 48$$

siendo N es el producto de los pares utilizados en cada característica diferencial, 2^{63} y el número de características diferenciales usadas en el ataque, 24.

Capítulo 5

Conclusiones

Desde su publicación, el criptoanálisis diferencial y el criptoanálisis lineal supusieron un gran cambio en el campo de la criptografía de clave secreta. No sólo obligaron a cambiar el cifrado estándar de la información digital, el DES, sino también contribuyeron a evaluar y diseñar los cifrados para aumentar su seguridad.

Algunos autores han relacionado ambos criptoanálisis de dos formas diferenciadas. Unos, estudiando la relación existente entre, por ejemplo, las tablas LAT y DDT. Otros combinando ambos criptoanálisis creando el criptoanálisis diferencial-lineal [20].

Existen muchos tipos de ataques derivados del criptoanálisis diferencial. Podemos citar algunos como el criptoanálisis diferencial múltiple, criptoanálisis diferencial truncado, criptoanálisis diferencial imposible o criptoanálisis diferencial improbable, entre otros [32, 31]. Todos ellos han supuesto una versión mejorada del original en diferentes aspectos y han comprometido, en algunos casos, a algunos cifrados. De hecho, si se profundizase más aún en el análisis del PRESENT, encontraríamos que es vulnerable al ataque diferencial truncado y al ataque de canal lateral [26]. La razón por la cual esto no supone un riesgo real en la práctica es porque no se justifica el esfuerzo que hay que dedicar en montar estos ataques para romperlo y acceder a la posible información que se esté encriptando.

Es digno de consideración matemática, la perspectiva del criptoanálisis diferencial desde el álgebra. En el desarrollo de este trabajo, se estudió este enfoque para ser añadido en el mismo. Desafortunadamente, por limitación de tiempo y para desarrollar de manera más completa los contenidos expuestos, no se ha podido incluir.

En criptografía, los llamados ataques algebraicos se basan en explotar la estructura algebraica de los cifrados para hallar la información [2]. Convierten el cifrado en un sistema de ecuaciones polinómicas en $\mathbb{F}_2^n[x_1, \ldots, x_m]$ y tratan de resolverlo mediante métodos que abarcan desde bases de Gröbner hasta modelarlos como problemas SAT. La gran desventaja de estos ataques es la elevada cantidad de polinomios de alto grado que suelen resultar del cifrado y que conllevan un alto coste en términos de recursos. Sin embargo, esto no frena la investigación de los mismos y se puede observar que se incluyen en los análisis de seguridad de los cifrados [6].

Algunos autores han querido combinar los ataques algebraicos con el criptoanálisis diferencial para dar lugar al criptoanálisis diferencial-algebraico [1, 3]. En el caso de Martin Albrecht [1], ha investigado en varios trabajos este enfoque: después de realizar una característica diferencial de R rondas del cifrado, transforma las últimas rondas del cifrado en un sistema polinómico de ecuaciones (usando, además, bases de Gröbner) para

acceder con ambas herramientas a bits de las subclaves. Existe cierta controversia con su trabajo ya que mientras él afirma que se consiguen con un rendimiento mejor y un uso menor de recursos respecto al ataque por fuerza bruta y el ataque tradicional diferencial otros autores ponen en tela de juicio ciertas suposiciones que da como ciertas [34]. En cualquier caso, el propio Albrecht admite que es un campo por explorar [1].

Como se ha podido comprobar, para abordar el criptoanálisis diferencial es necesario entender algunas nociones de la criptografía de clave secreta y, en particular, los cifrados por bloques. El lector de este trabajo se ha encontrado con una descripción de los principales tipos, así como, la explicación del criptoanálisis lineal. Este ataque contribuye a un mejor entendimiento del ataque diferencial. Por último, se ha detallado el tema central del documento, el criptoanálisis diferencial.

Apéndice A

PRESENT

Se puede consultar la representación gráfica de dos rondas del cifrado PRESENT, así como, la representación en forma de tabla de la fase de permutación π. Ésta última forma de reflejar la permutación es la más frecuente, además, de ser la que presentaron los autores del cifrado [6].

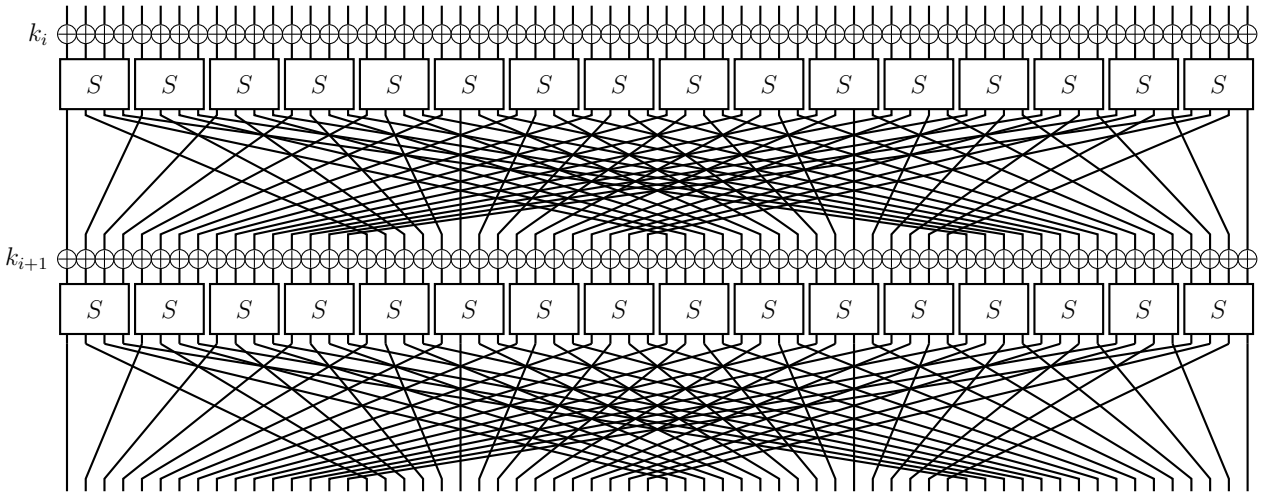

Figura A.1: Representación gráfica del cifrado PRESENT [17]

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi(j)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
j	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$\pi(j)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
j	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$\pi(j)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
j	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$\pi(j)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Cuadro A.1: Representación tabular de la fase de permutación del PRESENT siendo j la posición del bit y $\pi(j)$ la posición del bit permutado

Apéndice B

Red SPN de Heys

El artículo de Heys explica el funcionamiento de los criptoanálisis lineal y diferencial haciendo uso de una red SPN sencilla [16]. En este trabajo también ha sido utilizada para ejemplificar ambos ataques. A continuación se incluyen la S-box $\mathcal{S}(x)$, la fase de permutación π, la LAT y la DDT de dicha red.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\mathcal{S}(x)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Cuadro B.1: S-box de la red SPN de Heys [16]

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi(j)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Cuadro B.2: Permutación de la red SPN de Heys [16]

α/β	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	-2	-2	0	0	-2	6	2	2	0	0	2	2	0	0
2	0	0	-2	-2	0	0	-2	-2	0	0	2	2	0	0	-6	2
3	0	0	0	0	0	0	0	0	2	-6	-2	-2	2	2	-2	-2
4	0	2	0	-2	-2	-4	-2	0	0	-2	0	2	2	-4	2	0
5	0	-2	-2	0	-2	0	4	2	-2	0	-4	2	0	-2	-2	0
6	0	2	-2	4	2	0	0	2	0	-2	2	4	-2	0	0	-2
7	0	-2	0	2	2	-4	2	0	-2	0	2	0	4	2	0	2
8	0	0	0	0	0	0	0	0	-2	2	2	-2	2	-2	-2	-6
9	0	0	-2	-2	0	0	-2	-2	-4	0	-2	2	0	4	2	-2
A	0	4	-2	2	-4	0	2	-2	2	2	0	0	2	2	0	0
B	0	4	0	-4	4	0	4	0	0	0	0	0	0	0	0	0
C	0	-2	4	-2	-2	0	2	0	2	0	2	4	0	2	0	-2
D	0	2	2	0	-2	4	0	2	-4	-2	2	0	2	0	0	2
E	0	2	2	0	-2	-4	0	2	-2	0	0	-2	-4	2	-2	0
F	0	-2	-4	-2	-2	0	2	0	0	-2	4	-2	-2	0	2	0

Cuadro B.3: *LAT* de la red SPN de Heys [16]

$\Delta X/\Delta Y$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Cuadro B.4: *DDT* de la red SPN de Heys [16]

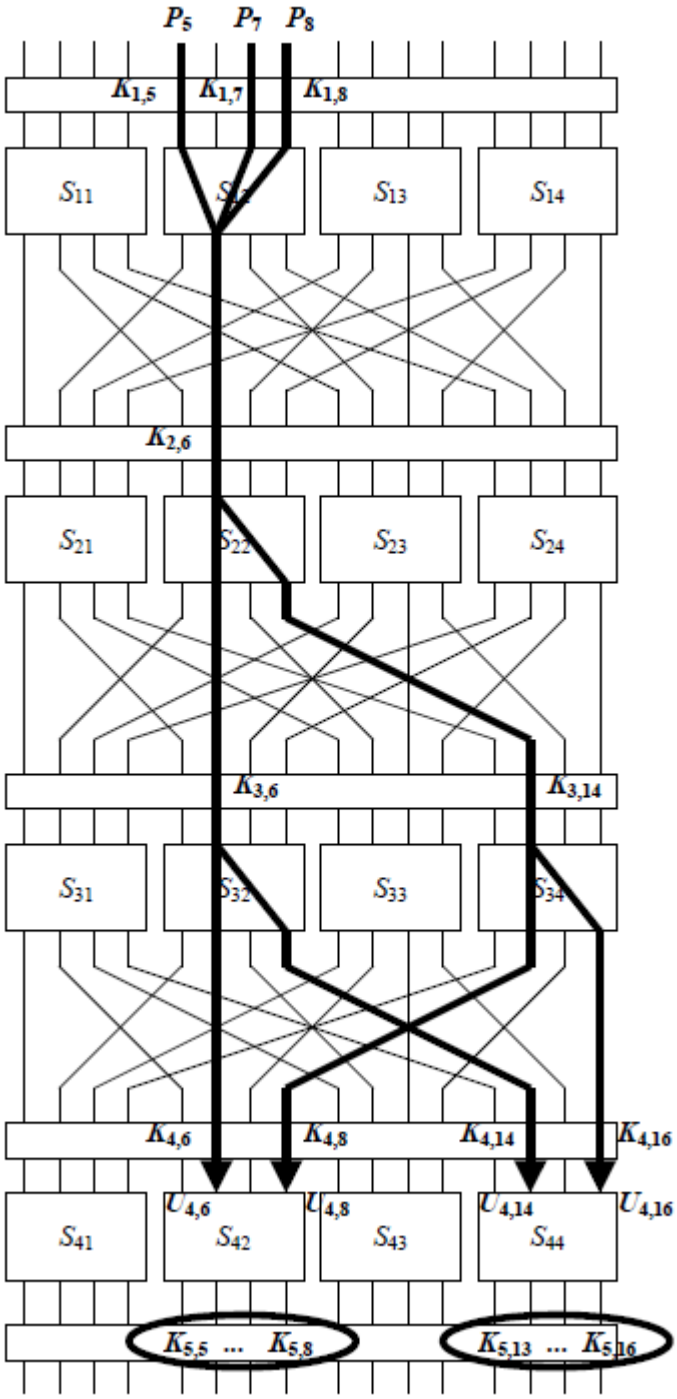

Figura B.1: Aproximación lineal de la red SPN de Heys [16]

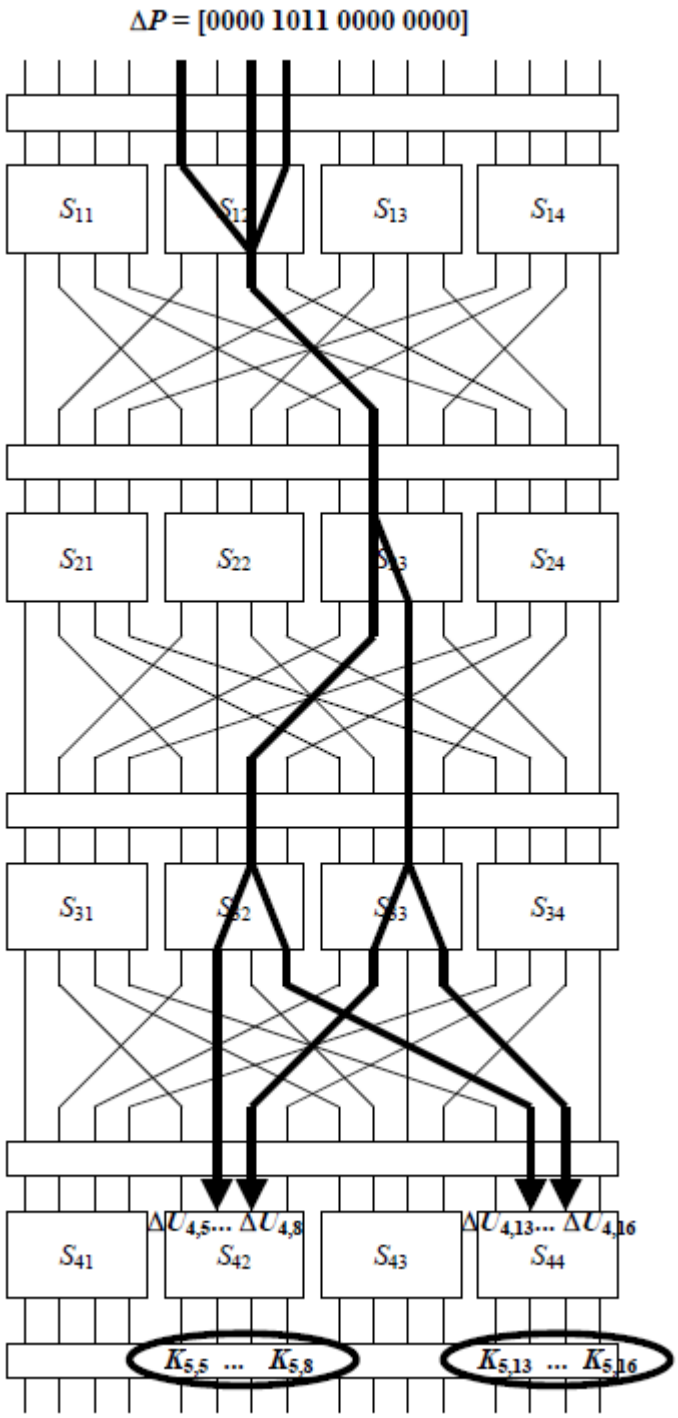

Figura B.2: Característica diferencial de la red SPN de Heys [16]

Apéndice C

Códigos correspondientes a los ejemplos

El análisis booleano incluido en la Sección 2.1.1 del cifrado PRESENT ha sido realizado, en parte, en el sistema algebraico computacional *SageMath*. Todos los códigos descritos han sido extraídos del sitio web de *SageMath* y modificados para este estudio. Han sido procesados en una interfaz web llamada *SageMathCell*. También han sido utilizados para hallar la LAT y DDT del cifrado PRESENT (véase las Figuras 3.2 y 4.2).

Utilizadas para el análisis booleano

A continuación se escriben los códigos utilizados para el análisis booleano de las funciones coordenadas de la S-box del PRESENT (véase la Tabla 2.2).

Cálculo del grado algebraico de las $\mathcal{S}_i(x)$

```
from sage.crypto.boolean_function import BooleanFunction
S1 = BooleanFunction([1,0,0,1,1,0,1,1,0,1,1,1,0,0,0,0])
S2 = BooleanFunction([1,1,1,0,0,0,0,1,0,1,1,0,1,1,0,0])
S3 = BooleanFunction([0,0,1,1,0,0,1,0,1,1,1,0,0,1,0,1])
S4 = BooleanFunction([0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0])
print(S1.algebraic_degree())
print(S2.algebraic_degree())
print(S3.algebraic_degree())
print(S4.algebraic_degree())
```

Cálculo de la no-linealidad de las $\mathcal{S}_i(x)$

```
from sage.crypto.boolean_function import BooleanFunction
S1 = BooleanFunction([1,0,0,1,1,0,1,1,0,1,1,1,0,0,0,0])
S2 = BooleanFunction([1,1,1,0,0,0,0,1,0,1,1,0,1,1,0,0])
S3 = BooleanFunction([0,0,1,1,0,0,1,0,1,1,1,0,0,1,0,1])
S4 = BooleanFunction([0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0])
print(S1.nonlinearity())
print(S2.nonlinearity())
```

```
print(S3.nonlinearity())
print(S4.nonlinearity())
```

Cálculo del balance de $\mathcal{S}_i(x)$

```
from sage.crypto.boolean_function import BooleanFunction
S1 = BooleanFunction([1,0,0,1,1,0,1,1,0,1,1,1,0,0,0,0])
S2 = BooleanFunction([1,1,1,0,0,0,0,1,0,1,1,0,1,1,0,0])
S3 = BooleanFunction([0,0,1,1,0,0,1,0,1,1,1,0,0,1,0,1])
S4 = BooleanFunction([0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0])
print(S1.is_balanced())
print(S2.is_balanced())
print(S3.is_balanced())
print(S4.is_balanced())
```

Cálculo de la función de autocorrelación de las $\mathcal{S}_i(x)$

```
from sage.crypto.boolean_function import BooleanFunction
S1 = BooleanFunction([1,0,0,1,1,0,1,1,0,1,1,1,0,0,0,0])
S2 = BooleanFunction([1,1,1,0,0,0,0,1,0,1,1,0,1,1,0,0])
S3 = BooleanFunction([0,0,1,1,0,0,1,0,1,1,1,0,0,1,0,1])
S4 = BooleanFunction([0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0])
print(S1.autocorrelation())
print(S2.autocorrelation())
print(S3.autocorrelation())
print(S4.autocorrelation())
```

Cálculo de la transformada de Walsh-Hadamard de las $\mathcal{S}_i(x)$

```
from sage.crypto.boolean_function import BooleanFunction
S1 = BooleanFunction([1,0,0,1,1,0,1,1,0,1,1,1,0,0,0,0])
S2 = BooleanFunction([1,1,1,0,0,0,0,1,0,1,1,0,1,1,0,0])
S3 = BooleanFunction([0,0,1,1,0,0,1,0,1,1,1,0,0,1,0,1])
S4 = BooleanFunction([0,1,0,1,1,0,0,1,1,0,1,0,0,1,1,0])
print(S1.walsh_hadamard_transform())
print(S2.walsh_hadamard_transform())
print(S3.walsh_hadamard_transform())
print(S4.walsh_hadamard_transform())
```

A continuación se escriben los códigos utilizados para el análisis booleano de la S-box del PRESENT (véase la Tabla 2.1).

Cálculo del máximo grado algebraico de todas las funciones componentes de una S-box

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.max_degree()
```

Cálculo de la no-linealidad de una S-box

```
from sage.crypto.sbox import SBox
```

```
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.nonlinearity()
```

Cálculo de si una función booleana es balanceada

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.is_balanced()
```

Utilizadas en los criptoanálisis

Cálculo de la LAT de una S-box

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
lat_abs_bias = S.linear_approximation_table()
```

Cálculo de la DDT de una S-box

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.difference_distribution_table()
```

Cálculo de la uniformidad diferencial

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.maximal_difference_probability_absolute()
```

Cálculo del número de ramas

```
from sage.crypto.sbox import SBox
S = SBox(12,5,6,11,9,0,10,13,3,14,15,8,4,7,1,2)
S.differential_branch_number()
```

Utilizadas en otras partes del documento

Cálculo de la autocorrelación de la función del Ejemplo 1.40

```
from sage.crypto.boolean_function import BooleanFunction
B = BooleanFunction([1,1,1,0,0,1,1,1])
print(B.autocorrelation())
```

Apéndice D

Distribución normal

Distribución normal

Una variable aleatoria X sigue una distribución normal de media μ y desviación típica σ, $\mathcal{N}(\mu, \sigma)$, si recorre toda la recta y la función de densidad es

$$\phi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{con } x \in \mathbb{R}$$

Su función de distribución es

$$\Phi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt \quad \text{con } x \in \mathbb{R}$$

Distribución normal estándar

Si tipificamos la variable X, $Z = \dfrac{X-\mu}{\sigma}$, la variable tipificada Z sigue una distribución normal estándar de media $\mu = 0$ y desviación típica $\sigma = 1$, $\mathcal{N}(0, 1)$.

Su función de densidad es

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad \text{con } x \in \mathbb{R}$$

Su función de distribución es

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\, dt \quad \text{con } x \in \mathbb{R}$$

En el documento, la notación para las variables que sigan una distribución normal $\mathcal{N}(\mu, \sigma)$ será f para la función de densidad y F para la función de distribución.

Apéndice E

Análisis de Selçuk de la probabilidad de éxito

Para el desarrollo de la demostración de la Proposición 4.27 es necesario las siguientes definiciones y teorema [29].

Definición E.1. *Sean $X_1, \ldots, X_n$ variables aleatorias idénticamente distribuidas e independientes. Denotemos por $X_1^*, \ldots, X_n^*$ a estas mismas variables pero ordenadas de forma creciente. Se llama* ***estadístico de orden i*** *a la variable X_i^* de la muestra $X_1, \ldots, X_n$*

Definición E.2. *Sea $0 < q < 1$. Se define* ***el cuantil muestral de orden q*** *al estadístico de orden $\lfloor qn \rfloor + 1$, $X^*_{\lfloor qn \rfloor + 1}$.*

Teorema E.3. *Sean $X_1, \ldots, X_n$ variables aleatorias idénticamente distribuidas e independientes con una función de distribución absolutamente continua $F(x)$. Supongamos que la función de densidad $f(x) = F'(x)$ es continua y positiva en el intervalo $[a, b)$. Si $0 < F(a) < q < F(b) < 1$ y $i(n)$ es una serie de integrales tales que*

$$\lim_{n\to\infty} P\left(\frac{X^*_{i(n)} - \mu_q}{\sigma_q} < x\right) = \Phi(x)$$

donde

$$\mu_q = F^{-1}(q)$$
$$\sigma_q = \tfrac{1}{f(\mu_q)}\sqrt{\tfrac{q(1-q)}{n}}$$

Tomando $i(n) = \lfloor qn \rfloor + 1$, el teorema dice que un cuantil muestral de orden q experimental de una muestra de n elementos, donde n es suficientemente grande, está distribuido normalmente con una media μ_q y una desviación estándar σ_q como las descritas en el mismo.

Apéndice F

Seguridad del PRESENT frente al criptoanálisis lineal

Teorema F.1. *[6] Sea ε_{4R} el sesgo máximo de una aproximación lineal de 4 rondas. Entonces,*

$$\varepsilon_{4R} \leq 2^{-7}$$

Demostración. Recordemos que el lema de apilamiento (veáse Lema 3.5) estimaba el sesgo total de una aproximación lineal en la que había n S-boxes:

$$\varepsilon_{1,2,\ldots,n} = 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$$

donde ε_i son los sesgos individuales e independientes de cada S-box como se menciona en el lema. Por criterio de diseño del PRESENT (consúltese [6]), el sesgo de todas las aproximaciones lineales es menor que 2^{-2}, además de que, el sesgo de cualquier aproximación de un sólo bit es menor que 2^{-3}. Denotemos por $\varepsilon_{4R}^{(j)} := \varepsilon_{4R}^{(j)}$ el sesgo de una aproximación lineal de 4 rondas que tiene un número j de S-boxes activas. Se consideran tres casos, dependiendo del número de S-box activas por ronda:

Caso 1 - Una S-box activa por ronda Supongamos que cada ronda de una aproximación lineal de 4 rondas tiene exactamente una S-box activa, es decir, $j = 4$. Entonces, el sesgo de cada una de las dos S-boxes de las rondas intermedias, la segunda y la tercera, es como mucho 2^{-3}. Aplicando el lema de apilamiento, el sesgo total de la aproximación lineal puede ser acotado como:

$$\varepsilon_{4R}^{(4)} \leq 2^3 \cdot (2^{-3})^2 \cdot (2^{-2})^2 = 2^{-7}$$

Caso 2 - Cinco S-boxes activas en las cuatro rondas Supongamos que hay exactamente 5 S-boxes activas en las 4 rondas. Luego, agrupando las S-boxes como en la figura que he usado en el del diferencial, las S-boxes activas sobre 3 rondas consecutivas no pueden formar un patrón 1-2-1. Para que ocurra esto, las dos S-boxes activas en las rondas intermedias son activadas por la misma S-box y tienen entonces que pertenecer a dos grupos diferentes de S-boxes. Pero si se da el caso, no podrían activar solo una S-box en la siguiente ronda. Consecuentemente, el número de S-boxes activas es tanto 2-1-1-1 o 1-1-1-2, por lo que:

$$\varepsilon_{4R}^{(5)} \leq 2^4 \cdot (2^{-3}) \cdot (2^{-2})^4 = 2^{-7}$$

Caso 3 - Más de cinco S-boxes activas en las cuatro rondas Finalmente, supongamos que hay más de 5 S-boxes activas. Por ello,

$$\varepsilon_{4R}^{(j)} \leq 2^{j-1} \cdot (2^{-2})^j = 2^{-j-1} \leq 2^{-7} \quad \text{para } j > 5$$

La igualdad se alcanza teóricamente para $j = 6$. Es una inecuación estricta para el resto de j.

□

Índice de figuras

Índice de cuadros

Referencias

[1] Martin Albrecht and Carlos Cid, *Algebraic techniques in differential cryptanalysis*, International Workshop on Fast Software Encryption, Springer, 2009, pp. 193 – 208.

[2] Gregory Bard, *Algebraic cryptanalysis*, Springer, 2009.

[3] Alena Bednáriková and Pavol Zajac, *A new representation of s-boxes for algebraic differential cryptanalysis*, Rad Hrvatske Akademije Znanosti i Umjetnosti, Matematicke Znanosti (2021), 33 – 49.

[4] Eli Biham and Adi Shamir, *Differential cryptanalysis of des-like cryptosystems*, Journal of Cryptology **4** (1991), 3 – 72.

[5] Céline Blondeau, Benoît Gérard, and Jean-Pierre Tillich, *Accurate estimates of the data complexity and success probability for various cryptanalyses*, Designs, Codes, and Cryptography **59** (2011), 3 – 34.

[6] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, *Present: An ultra-lightweight block cipher*, Lecture Notes in Computer Science **4727 LNCS** (2007), 450 – 466.

[7] Claude Carlet, Yves Crama, and Peter L Hammer, *Boolean models and methods in mathematics, computer science, and engineering*, vol. 2, ch. Boolean Functions for Cryptography and Error Correcting Codes, Cambridge University Press, 2010.

[8] ———, *Boolean models and methods in mathematics, computer science, and engineering*, vol. 2, ch. Vectorial Boolean Functions for Cryptography, Cambridge University Press, 2010.

[9] Don Coopersmith, *The data encryption standard (DES) and its strenght against attacks*, IBM Journal of research and development **38** (1994), 243 – 250.

[10] Thomas W. Cusick and Pantelimon Stănică, *Cryptographic boolean functions and applications: Second edition*, Academic Press LTD-ELSEVIER SCIENCE LTD, 2017.

[11] Joan Daemen, *Cipher and hash function design strategies based on linear and differential cryptanalysis*, Ph.D. thesis, Universidad KU Leuven, 1995.

[12] Joan Daemen and Vincent Rijmen, *AES proposal: Rijndael*, (1999), 45.

[13] Organización Internacional de Normalización, *Information security — lightweight cryptography — part 2: Block ciphers*, `https://www.iso.org/standard/78477.html`, Accedido el 27 de mayo de 2023.

[14] Maria Eichlseder and Marcel Nageler, *Notas de la asignatura cryptanalysis*, Universidad Técnica de Graz, 2023.

[15] Antonio Florez-Gutiérrez and María Naya-Plasencia, *Improving key-recovery in linear attacks: Application to 28-round present*, Advances in Cryptology - Eurocrypt 2020, Lecture Notes in Computer Science, vol. 12105, Springer International Publishing, 2020, pp. 221–249.

[16] Howard M Heys, *A tutorial on linear and differential cryptanalysis*, Cryptologia **26** (2002), 189 – 221.

[17] Jérémy Jean, *Tikz for cryptographers*, `https://www.iacr.org/authors/tikz/`, 2016.

[18] Pascal Junod, *Statistical cryptanalysis of block ciphers*, Ph.D. thesis, Escuela Politécnica Federal de Lausana, 2005.

[19] David Kahn, *The codebreakers: The comprehensive history of secret communication from ancient times to the internet*, Council Foreign Relat Ions Inc., 1997.

[20] Susan K Langford and Martin E Hellman, *Differential-linear cryptanalysis*, Advances in Cryptology—CRYPTO'94: 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings 14, Springer, 1994, pp. 17–25.

[21] Abdurashid Mamadolimov, Herman Isa, and Moesfa Soeheila Mohamad, *Practical bijective s-box design*, arXiv preprint arXiv:1301.4723 (2013), 584 – 588.

[22] Mitsuru Matsui, *Linear cryptanalysis method for des cipher*, Advances in Cryptology - Eurocrypt 1993, Springer Berlin Heidelberg, 1994, pp. 386–397.

[23] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1996.

[24] Michał Misztal, *The signal to noise ratio in the differential cryptanalysis of 9 rounds of data encryption standard*, Journal of Telecommunications and Information Technology (2006), 49 – 59.

[25] Jorge Nakahara, Pouyan Sepehrdad, Bingsheng Zhang, and Meiqin Wang, *Linear (hull) and algebraic cryptanalysis of the block cipher present*, Cryptology and Network Security: 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings 8, Springer, 2009, pp. 58 – 75.

[26] V Panchami and Mahima Mary Mathews, *A substitution box for lightweight ciphers to secure internet of things*, Journal of King Saud University-Computer and Information Sciences **35** (2023), 75 – 89.

[27] Francisco José Plaza Martín, *Manual de criptografía: fundamentos matemáticos de la criptografía para un estudiante de grado*, Ediciones Universidad de Salamanca, 2021.

[28] Axel Poschmann, *Lightweight cryptography: cryptographic engineering for a pervasive world*, Ph.D. thesis, Ruhr University Bochum, 2009.

[29] Ali Aydin Selçuk, *On probability of success in linear and differential cryptanalysis*, Journal of Cryptology **21** (2008), 131 – 147.

[30] Douglas Robert Stinson and Maura Paterson, *Cryptography: theory and practice*, CRC press, 2005.

[31] Cihangir Tezcan, *Improbable differential cryptanalysis*, Ph.D. thesis, Middle East Technical University, 2014.

[32] Henk CA Van Tilborg and Sushil Jajodia, *Encyclopedia of cryptography and security*, Springer Science & Business Media, 2014.

[33] Meiqin Wang, *Differential cryptanalysis of reduced-round present*, International Conference on Cryptology in Africa, Springer, 2008, pp. 40 – 49.

[34] Meiqin Wang, Yue Sun, Nicky Mouha, and Bart Preneel, *Algebraic techniques in differential cryptanalysis revisited*, Australasian Conference on Information Security and Privacy, Springer, 2011, pp. 120–141.