Universidad Central “Marta Abreu” de las Villas
Facultad de Matemática Física y Computación

TRABAJO DE DIPLOMA

Título: *“El diseño de funciones booleanas criptográficamente fuertes”*

Autora: Nayla Elizabeth Vizcaino Alderete
Tutor: MSc. Guillermo Sosa Gómez

Santa Clara
2011

Hago constar que el presente trabajo fue realizado en la Universidad Central "Marta Abreu" de las Villas como parte de la culminación de los estudios de la especialidad de Licenciatura en Matemática, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

_____

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del Tutor

Firma del Jefe de Seminario donde se defiende el trabajo

_____

Firma del Responsable De información Científico–Técnica

Pensamiento

La obtención de un título o diploma, representa para muchos el término de sus estudios, pero para un verdadero revolucionario dicha formación significa una vida entera consagrada al estudio.

Dedicatoria

A mi abuela Xiomara y a mi madre por haberme guiado por el camino correcto.

Agradecimientos

Debo agradecer:

A mi familia por todo el apoyo brindado durante toda la vida.

De forma especial al MSc. Guillermo Sosa Gómez por la cooperación, su paciencia y conocimiento brindado a este trabajo.

A mi tio Betarte por quererme y cuidarme como un padre.

A Jorge Luis por la comprensión, el apoyo y la paciencia ante mis malcriadeces.

A Néstor por brindarme su cariño y apoyo justo cuando más lo necesito.

A mis amigos.

A Tamara por guiarme durante el transcurso de estos cuatro años.

Y a todos los que de una forma u otra hicieron posible la realización de este trabajo.

A todos muchas gracias.

Resumen

Las funciones booleanas son un área importante de estudio para la Criptografía. Estas funciones, que consiste sólo de uno y de cero, son el corazón de numerosos sistemas de cifrado y su capacidad para proporcionar una comunicación segura. Las funciones booleanas tienen aplicación en una variedad de sistemas, incluyendo sistemas de Cifrado de bloques, Cifrados de flujo y las funciones hash. El estudio continuo de las funciones booleana para la Criptografía, es fundamental para la prestación de una comunicación segura en el futuro. Esta tesis presenta una investigación sobre el análisis de funciones booleanas y en el análisis particular del diseño y aplicación de funciones booleana para la Criptografía. La tesis se iniciará con una breve descripción de la teoría de funciones booleanas. Seguido, se trata del desarrollo de bloques de criptosistemas en la Criptografía simétrica. Los bloques pueden verse como funciones vectoriales booleanas, también llamadas S-cajas para una salida multidimensional y nuestro objetivo es buscar aquellas funciones de mayor interés para la construcción de bloques o S-cajas con una alta confiabilidad.

Abstract

Boolean functions are an important area of study for cryptography. These functions, consisting merely of one's and zero's, are the heart of numerous cryptographic systems and their ability to provide secure communication. Boolean functions have application in a variety of such systems, including block ciphers, stream ciphers and hash functions. The continued study of Boolean functions for cryptography is therefore fundamental to the provision of secure communication in the future. This thesis presents an investigation into the analysis of Boolean functions and in particular, analysis of affine transformations with respect to both the design and application of Boolean functions for cryptography. The thesis will begin with a brief overview of Boolean function theory; including an introduction to the main theme of the research, namely the affine transformation.

Índice

Contenido

Índice

Introducción

Las funciones booleanas desempeñan un papel importante en la Criptografía moderna por su capacidad para satisfacer, con mayor seguridad, la demanda continua de las comunicaciones. El estudio de las funciones booleanas, tanto desde el punto de vista teórico como práctico, es fundamental para proteger las aplicaciones de cifrado tales como sistemas de cifrado de bloques, cifrados de flujo y las funciones hash.

Aunque desde finales de la década de 1980 ha podido apreciarse un creciente interés por investigar en esta área, existen todavía muchos problemas en relación con el diseño y análisis de funciones booleanas para la Criptografía. El nivel de seguridad alcanzado en las aplicaciones basadas en estas funciones se mide por la calidad de las propiedades combinatorias dentro de las mismas. La selección de funciones booleanas con fuertes propiedades criptográficas reduce la eficacia de los ataques de criptoanálisis avanzados, incluyendo el criptoanálisis lineal [1] y el criptoanálisis diferencial [2].

Por otra parte, la adecuación específica de una función booleana para su uso en Criptografía consiste, normalmente, en la evaluación de varias propiedades de la forma normal algebraica (ANF) y de la Transformada de Hadamard (TH). Asimismo, también se ofrecerá el examen preliminar de la teoría de funciones booleanas, incluyendo su representación, la ANF, el TH y la función de auto correlación (AC), así como las distintas propiedades de la importancia de cifrado derivados de cada uno.

El diseño y análisis de las funciones booleanas para las aplicaciones criptográficas normalmente implican una cantidad considerable de procesamiento computacional. En particular, debido al gran número de variables de entrada que este análisis supone, se requiere una elevada demanda de recursos informáticos.

La construcción de funciones criptográficamente útiles es también una tarea difícil. Una gama de técnicas algebraicas y técnicas heurísticas están disponibles actualmente para la construcción de tales funciones, sin embargo, estos métodos pueden ser complejos, computacionalmente difíciles de aplicar y no siempre producen una variedad suficiente de funciones.

El rápido crecimiento de la comunicación electrónica (principalmente internet) tiene como resultado que el problema de la seguridad de la información sea de una importancia práctica creciente. Los mensajes que se intercambian en todas partes del mundo, y que son públicamente accesibles a redes de computadoras, deben ser mantenidos confidencialmente y protegidos contra manipulación

[3]. El comercio electrónico requiere firmas digitales que son válidas para la ley y protocolos seguros de pago. La Criptografía moderna proporciona solución a todos estos problemas.

La Criptografía es el arte y la ciencia de ofuscar mensajes. Antes de la época de las computadoras, un mensaje era una cadena de letras y era encriptado reemplazando cada letra con otra o un número. En la época de las computadoras, un mensaje es una cadena binaria (bitstring) en una computadora, y es encriptado reemplazando un bitstring por otro, normalmente de la misma longitud [4]. Es importante que estos mensajes no puedan ser alterados y que la seguridad de la información se mantenga. La Criptografía moderna brinda métodos matemáticos para solucionar, de manera relativa, estos problemas.

Históricamente la Criptografía ha sido vista como un arte más que una ciencia, pero existiendo siempre dos grupos bien diferenciados. Los criptógrafos, cuyo trabajo es diseñar sistemas criptográficos; y los criptoanalistas, cuyo trabajo es tratar de infringir estos sistemas criptográficos[5]. Así, el arte y la ciencia de la Criptografía consisten en dos mundos. Por un lado, el mundo de las comunicaciones legales, como usuarios que intercambian mensajes de datos bancarios, que puede ser visto como un mundo abierto y soleado (Criptografía). Por otro lado, el mundo oscuro del enemigo (criptoanálisis) que ilegalmente trata de interceptar los mensajes y hacer todo tipo de cosas maliciosas. Para la gente del mundo legal, es conveniente que el enemigo entienda muy poco de los mensajes, sin embargo, al enemigo, por otro lado, le gustaría tener fácilmente descifrable estos mensajes. Por tanto, la Criptografía es una lucha continua entre estos dos mundos donde el éxito obtenido por uno conduce a la necesidad de reforzar los métodos de encriptación mientras, para el otro, en cambio, esto constituye un nuevo reto. En la referencia [7] se pueden encontrar aspectos históricos más precisos de la Criptografía.

El campo de la Criptografía se ha expandido en los últimos años. Las disciplinas matemáticas que están involucradas con la Criptografía incluyen teoría de números, teoría de grupos, lógica combinatoria, teoría de la complejidad, teoría de la información y otras áreas de la matemática. Este campo puede ser visto, en la actualidad, como una subdivisión de la matemática aplicada y las ciencias de la computación [5].

Así, el **problema científico** de nuestra investigación es:

¿Cómo proceder respecto a la construcción de las funciones booleanas, en su uso correcto en la Criptografía, de forma que contribuya al desarrollo de nuevos sistemas criptográficos?

En este trabajo se hace una revisión de algunos aspectos teóricos que relacionan las funciones booleanas y la Criptografía. El tema se enmarca dentro del análisis de Fourier de las funciones booleanas. Más exactamente, en el caso de característica 2 de la transformada discreta de Fourier (la

transformada de Hadamard). Esta herramienta matemática permite estudiar algunas propiedades criptográficas de las funciones booleanas.

Cuatro son las propiedades criptográficas deseables en las funciones booleanas: grado algebraico, inmunidad-correlación, balance y no linealidad. Las propiedades anteriores son a menudo investigadas mediante la transformada de Hadamard [8]. Las funciones booleanas que tienen estas propiedades son resistentes a los ataques criptoanalíticos. La Criptografía, entonces, necesita la manera de buscar buenas funciones booleanas que sean resistentes a estos ataques.

Existen, además, otras propiedades criptográficas como avalancha y (no existencia de) estructuras lineales diferentes de cero. Un ligero comentario de estas propiedades puede encontrarse en[9]. Existen aún varios problemas abiertos sobre funciones booleanas que son de importancia primordial para la Criptografía [10]. Muchos de ellos están relacionados con problemas de complejidad computacional.

Ideas de investigación

"El buen diseño de funciones booleanas criptográficamente fuertes contribuirá a aumentar la seguridad de los sistemas criptográficos que se utilizan."

Como **objetivo general** se propone entonces:

"Mejorar la comprensión de las funciones booleanas, proporcionando nuevas perspectivas, la búsqueda superior de funciones booleanas con óptimas propiedades criptográficas."

Para lograr dicho objetivo general, se proponen los siguientes **objetivos específicos**:

1. Estudiar las herramientas matemáticas necesarias para la comprensión de las funciones booleanas.
2. Estudiar las funciones booleanas y sus propiedades criptográficamente deseables.
3. Introducir nuevas ideas en la búsqueda de funciones booleanas usando métodos heurísticos.

Para dar cumplimiento a estos objetivos fue necesario plantearse y solucionar las siguientes **tareas de investigación**:

1. Detección del problema y recolección de la información necesaria para su posible solución.
2. Estudio de conceptos matemáticos del Álgebra, de las funciones booleanas, inteligencia artificial y de la Criptografía.
3. Formulación de una teoría matemática que sustente la solución del problema planteado.

El primer paso para la realización de este trabajo fue la confección del marco teórico, para ello se realizó una amplia revisión de la literatura consultando libros, artículos y páginas de internet, entre

otras fuentes. Los elementos esenciales se encuentran expuestos de manera resumida en el primer capítulo de la presente tesis.

Novedad científica:

a. La elaboración de toda una teoría que permite determinar funciones booleanas deseables de manera sencilla y con menor costo computacional.

b. Se plantean nuevas estrategias de búsqueda de funciones booleanas con propiedades criptográficas fuertes usando métodos heurísticos.

c. Nuevas posibilidades de aplicación de las temáticas que se abordan en la tesis en investigaciones dirigidas por la Dirección de Criptografía del MININT y generalización de estos resultados.

La tesis está estructurada en: Introducción, tres Capítulos, Conclusiones, Recomendaciones y Referencias Bibliográficas.

En el primer capítulo abordamos la teoría necesaria de las funciones booleanas así como su aplicación en el mundo de la Criptografía, dando una panorámica breve de este tema reflejando su utilización en varios de los cifradores existentes en la actualidad.

En el segundo capítulo se incluyen las definiciones básicas y las propiedades de las matrices de Hadamard, en forma abreviada. La emoción puramente intelectual y el desafío de encontrar nuevas matrices de Hadamard y la confirmación de la conjetura de Hadamard se ve reforzada por el conocimiento de que son maravillosamente útiles. También se enfocan las matrices de Hadamard en el mundo de la Teoría de la Información haciendo una simple introducción al cálculo de entropías en las matrices. Así también se da una breve idea de qué son los métodos heurísticos y los autómatas celulares.

El tercer capítulo está dedicado a las propiedades de las funciones booleanas y se introduce las nuevas ideas para desarrollar funciones booleanas utilizando métodos heurísticos.

La adecuada formación de un criptógrafo en las áreas de las matemáticas y del entendimiento de la importancia de las funciones booleanas le ayuda a crear una atmósfera científica que contribuye a desarrollar su rigor y prestigio en esta rama del saber. Así como a prepararse para realizar un trabajo realmente útil y calificado que requiera de iniciativas.

La importancia social de este trabajo radica en que le permite a un criptógrafo conocer cuáles deben ser las funciones booleanas que él requiere tomando en consideración todo lo que se expone aquí, de forma tal que le proporcione una sólida formación científica con excelentes resultados.

Capítulo 1. Funciones booleanas

1.1 Introducción

En diversos cursos en las carreras de Ingeniería, Computación, Física, Química, Economía, Matemáticas, etc., se encuentra el concepto de *función* en varios contextos. En muchos casos estas funciones toman valores en (el campo de) los números reales ($\mathbb{R}$), por ejemplo las funciones reales de una variable real $f: \mathbb{R} \to \mathbb{R}$; las funciones de varias variables reales con valores en los números reales $f: \mathbb{R}^n \to \mathbb{R}$, donde $\mathbb{R}^n$ es el espacio cartesiano de dimensión $n$sobre los números reales; o de manera más general funciones vectoriales sobre los reales,$f: \mathbb{R}^n \to \mathbb{R}^m$. En cursos posteriores se estudian, por ejemplo funciones de variable compleja, sobre grupos y otras estructuras tales como variedades (topológicas, complejas, algebraicas, diferenciables, etc.). Tanto los números reales, los números complejos como las variedades mencionadas anteriormente son, en general, conjuntos cuya cardinalidad no es finita.

En pocos cursos básicos se mencionan o estudian funciones sobre conjuntos cuya cardinalidad es finita. En este trabajo se introducirá el concepto de función cuyo dominio es un conjunto finito y su contradominio es también un conjunto finito cuya cardinalidad es 2. Nos referimos a los números binarios como contradominio y a estructuras construidas a partir de estos como dominio de tales funciones. Los números binarios, como el lector debe saber, son de suma importancia en la actualidad, con el uso de las computadoras, el manejo de información en formato digital y cuestiones relacionadas.

Como aplicaciones de este tipo de funciones, se mencionan brevemente debido al espacio, su uso en dos áreas que hoy en día juegan un papel muy importante en varias actividades de la vida cotidiana: la *Criptografía,* la cual está relacionada con la seguridad en el manejo de la información, y la *Teoría de Códigos Detectores-Correctores de Errores,* que como su nombre lo indica, trata de la detección y corrección de los errores que se adquieren en la transmisión de información, en ambos casos por cualquier canal de transmisión convencional que se use.

1.2 El espacio F_2^n

Al conjunto de los números binarios, también llamados *bits,* denotado por $\mathrm{F}_2 = \{0,1\}$, se le puede dotar de dos operaciones, la suma$(\oplus)$y el producto$(*)$ definidos en las siguientes tablas:

$\oplus$	0	1
0	0	1
1	1	0

$*$	0	1
0	0	0
1	0	1

Con estas dos operaciones, la terna $(\mathrm{F}_2, \oplus, *)$, adquiere la estructura algebraica de *campo.* Es decir, satisface las mismas propiedades que los números reales con las operaciones usuales de suma y producto, o bien, los números complejos con las correspondientes operaciones. Para el lector familiarizado con algunos conceptos matemáticos, particularmente de Teoría de Números, le será fácil identificar a los números binarios con los enteros módulo 2 y sus operaciones usuales: $(\mathbb{Z}_2, +, *)$

Veamos ahora como construir una estructura algebraica también interesante a partir de los números binarios. Primero recordemos que si n es un entero positivo y $\mathbb{R}^n$ es el producto cartesiano de $\mathbb{R}$ consigo mismo n veces, se puede definir en $\mathbb{R}^n$ una operación de suma de vectores en la forma natural: coordenada a coordenada. También, si $\alpha \in \mathbb{R}$ (un escalar) y $x \in \mathbb{R}^n$, se define $\alpha \boldsymbol{x}$ multiplicando cada coordenada de $\boldsymbol{x}$ por α (producto por escalares). Con estas dos operaciones el conjunto $\mathbb{R}^n$ adquiere la estructura de *espacio lineal* (vectorial) sobre el campo de los números reales, cuya dimensión es n.

En forma similar se procede en el caso de los números binarios: si $n$es un entero positivo, sea

$$\mathrm{F}_2^n = \{\boldsymbol{a} = (a_1, \dots, a_n) : a_i \in \mathrm{F}_2\}$$

el producto cartesiano de F_2 consigo mismo n veces. Obsérvese que cada elemento de F_2^n tiene $n$entradas que consisten de $0's$ y $1's$. De igual forma que en el caso de $\mathbb{R}^n$, al conjunto F_2^n se le puede dar la estructura de *espacio lineal* sobre los números binarios F_2 con las siguientes operaciones:

si$a = (a_1, \dots, a_n)$, $\boldsymbol{b} = (b_1, \dots, b_n)$ son elementos de F_2^n y $\alpha \in \mathrm{F}_2$ entonces:

$$\mathbf{a} \oplus \mathbf{b} = (a_1 \oplus b_1, \dots, a_n \oplus b_n)$$

$$\alpha a = (\alpha * a_1, \dots, \alpha * a_n)$$

Es fácil ver que una base (natural) del F_2-espacio lineal F_2^n es:

$$\boldsymbol{e_1} = (1, \dots, 0), \boldsymbol{e_2} = (0,1, \dots, 0), \dots, \boldsymbol{e_n} = (0, \dots, 1)$$

Y su dimensión (sobre F_2) es n. Obsérvese que la cardinalidad de F_2^n es 2^n. Por ejemplo, si $n = 2$, entonces:

$$\mathrm{F}_2^2 = \{(0,0), (0,1), (1,0), (1,1)\}$$

Si $n = 8$, el espacio F_2^8 tiene cardinalidad $2^8 = 256$ y sus elementos se pueden identificar con los elementos del código ASCII muy usado actualmente.

1.3 Funciones Booleanas

Si se procediera, como en el caso de funciones reales de variable real, a estudiar funciones $f: \mathrm{F}_2 \to \mathrm{F}_2$, es fácil ver que no existen muchas de ellas, ya que la cardinalidad de F_2 es 2 (¿cuáles son tales funciones?), por consiguiente su estudio no es muy interesante. Esto nos conduce a la siguiente:

Definición 1.1. *Si n es un entero positivo, una función booleana es simplemente:*

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

El conjunto de tales funciones se denotara por $\mathcal{B}_n$

Veamos algunos ejemplos de funciones booleanas:

1. Si $n \geq 2$, sea $A: \mathrm{F}_2^n \to \mathrm{F}_2$ definida como

   $$A(x_1, \dots, x_n) = a_0 + a_1 x_1 + \cdots + a_n x_n,$$

   Donde $a_0 + a_1 + \cdots + a_n$ son elementos del campo de los números binarios F_2. A estas funciones se les conoce como *afines. Si* $a_0 = 0$ la funcion es *lineal.*

2. Si $n = 2$, sea $f: \mathrm{F}_2^2 \to \mathrm{F}_2$definida como

   $$f(x_1, x_2) = 1 + x_1 + x_1 x_2.$$

   Así por ejemplo, $f(1,0) = 1 + 1 + 0 = 0$

3. Si $n = 5$, sea $g: \mathrm{F}_2^5 \to \mathrm{F}_2$ definida como

   $$f(x_1, x_2, x_3, x_4, x_5) = 1 + x_1 x_5 + x_2 x_4 + x_1 x_3 x_5$$

A continuación se mencionan algunas propiedades básicas de las funciones booleanas. Para mayores detalles el lector puede consultar por ejemplo [11] y [12].

1. Toda función booleana se puede expresar como una función polinomial.
2. La potencia a la cual aparece cada variable x_i en una función booleana $f(x_1, \dots, x_n)$ es igual a 1. Esto es debido a la estructura del campo de los números binarios ya que para todo elemento $a \in \mathrm{F}_2$ se tiene que $a^2 = a$.
3. Cada elemento f de $\mathcal{B}_n$ se puede identificar con un elemento de $\mathrm{F}_2^{2^n}$: en efecto, si $n$es un entero positivo sea $N = 2^n$y sea $\mathrm{F}_2^n = \{p_1, \dots, P_N\}$. Entonces a la función booleana $f: \mathrm{F}_2^{\mathrm{n}} \to \mathrm{F}_2$se le asocia el elemento $(f(P_1), \dots, f(P_N))$ de F_2^N. Es obvio que para cada elemento $\boldsymbol{y} = (y_1, \dots, y_N) \epsilon\, \mathrm{F}_2^N$ hay una función $g \in \mathcal{B}_{\mathrm{n}}$ tal que $(g(P_1), \dots, g(P_N)) = y$.

   A modo de ilustración consideremos la función $f: \mathrm{F}_2^2 \to \mathrm{F}_2$ dada por

$f(x_1, x_2) = 1 + x_1 x_2$. Como $\mathrm{F}_2^2 = \{(0,0), (1,0), (0,1), (1,1)\}$, la función f se identifica con el vector:

$$(f(0,0), f(1,0), f(0,1), f(1,1)) = (1,1,1,0)$$

De lo anterior se sigue que la cardinalidad del conjunto de funciones booleanas $\mathcal{B}_n$ es 2^{2^n}. Por ejemplo, si $n = 3$, hay $2^{2^3} = 2^8 = 256$ funciones booleanas de 3 variables. El lector puede pasar algunos minutos y determinar todas las funciones booleanas en 3 variables. (¿Cuántas funciones booleanas hay de 10 variables?).

4. Si $\mathbf{x} = (x_1, \ldots, x_n) \in \mathrm{F}_2^n$, el conjunto $S(\mathbf{x}) = \{x_i \neq 0\}$ se llama el *soporte* de $\mathbf{x}$. Con ayuda del soporte de un elemento de F_2^n se puede definir una métrica en el espacio lineal F_2^n de la siguiente manera: si $\mathbf{x}, \mathbf{y} \in \mathrm{F}_2^n$, entonces:

$$d(\boldsymbol{x}, \boldsymbol{y}) = |S(\mathbf{x} + \mathbf{y})|$$

donde"|.|" indica cardinalidad. Se puede ver que la función d es en efecto una métrica (en el sentido usual), la cual se llama la distancia de *Hamming.* Esta distancia es muy importante en la Teoría de Códigos Detectores-Correctores de Errores ya que permite determinar el número de errores que puede detectar y corregir uno de tales códigos (cf.[12]). Asimismo se puede definir la distancia de Hamming en el espacio de funciones booleanas, la cual es de suma importancia para su estudio. Una clase interesante de funciones booleanas son las llamadas *funciones bent,* las cuales tiene la propiedad de ser las más distantes de las funciones lineales, con la distancia de Hamming. Existen en la literatura una gran cantidad de resultados sobre este tipo de funciones y hasta la fecha no se conoce exactamente la estructura de tales funciones, ni su cardinalidad cuando el número de variables es ≥ 10 (¡problema abierto!)

1.4 Funciones booleanas, seguridad y trasmisión de información

En esta sección se describirían brevemente algunas conexiones entre las funciones booleanas y la transmisión y seguridad de la información cuando esta se envía por un canal convencional, el cual en general es "ruidoso" y poco seguro.

1.4.1 Funciones booleanas y Criptografía

La *Criptografía,* cuyo significado proviene del griego *krypto, (**esconder**) y* graphein**, (escritura),** se ha empleado desde hace mucho tiempo, por ejemplo, se sabe que Julio Cesar enviaba mensajes cifrados ([13], [14]). Uno de los principales objetivos de la Criptografía es la comunicación segura entre entidades que usan canales convencionales (inseguros y ruidosos) tales como el teléfono, fax, satélite, internet, etc., de tal manera que entidades no autorizadas no puedan conocer el contenido de la información original. Una de sus aplicaciones ha estado relacionada con cuestiones bélicas,

pero con el desarrollo de la computación y la introducción de la Criptografía de *llave pública* ([15]), su enfoque y aplicación ha cambiado en las últimas décadas. Las aplicaciones más importantes de la Criptografía moderna se encuentran en el comercio, banca, y correo electrónico, declaración de impuestos, firma y factura digital, tarjetas inteligentes, entre otras.

Actualmente los sistemas criptográficos (cifrado) se dividen en dos grandes grupos: los sistemas de *llave privada* y los de *llave pública.* Entre los más conocidos en el primer grupo se encuentran los cifrados en "cascada" (streamciphers)([16]), el Data Encryption Standard (DES) ([17]) y el Advance Encryption Standard (AES) ([18]) estos últimos son cifrado de bloques. En el segundo grupo se incluye el sistema RSA ([19]) y el basado en curvas elípticas ([20], [16], [21], [22]) también ambos cifrados de bloques.

1.4.2 Cifrado de Vernam

El cifrado de Vernam es un sistema en cascada y tiene una gran relación con funciones booleanas. Para mayores detalles el lector puede consultar por ejemplo [16].

En el cifrado de Vernam, para cada texto original que se desea cifrar, el cual es expresado en bits, se produce una llave secreta en forma aleatoria de la misma longitud (en bits) que el texto a cifrar, la cual se suma con el texto original usando la aritmética binaria, produciendo de esta manera la información cifrada (este método fue usado por oficiales de E.U. y la URSS durante la llamada "guerra fría").

Para producir la llave secreta las partes interesadas seleccionan un método para obtener sucesiones pseudoaleatorias a partir de "semillas" (también secreta). Una forma de producir tales sucesiones es por medio de los llamados "Linear Feedback Shift Registers" (LFSR), que son criptográficamente débiles. Para hacerlos más robustos se usan funciones booleanas, las cuales tienen como argumento (input) $n$bits $x_1, \dots, x_n$producidos por nLFSR's y cuyo valor (output) $s = f(x_1, \dots, x_n)$ es uno de los elementos de la sucesión que se desea generar. Este proceso se repite hasta obtener una sucesión de la longitud deseada que se usa como la llave para cifrar.

1.4.3 Cifrado en bloques

Este tipo de cifrado tiene como entrada (input) el texto original, el cual se puede pensar como una colección de vectores binarios de la misma longitud $(x_1, \dots, x_n)$. Para cada uno de estos vectores se produce el texto cifrado $(y_1, \dots, y_n)$, donde $y_i = f_i(x_1, \dots, x_n, k_1, \dots, k_n)$ siendo $(k_1, \dots, k_n)$ la llave (secreta) para cifrar y las $f_i's$ funciones booleanas. El texto cifrado correspondiente al original es la concatenación de estos bloques cifrados. Entre los sistemas de cifrado más comunes en esta categoría se encuentran el Data Encryption Standard (DES) ([17]) y el Advance Encryption Standard

(AES) ([18]). Para el sistema DES, $n = 64$ y las llamadas *S*-cajas juegan un papel muy importante. Dichas S-cajas son funciones booleanas $S: \mathrm{F}_2^{48} \to \mathrm{F}_2^{32}$usándose 32 de ellas. En este sistema también se emplean otras 32 funciones booleanas $f_i: \mathrm{F}_2^6 \to \mathrm{F}_2^4$ ([17],[16]). En el caso del sistema AES (estándar), la entrada es de 128 bits divididos en 16 bloques de 8 bits cada uno. Cada *S*-caja es la concatenación de 16 funciones booleanas de 8 variables cada una. Cabe mencionar que en este cifrado también juega un papel muy importante la estructura y propiedades de los campos de Galois, también llamados campos finitos ([18], [23]).

1.4.4. Funciones Booleanas y Teoría de Códigos

En la teoría de códigos un bloque de información de k bits, antes de ser transmitido, es procesado (codificado) para producir un bloque de $n > k$ bits, el cual consiste de $k$bits de información y $n - k$ bits de redundancia. Estos bits de redundancia se obtienen a partir de los bits de información. Una vez codificada la información original es trasmitida por el canal convencional, el cual en general es "ruidoso". Estos bits de redundancia permiten al receptor detectar (y corregir) los errores adquiridos durante la trasmisión o almacenamiento de la información (por ejemplo en CD's, DVD's, HD's, etc.).

En la actualidad hay varias clases de códigos usados en la detección y corrección de errores en la transmisión de información, pero una de las más usadas es la de los Códigos Lineales en bloques. Un código lineal binario de longitud n dimensión k y distancia mínima d, es simplemente un subespacio lineal *C* de F_2^n de dimensión k que tiene palabras (vectores) de distancia de Hamming al origen igual a d. Los códigos lineales no sólo se pueden definir sobre el campo de los números binarios, sino también sobre cualquier campo finito y una amplia clase de anillos finitos, la cual incluye a los de Galois y de cadena, entre otros ([24], [25]). Por supuesto, desde un punto de vista práctico, los códigos binarios son los más importantes.

Una clase de códigos lineales con varias aplicaciones, entre las cuales se incluyen CD's y DVD's, es la de los códigos de Reed-Solomon (y sus variantes) ([12], [26]). Otra clase de códigos lineales que han sido usados por naves espaciales para enviar información, en particular imágenes, de cuerpos celestes, es la de los códigos de Reed-Muller. En la literatura se encuentran varias definiciones de estos códigos (todas equivalentes), y una de ellas está dada en términos de funciones booleanas:

CONCLUSION

Las funciones booleanas juegan un papel muy importante en el diseño de sistemas de cifrado (Criptografía) para tener seguridad en el manejo de la información, así como en el diseño de códigos

lineales usados en la detección y corrección de errores adquiridos en la transmisión de información. Cabe mencionar que en ambos casos hay una gran cantidad de resultados y problemas interesantes que relacionan las funciones booleanas con otras áreas tanto de Matemáticas como de Ciencias de la Computación e Ingeniería, entre las que se pueden mencionar: Análisis de Fourier (discreto), Combinatoria, Teoría de Gráficas, Teoría de Números, Algebra Moderna, Geometría Algebraica, diseño y análisis de algoritmos, "zero-knowledge", complejidad computacional, diseño de circuitos, etc.

Capítulo 2. Herramientas Matemáticas.

2.1. Transformada y matrices de Hadamard

En la literatura las referencias de la Transformada de Hadamard, las matrices de Hadamard y las funciones relacionadas con ellas son extensas. Estas referencias abarcan los asuntos de procesamiento de señal, la codificación y transmisión de imagen, el análisis estadístico, el procesamiento y codificación de la voz, los circuitos lógicos, la filtración, las olas electromagnéticas, los dispositivos ópticos y su modelado matemático, el procesamiento de radar, la sismología, la holografía, el reconocimiento de patrones, la compresión de datos y el análisis químico [27].

La serie de funciones de Hadamard puede aplicarse a muchas áreas dónde las técnicas sinusoidales habían dominado anteriormente. Esto es, por ejemplo, en el diseño de equipamiento digital para la comunicación y aplicaciones computacionales [28].

En el procesamiento de imágenes y el reconocimiento de patrones la motivación para usar otras transformadas como la de Fourier es que reduce el tiempo computacional para una resolución dada, o para aumentar la resolución sin incurrir en la penalidad de tiempo de cómputo largo. La transformada de Hadamard se ha usado de manera efectiva para satisfacer estos requisitos. Un acercamiento general al reconocimiento de patrones es llevar a cabo una transformación de un modelo de patrones y la autocorrelación de conjuntos transformados de valores para determinar el grado de reconocimiento, en lugar de intentar encontrar auto correlaciones de las señales originales. Pueden encontrarse ahorros sustanciales en el tiempo de cómputo de esta manera. Usada de la manera correcta, la transformada de Hadamard puede reducir la complejidad de dos procesos dimensionales al nivel de adiciones y substracciones de coeficientes.[29]

2.2. Transformada de Hadamard

La Transformada de Hadamard es quizás la más conocida de las transformadas ortogonales no sinusoidales. La Transformada de Hadamard ha ganado la prominencia en las aplicaciones en el procesamiento de señales digitales, dado que sólo usa sumas y substracciones para computar. Por consiguiente, su implementación en el hardware es muy simple [30].

Sea F_2^n el espacio vectorial de dimensión n sobre el campo binario F_2. Para dos vectores $a = (a_1, \dots, a_n)$ y $b = (b_1, \dots, b_n)$ de F_2^n, nosotros definimos el producto escalar $a \cdot b = a_1 b_1 \oplus \dots \oplus a_n b_n$, dónde la multiplicación y suma $\oplus$ (llamada XOR) son sobre F_2.

Definición 2.1. La transformada de Hadamard de una función f en F_2^n (donde los valores de f pueden ser 0 y1) es una aplicación $H(f)\colon \mathrm{F}_2^n \to \mathbb{R}$, definida por

$$H(f)(h) = \sum_{x \in \mathrm{F}_2^n} f(x)\,(-1)^{h \cdot x}, \qquad (2.1)$$

Conociendo que la forma polar de una función booleana f es $\hat{f}(x) = (-1)^{f(x)}$, podemos entonces escribir la transformada de Hadamard de la función polar

$$H\big(\hat{f}\big)(h) = \sum_{x \in F_2^n} (-1)^{h \cdot x + f(x)}$$

En la cual se definen los coeficientes de f con respecto a las bases orto normales del carácter de grupo $Q_x(h) = (-1)^{h \cdot x}$; f puede ser recuperada por la transformada inversa de Hadamard

$$f(x) = 2^{-n} \sum_{h \in \mathrm{F}_2^n} H(f)(h)\,(-1)^{h \cdot x} \qquad (2.2)$$

El espectro de Hadamard de f es una lista de los 2^n coeficientes de Hadamard dado por (2.1) con h variable. Las funciones booleanas simples son funciones constantes 0 y 1. Obviamente, $H(0)(u) = 0$ y los coeficientes de Hadamard para la función 1 son dados por el siguiente lema enunciado en [31].

Lema 2.1. Si $h \in \mathrm{F}_2^n$, tenemos $\sum_{u \in \mathrm{F}_2^n} (-1)^{u \cdot h} = \begin{cases} 2^n, & si\ h = 0 \\ 0, & en\ otro\ caso \end{cases}$

Demostración

Primero, si $h = 0$, entonces todos los sumandos son1. Ahora, asumamos que $h \neq 0$, y consideremos el hiperplano $M = \{u \in \mathrm{F}_2^n : u \cdot h = 0\}$, $\bar{M} = \{u \in \mathrm{F}_2^n : u \cdot h = 1\}$. Entonces, para cualquier $u \in M$, el sumando es1, y para cualquier $u \in \bar{M}$, el sumando es -1. La cardinalidad de $M, \bar{M}$ coinciden, que es 2^{n-1}, así tenemos el lema [32].

Un cálculo directo del espectro de Hadamard completo utilizando (2.1) implica una complejidad de N^2 pasos, con $N = 2^n$. Sin embargo tal y como ocurre con la transformada rápida de Fourier (ver en [33]), es posible definir un procedimiento rápido para el cálculo de la transformada de Hadamard que puede ser computado con únicamente $NLog(N)$ pasos. Para lograr esa aceleración, la Transformada Rápida de Hadamard (TRH) utiliza el concepto de diagrama de mariposa. Un diagrama de mariposa de tamaño 2 (el tamaño más pequeño), toma dos bits de entrada (x_0, x_1), y produce dos bits de salida (y_0, y_1), de la siguiente manera:

$$\begin{cases} y_0 = x_0 + x_1 \\ y_1 = x_0 - x_1 \end{cases}$$

En general, la TRH divide recursivamente el cálculo de un vector de tamaño $n = rm$, en r transformaciones más pequeñas de tamaño m, donde $r$es la base de la transformación. Estas r transformaciones pequeñas son combinadas utilizando diagramas de mariposa de tamaño r, las cuales a su vez, son TRH de tamaño r.

2.3. Las matrices de Hadamard

Una matriz de Hadamard es una matriz H de orden $n \times n$ con las entradas ± 1 qué satisface $HH^t = nI$, donde H es real, simétrica y las filas y las columnas de las mismas son ortogonales dos a dos [28].

Estas matrices deben su nombre a un teorema del propio Hadamard:

Teorema 2.1. Sea $X = (x_{ij})$ una matriz real de orden $n \times n$ cuyas entradas satisfacen que $|x_{ij}| \leq 1$ para toda i , j. Entonces $|det(X)| \leq n^{\frac{n}{2}}$. La igualdad se tiene si y solo si X es una matriz de Hadamard [29].

Sea $x_1, \dots, x_n$ las filas de X, entonces por la geometría euclidiana simple, $|det(X)|$ es el volumen del paralelepípedo con lados $x_1, \dots, x_n$; así $|det(X)| \leq |x_1| \cdots |x_n|$, donde $|x_i|$ es la longitud euclidiana de x_i ; la igualdad se tiene si y solo si $x_1, \dots, x_n$ son mutuamente perpendiculares.

Por hipótesis $|x_i| = ({x_{i1}}^2 + \cdots + {x_{in}}^2)^{1/2} \leq n^{1/2}$, con igualdad si y solo si $|x_{ij}| = 1$ para toda j.

Las matrices de Hadamard encuentran las aplicaciones naturales en códigos error-corrector [16] y proporciona " los códigos del bi-ortogonales". Hay trabajo extenso en encontrar el matrices con entradas ± 1 qué tiene el máximo determinante. Éste es una subclase del problema del Hadamard. Hay también muchas generalizaciones y parientes del problema. Por ejemplo, [34] los estudios el problema difícil computacionalmente de encontrar un simplex j-dimensionales más grande en un cubo d-dimensional dado.

2.4. Equivalencia de matrices de Hadamard

En las matrices de Hadamard se realizan varias operaciones. De estas las que conservan la propiedad de Hadamard son:

a) Permutar filas, y cambiar el signo de algunas de ellas;

b) Permutar columnas, y cambiar el signo de algunas de ellas;

c) Transponer

Dos matrices de HadamardH_1 y H_2 son equivalentes si una puede ser obtenida a partir de otra por operaciones de tipo a) y b); es decir, si $H_2 = P^{-1}H_1Q$, donde P y Q son matrices monomiales (tienen solamente un elemento no nulo en cada fila o columna) con entradas no nulas ± 1.

El grupo de automorfismo de una matriz de Hadamard$H$ consiste en el grupo de todos los pares (P, Q) de matrices monomiales con entradas no nulas ± 1 satisfaciendo $H = P^{-1}HQ$; la operación de grupo está dada por $(P_1, Q_1) \circ (P_2, Q_2) = (P_1P_2, Q_1Q_2)$[35].

Note que hay siempre un automorfismo$(-I, -I)$, el cual queda en el centro del grupo de automorfismo[36].

2.5. Construcción Sencilla

Las investigaciones en el área de las matrices de Hadamard y sus aplicaciones han ido creciendo rápidamente, especialmente durante las tres últimas décadas. Estas matrices pueden ser transformadas para producir los diseños de bloques incompletos, los t-diseños, los diseños de Youden, los diseños ortogonales de F-cuadrado, el diseño óptimo de peso, el conjunto maximal de dos a dos conjuntos de variables aleatorias independientes con la medida uniforme, el error, la corrección y la detección de los códigos, las funciones de Walsh, y otros objetos matemáticos y estadísticos. En este trabajo se estudia la existencia de matrices de Hadamard y algunas de sus aplicaciones [27].

Sin embargo, la conjetura de la matriz Hadamard es de distinta naturaleza. A pesar de que una serie de ideas asociadas se han desarrollado en la búsqueda de matrices Hadamard, la existencia misma de estas matrices tiene amplias consecuencias en muchos campos de investigación, tales como la teoría del diseño óptimo, la teoría de la información y la teoría de grafos [29].

La construcción más simple de nuevas matrices de Hadamard son las llamadas matrices de Sylvester-Hadamard que se construyen a través del producto de Kronecker. En general, si $A = (a_{ij})$ y $B = (b_{kl})$ son matrices de tamaño $m \times n$ y $p \times q$ respectivamente, el producto de Kronecker $A \otimes B$ es la matriz $mp \times nq$ hecha de bloques de tamaño $p \times q$, donde el bloque (i, j) es $a_{ij}B$. Entonces, la matriz de Sylvester $S(k)$ de orden 2^k es el producto de Kronecker iterado de k copias de la matriz de Hadamard $\begin{pmatrix} + & + \\ + & - \end{pmatrix}$ de orden 2 [31].

2.6. Relación Transformada-Matriz

Podemos expresar ahora la Transformada de Hadamard en términos de las matrices de Sylvester-Hadamard $\mathrm{H_n}$, es decir,

$H(f) = fH_n$; donde $(-1)^{\mathrm{u}\cdot\mathrm{v}}$ es la entrada en la posición $(\mathrm{u},\mathrm{v}) \in \mathrm{F}_2^n \times \mathrm{F}_2^n$, en la matriz $\mathrm{H_n}$. Consecuentemente,

$f = \frac{1}{2^n} H(f)H_n$ ó $f(u) = \frac{1}{2^n}\sum_{v\in \mathrm{F}_2^n}(-1)^{u\cdot v} H(f)(v)$

Similarmente, si ζ es una secuencia $(1,-1)$ en F_2^n, entonces la transformada de Hadamard es definida por

$$\mathrm{H}_\zeta = \zeta\, \mathrm{H_n}$$

Lema 2.2. Si la matriz de Sylvester-Hadamard$\mathrm{H_n}$ está dada por

$H_n = \begin{pmatrix} l_0 \\ l_1 \\ \vdots \\ l_{2^n-1} \end{pmatrix}$, donde l_i es una fila de H_n, entonces l_i como un vector es $l_i = ((-1)^{\alpha_i\cdot\alpha_0}, (-1)^{\alpha_i\cdot\alpha_1}, \dots, (-1)^{\alpha_i\cdot\alpha_{2^n-1}})$, donde α_i es la representación binaria de i, $0 \le i \le 2^n - 1$ escrito como un vector de longitud 2^n .

Demostración

Por inducción en n.

Para $n = 1$, tenemos $H_1 = \begin{bmatrix} + & + \\ + & - \end{bmatrix}$, $l_0 = (+\quad +)$, la sucesión de $\langle 0, x\rangle$ y $l_1 = (+\quad -)$, la sucesión de $\langle 1, x\rangle$ donde $x \in F_2$.

Supongamos que el lema es verdadero para $n = 1,2,\dots,k-1$

Puesto que $H_k = H_1 \otimes H_{k-1}$, donde $\otimes$ es el producto de Kronecker, cada fila de H_k puede ser expresada como $\delta \otimes l$ donde $\delta = (+\quad +)$ó$(+\quad -)$, y l es una fila de H_{k-1}. Asumiendo que l es la sucesión de una función, decimos $h(x) = \langle \alpha, x\rangle$ donde $\alpha, x \in \mathrm{F}_2^{k-1}$. De esta manera $\delta \otimes l$ es la sucesión de $\langle \beta, y\rangle$ donde $y \in \mathrm{F}_2^k$, $\beta = (0\quad \alpha)$ ó $(1\quad \alpha)$ acordando como $\delta = (+\quad +)$ó$(+\quad -)$. Así el lema es verdadero para $n = k$

Definiendo $l_{i+2^n} = -l_i$. Entonces como una consecuencia del lema anterior, se tiene tenemos que todas las secuencias afines se encuentran entre las filas de $\pm H_n$, es decir $l_0, \dots., l_{2^{n+1}-1}$.

Definición 2.2. El peso de Hamming definido en [37] de un vector $\mathrm{x} \in \mathrm{F}_2^n$, denotado por $\mathrm{wt(x)}$, es el número de 1s en el vector x.

Para una función booleana en F_2^n , sea $\Omega_\mathrm{f} = \{\mathrm{x} \in \mathrm{F}_2^n : f(\mathrm{x}) = 1\}$ el soporte de f. El peso de Hamming de una función f es el peso de Hamming de su tabla de verdad, es decir, la cardinalidad de $f^{-1}(1)$, o

equivalente $\mathrm{wt}(f) = |\Omega_f|$ tomado de [38]. La distancia de Hamming entre dos funciones $f, g: \mathrm{F}_2^n \to F_2)$, denotado por $d(f,g)$ es definida como $\mathrm{d}(f,\mathrm{g}) = \mathrm{wt}(f \oplus \mathrm{g})$.

La no linealidad de una función f, denotada por $\mathcal{N}_f$ y enunciada en [39-41], es definida como $\mathcal{N}_f = \min_{\phi \in \mathcal{A}_\mathrm{n}} \mathrm{d}(f, \phi)$, donde $\mathcal{A}_\mathrm{n}$ es la clase de todas las funciones afines en F_2^n. Una función de n variables es llamada balanceada si su peso es exactamente $2^{\mathrm{n}-1}$.

Lema 2.2. [42]El peso y la distancia de Hamming satisfacen las siguientes propiedades:

1. $\mathrm{wt}(\mathrm{x} \oplus \mathrm{y}) = \mathrm{wt}(\mathrm{x}) + \mathrm{wt}(\mathrm{y}) - 2\,\mathrm{wt}(\mathrm{x} * \mathrm{y})$;
2. $\mathrm{d}(f,g) = |\{\mathrm{x} \in \mathrm{F}_2^n : f(\mathrm{x}) \neq g(\mathrm{x})\}|$;
3. $d(f,g) + d(g,h) \geq d(f,h)$;
4. $d(f,g) = 2^n - \frac{1}{2}\sum_x \hat{f}(x) \cdot \hat{g}(x)$
5. $d(f,\bar{g}) = 2^n - d(f,g)$

2.7. Función de Autocorrelación

Definición 2.3.[43-45] La función de Autocorrelación$\hat{r}_f(a)$ es definida como

$$\hat{r}_f(a) = \sum_{x \in \mathrm{F}_2^n} \hat{f}(x) \cdot \hat{f}(x \oplus a).$$

En lo adelante escribiremos $\hat{\mathrm{r}}(\mathrm{a})$ si no hay ningún riesgo de confusión. Note que $\hat{\mathrm{r}}(0)$ es igual a 2^{n}. El valor de correlación entre dos funciones booleanas g y h es definido por[31]

$$\mathrm{c}(\mathrm{g},\mathrm{h}) = 1 - \frac{\mathrm{d}(\mathrm{g},\mathrm{h})}{2^{\mathrm{n}-1}}$$

Nosotros definimos la función de correlación cruzada entre $f, g: \mathrm{F}_2^n \to F_2$ por

$$c(\hat{f}, \hat{g})(y) = \sum_{x \in \mathrm{F}_2^n} \hat{f}(x) \cdot \hat{g}(x \oplus y)$$

Una relación muy importante afirma que la transformada inversa de la función polar es la función de autocorrelación.

Teorema 2.2.[46] Una función booleana en F_2^n satisface

$\mathrm{H}(\hat{\mathrm{r}})(\mathrm{h}) = \mathrm{H}\left(\hat{\mathrm{f}}\right)^2(\mathrm{h})$, para toda $\mathrm{h} \in \mathrm{F}_2^n$.

Ya que $\Pr\left(\hat{\mathrm{f}}(\mathrm{x}) \neq \hat{\mathrm{f}}(\mathrm{x} \oplus \mathrm{a})\right) = \frac{1}{2} - \frac{\hat{\mathrm{r}}(\mathrm{a})}{2^{\mathrm{n}+1}}$ para valores grandes de n, este teorema permite un cómputo eficiente de estas probabilidades, requiriendo $O(n\,2^n)$ operaciones, en vez de $O(2^{2n})$ para un cálculo directo.

La relación principal entre la transformada de Hadamard de f y $\hat{f}$ es mostrada en el siguiente lema.

Lema 2.3. Tenemos

$$H(\hat{f})(h) = -2H(f)(h) + 2^n\delta(h),$$

o

$$H(f)(h) = 2^{n-1}\delta(h) - \frac{1}{2}H(\hat{f})(h),$$

Donde $\delta(\mathrm{h}) = \begin{cases} 1, \mathrm{h} = 0 \\ 0, \ \mathrm{h} \neq 0 \end{cases}$

Demostración

A partir de la parte izquierda, obtenemos

$$H(\hat{f})(h) = \sum_{x\in \mathrm{F}_2^n} (-1)^{(f(x)\oplus(h\cdot x))} = \sum_{x\in \mathrm{F}_2^n} \big(1-2f(x)\big)(-1)^{(h\cdot x)}$$

$$= \sum_{x\in \mathrm{F}_2^n} (-1)^{(h\cdot x)} - 2\sum_{x\in \mathrm{F}_2^n} f(x)(-1)^{(h\cdot x)} = 2^n\delta(h) - 2H(f)(h)$$

Por el lema 2.1

2.8. Transformada de Hadamard en subespacio.

Uno puede encontrar una ecuación muy importante entre $H(f)$ y f restringido a un subespacio arbitrario de F_2^n, llamada Fórmula de Suma de Poisson, tomado de [47].

Teorema 2.3. Sea $f: \mathrm{F}_2^n \to F_2$ y $H(f)$ su transformada de Hadamard. Sea S un subespacio arbitrario de F_2^n y sea $S^{\perp}$ el dual de S, es decir, $S^{\perp} = \{x \in \mathrm{F}_2^n : x \cdot s = 0, \forall s \in S\}$.Entonces

$$\sum_{u\in S} H(f)(u) = 2^{dim(S)} \sum_{u\in S^{\perp}} f(u)$$

Demostración

Tenemos

$$\sum_{u\in S} H(f)(u) = \sum_{u\in S}\left(\sum_{v\in \mathrm{F}_2^n} f(v)(-1)^{u\cdot v}\right)$$

$$= \sum_{v\in \mathrm{F}_2^n} f(v)\left(\sum_{u\in S}(-1)^{u\cdot v}\right) = 2^{dim(S)} \sum_{v\in S^{\perp}} f(v)$$

Corolario 2.1. Para cualquier función booleana $f: \mathrm{F}_2^n \to F_2$

$$\sum_{u \le v} H(f)(u) = 2^{wt(v)} \sum_{u \le \bar{v}} f(u)\,,$$

Donde $\mathrm{u} \le \mathrm{v}$ significa que si $\mathrm{u_i} = 1$, entonces $\mathrm{v_i} = 1$, para cada $1 \le \mathrm{i} \le \mathrm{n}$.

2.9. Ecuación de Parseval

De la definición de la Transformada de Hadamard deducimos que $H(\hat{f})(u)$ es igual al número de ceros menos el número de unos en el vector binario $f \oplus l_u$ y entonces,

$H(\hat{f})(u) = 2^n - 2\, d(f, l_u(v))$

$d(f, l_u(v)) = \frac{1}{2}\,(2^n - H(\hat{f})(u))$ (2.3)

$$d(f, 1 \oplus l_u(v)) = \frac{1}{2}\,(2^n + H(\hat{f})(u))$$

Resumiremos esto en el siguiente teorema.

Teorema 2.4. La no linealidad de f es determinada por la transformada de Hadamard de f, es decir, $\mathcal{N}_f = 2^{n-1} - \frac{1}{2} max_{u \in \mathrm{F}_2^n} \left| H(\hat{f})(u) \right|$

Demostración

En una sección 2.6 se había definido la no linealidad como

$$\mathcal{N}_f = \min_{l_u(v) \in \mathcal{A}_n} d(f, l_u(v)) = \min_{l_u(v) \in \mathcal{A}_n} wt(f \oplus l_u(v))$$

y sustituyendo (2.3) en la definición

$\mathcal{N}_f = min_{l_u(v) \in \mathcal{A}_n} \left\{ \frac{1}{2}\,(2^n - H(\hat{f})(u)) \right\} = 2^{n-1} - \frac{1}{2} max_{u \in \mathrm{F}_2^n} \left| H(\hat{f})(u) \right|$. Por lo que queda demostrado.

Lema 2.4.

$$\sum_{u \in \mathrm{F}_2^n} H(\hat{f})(u)\, H(\hat{f})(u \oplus v) = \begin{cases} 2^{2n}, v = 0 \\ 0 \quad , v \neq 0 \end{cases}$$

Demostración

Tenemos

$$\sum_{u\in \mathrm{F}_2^n} H(\hat{f})(u)\, H(\hat{f})(u\oplus v) = \sum_{u\in \mathrm{F}_2^n}\sum_{w\in \mathrm{F}_2^n} (-1)^{u\cdot w}\hat{f}(w) \sum_{x\in \mathrm{F}_2^n} (-1)^{(u\oplus v)\cdot x}\,\hat{f}(x)$$

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

$$= 2^n \sum_{w\in \mathrm{F}_2^n} (-1)^{v\cdot w}\hat{f}(w)^2 = 2^n \sum_{w\in \mathrm{F}_2^n} (-1)^{v\cdot w} = 2^n 2^n = 2^{2n}$$

Donde $\hat{f}(w)^2 = 1$e y queda probado el lema.

Corolario 2.2. [48] (Ecuación de Parseval). Para una función booleana f en n variables, la siguiente ecuación es válida

$$\sum_{u\in \mathrm{F}_2^n} H(\hat{f})(u)^2 = 2^{2n}$$

Una consecuencia inmediata de este resultado es que $max_{u\in \mathrm{F}_2^n}\left|H(\hat{f})(u)\right| \geq 2^{n/2}$. Basado en esta observación se definen las funciones curvas o de Bent, las cuales son funciones booleanas de n variables de entradas tales que,

$H(\hat{f})(h) = 2^{n/2}$, $\forall\, h \in 0, \dots, 2^n - 1$(2.2)

Las relaciones dadas en (2.1) sirven para encontrar una función afín para f (en términos de la distancia de Hamming): $l_{u,a_0}(v) = a_0 \oplus u \cdot v$ donde $\left|H(\hat{f})(u)\right|$ es grande.

2.10. Distribuciones de Probabilidad

Sea $i(w)$ el índice $i-ésimo$ de la componente no nula de un vector $\mathrm{w} \in \mathrm{F}_2^n$. Suponiendo que las componentes de entrada $(x_1, \dots, x_n)$ de una función booleana $f(x)$ en F_2^n son variables aleatorias binarias independientes con distribución de probabilidad $\Pr(\mathrm{x_i} = 1) = \frac{1}{2} - \epsilon_\mathrm{i}$, así $\Pr(\mathrm{x_i} = 0) = \frac{1}{2} + \epsilon_\mathrm{i} i = 1,2, \dots, n$. La conexión entre la distribución de probabilidad de una función booleana f y la distribución de probabilidad de sus argumentos puede ser expresada en términos de la Transformada de Hadamard, bajo la condición de no uniformidad de la entrada [49].

Teorema 2.5. Si f es una función booleana arbitraria en F_2^n , entonces

$$\frac{1}{2} - Pr(f=1) = \frac{1}{2^{n+1}}\left[H(\hat{f})(0) + \sum_{s=1}^{n}\sum_{\substack{w\in \mathrm{F}_2^n \\ wt(w)=s}} 2^s H(\hat{f})(w)\epsilon_{1(w)}\dots\epsilon_{n(w)}\right]$$

Demostración

Sea $a = (a_1, \dots, a_n)$. Tenemos

$$Pr(f = 1) = \sum_{\substack{a \in \mathrm{F}_2^n \\ f(a)=1}} Pr(x_1 = a_1, \dots, x_n = a_n)$$

$$= \sum_{\substack{a \in \mathrm{F}_2^n \\ f(a)=1}} Pr(x_1 = a_1), \dots, \Pr(x_n = a_n)$$

$$= \sum_{\substack{a \in \mathrm{F}_2^n \\ f(a)=1}} \left(\frac{1}{2} + (-1)^{a_1}\epsilon_1\right) \dots \left(\frac{1}{2} + (-1)^{a_n}\epsilon_n\right)$$

$$= \frac{1}{2^n} wt(f) + \sum_{\substack{a \in \mathrm{F}_2^n \\ f(a)=1}} \sum_{s=1}^{n} \frac{1}{2^{n-s}} \sum_{\substack{w \in \mathrm{F}_2^n \\ wt(w)=s}} (-1)^{a \cdot w} \epsilon_{1(w)} \dots \epsilon_{n(w)}$$

$$= \frac{1}{2^n} wt(f) + \sum_{s=1}^{n} \frac{1}{2^{n-s}} \sum_{\substack{w \in \mathrm{F}_2^n \\ wt(w)=s}} \sum_{\substack{a \in \mathrm{F}_2^n \\ f(a)=1}} (-1)^{a \cdot w} \epsilon_{1(w)} \dots \epsilon_{n(w)}$$

Usando el Lema 2.2 y el hecho que $H(\hat{f})(0) = 2^n - 2wt(f)$, tenemos el corolario.

Corolario 2.4. Si f es una función booleana, entonces

$$\Delta_f(\epsilon) \coloneqq \max_{\substack{|\epsilon_i| \le \epsilon \\ 1 \le i \le n}} \left|\frac{1}{2} - Pr(f = 1)\right| = \frac{1}{2^{n+1}} \max_{x \in \mathrm{F}_2^n} \left| \sum_{w \in \mathrm{F}_2^n} (-1)^{x \cdot w} H(\hat{f})(w) (2\epsilon)^{wt(w)} \right|$$

2.11. Un nuevo problema: Entropía y matrices de Hadamard

Considere las n variables al azar con un juego de posibles resultados $i = 1, \dots, n$ que tiene la p_i de probabilidades $i = 1, \dots, n$. Nosotros tenemos la $\sum_{i=1}^{n} p_i = 1$. La(Shannon) entropía es

$$H\{p_i\} = \sum_{i=1}^{n} p_i \, ln \frac{1}{p_i} \qquad (2)$$

Esto tiene el cero de valor mínimo para el caso de certeza,

$$p_i = \begin{cases} 1 & i = j \ para \ algún \ j, \\ 0 & i \neq j. \end{cases}$$

Tiene el valor máximo $ln\, n$ cuando todos los resultados son igualmente probables,

$$p_i = \frac{1}{n}, \forall \, i = 1, \dots, n.$$

Nosotros definimos entropía de una matriz ortogonal ahora $O^{i}{}_{j}, i, j = 1,2, \ldots n$. Aquí $O^{i}{}_{j}$'s son los números reales con la restricción.

$$\sum_{i=1}^{n} O^{i}{}_{j} O^{i}{}_{k} = \delta_{jk} \qquad (3)$$

En particular, la i^{th} fila de la matriz es un vector normalizado para cada $i = 1, \ldots, n$. Nosotros podemos asociar las probabilidades $p_j^{(i)} = \left(O^{i}{}_{j}\right)^2$ con la i^{th} fila, como

$\sum_{j=1}^{n} p_j^{(i)} = 1$, para cada i. Nosotros definimos la entropía para matrices ortogonales como la suma de las entropías de cada fila:

$$H\{O^{i}{}_{j}\} = -\sum_{i,j=1}^{n} \left(O^{i}{}_{j}\right)^2 ln\left(O^{i}{}_{j}\right)^2.$$

El cero de valor mínimo se logra por la matriz de identidad $O^{i}{}_{j} = \delta_j^i$ y las matrices relacionadas obtenidas por intercambiando las filas o cambiando los signos de los elementos. La entropía de la fila i^{th}puede tener el valor máximo $\ln \mathrm{n}$ que se logra cuando cada elemento de la fila es $\pm \frac{1}{\sqrt{n}}$. Esto da el límite$H\{O^{i}{}_{j}\} \leq n\, ln\, n$. En general, la entropía de una matriz ortogonal no pueden lograr ya que esto limitado debido a la restricción de ortogonalidad (3) qué restringe $p_j^{(i)}$ para las filas diferentes. De hecho el límite sólo se obtiene por las matrices de Hadamard. Así nosotros tenemos un nuevo criterio para las matrices de Hadamard (apropiadamente normalizada): esas matrices ortogonales que saturan el límite para la entropía.

Note que la entropía es grande cuando cada elemento está cerca de $\pm \frac{1}{\sqrt{n}}$ como sea posible, es decir, a una diagonal principal. Así la condición de entropía máxima es similar a la condición determinante máxima del Hadamard. También, las matrices que corresponden a los máximos tienen los rasgos muy interesantes incluso para esas dimensiones n para que las matrices de Hadamard no existan.

2.12. Búsquedas heurísticas

Búsquedas heurísticas

Los métodos de búsqueda heurísticas (del griego *heuriskein*, que significa *encontrar*) están orientados a reducir la cantidad de búsqueda requerida para encontrar una solución. Cuando un problema es presentado como un árbol de búsqueda el enfoque heurístico intenta reducir el tamaño del árbol cortando nodos pocos prometedores. Estos métodos se llaman métodos fuertes porque ellos

son más poderosos que los estudiados hasta aquí al incorporar conocimiento heurístico o heurística. Hay una contradicción entre generalidad y potencia en el sentido que los métodos débiles son esencialmente aplicables universalmente mientras que los fuertes son menos universales en su aplicabilidad y el conocimiento o heurística usada en un problema dado puede no ser totalmente aplicable o ser inaplicable en otro dominio o tarea.

Feigenbaum y Feldman definen la heurística como sigue: "Una heurística es una regla para engañar, simplificar o para cualquier otra clase de ardid el cual limita drásticamente la búsqueda de soluciones en grandes espacios de estados". En esencia una heurística es simplemente un conjunto de reglas que evalúan la posibilidad de que una búsqueda va en la dirección correcta. Generalmente los métodos de búsqueda heurísticas se basan en maximizar o minimizar algunos aspectos del problema. Un ejemplo sencillo de heurística es el siguiente:

Un hombre se encuentra en una extensa llanura y tiene sed, en ese momento ha llegado a una pequeña elevación que es la única en esa región y se sube a ella. Desde la elevación el hombre observa el cuadro siguiente:

NORTE: vegetación verde y movimiento de animales

SUR: vegetación amarilla

ESTE: vegetación amarilla

OESTE: vegetación verde

Evidentemente la vegetación verde es un indicio de que hay humedad, luego es muy probable que exista agua en la superficie o subterránea. El movimiento de animales puede indicar que ellos se dirigen allí a beber, lo cual sugiere que el agua está en la superficie. Esta información le dice al hombre que debe dirigirse al norte, constituye una heurística.

La Heurística no garantiza que siempre se tome la dirección de la búsqueda correcta, por eso este enfoque no es óptimo sino suficientemente bueno. Frecuentemente son mejores los métodos heurísticos que los métodos de búsquedas a ciegas. Las desventajas y limitaciones principales de la heurística son:

- La flexibilidad inherente de los métodos heurísticos pueden conducir a errores o a manipulaciones fraudulentas.
- Ciertas heurísticas se pueden contradecir al aplicarse al mismo problema, lo cual genera confusión y hacen perder credibilidad a los métodos heurísticos.

- Soluciones óptimas no son identificadas. Las mejoras locales determinadas por las heurísticas pueden cortar el camino a soluciones mejores por la falta de una perspectiva global. La brecha entre la solución óptima y una generada por heurística puede ser grande.

El significado técnico de la palabra heurística ha variado en la historia de la Inteligencia Artificial. En 1957, George Polya en su libro Howtosolveit usó este término para referirse al estudio de métodos para descubrir e inventar técnicas de solución de problemas.

En otras ocasiones se ha usado como un término opuesto a algorítmico. Por ejemplo, Newell, Shaw y Simon plantearon en 1993 "Un proceso que puede resolver un problema dado, pero no ofrece garantía de hacerlo, es llamado una heurística para ese problema".

Actualmente, la heurística es más frecuentemente usada como un adjetivo para referirse a cualquier técnica que mejore la media del proceso de solución de problemas.

Según Shapiro, uno de los resultados empíricos de los últimos treinta años de la Inteligencia Artificial es que para muchos problemas la relación (balance) entre conocimiento, tiempo de cálculo y calidad de la solución es bastante favorable. Es decir, el uso de una pequeña cantidad de conocimiento específico del problema puede mejorar significativamente la calidad de la solución o el costo del proceso de búsqueda.

Funciones de evaluación heurística

La calidad de un nodo (estado, situación) del espacio de búsqueda se puede estimar de varias formas:

- Nivel de dificultad de resolver el sub-problema representado por el nodo.
- Calidad del conjunto de soluciones candidatas codificadas por el nodo.
- Cantidad de la información que se puede ganar expandiendo un nodo dado y la importancia de la información para guiar la búsqueda.

En todos estos casos la calidad el nodo se estima numéricamente por una función de evaluación heurística $f(n)$. Una función de evaluación heurística es una función que hace corresponder situaciones del problema con números. Es decir, da una medida conceptual de la distancia entre un estado dado y el estado objetivo. En general depende de la descripción de n (la descripción del objetivo, la información obtenida hasta ese punto de la búsqueda y cualquier conocimiento extra sobre el dominio del problema).

El proceso de construcción de funciones heurísticas bien puede ser considerado un proceso de descubrimiento, pues es muy difícil articular el mecanismo por el cual se llega a estas funciones. Sin embargo, se puede formular el siguiente paradigma general: *las heurísticas se descubren consultando modelos simplificados del dominio del problema.*

Estos valores son usados para determinar cuál operación ejecutar a continuación, típicamente seleccionando la operación que conduce a la situación con máxima o mínima evaluación. Un inconveniente de los métodos heurísticos es que en ocasiones no es posible conocer la calidad de la solución, es decir, cuán cerca está el óptimo (x^*) la solución heurística encontrada (x_{heu}); si por ejemplo, el problema es de maximización lo único que sabemos es que $x_{heu} \leq x^*$. Una forma simple de evaluar la calidad de una solución heurística es generar aleatoriamente varias soluciones y si son similares a la misma entonces cabría poner en duda la efectividad de la heurística.

Algunas consideraciones sobre las funciones heurísticas son:

a) La función debe dar un estimado útil y realista del mérito de un estado particular.

b) La evaluación de la función en general no debe requerir un gran cálculo en su aplicación. Si la evaluación de la función es computacionalmente compleja puede ser más eficiente hacer una búsqueda a ciegas en lugar de gastar recurso (tiempo y memoria) en el cálculo de la función.

c) Frecuentemente el costo de una solución exacta a un problema flexibilizado es una buena heurística para el problema original. Un problema flexibilizado es uno obtenido a partir del problema original simplificando las restricciones sobre los operadores. La idea es que el número exacto de movimientos requeridos para resolver un problema más simple puede ser fácil de calcular y puede servir como un estimado de la cantidad de movimientos necesitados para resolver el problema original.

d) Es siempre mejor usar una función heurística con valores más altos que otras, siempre que esta no este sobreestimada. Para un problema puede haber una colección de heurísticas admisibles $h_1, \dots, h_m$. Si una de ellas domina a las otras, es decir alcanza valores mayores para todos los nodos, se debe seleccionar esta. Si ninguna es dominante lo mejor es definir una heurística compuesta de la forma siguiente

   $h(n) = max(h_1(n), \dots, h_m(n))$.

   De esta forma h dominará todas las heurísticas individuales.

e) Otra forma de inventar una buena heurística es usar información estadística. Esto puede ser hecho realizando una búsqueda sobre una cantidad de problemas de entrenamiento, por ejemplo, en el juego de las ocho piezas cien configuraciones generadas aleatoriamente.

f) Frecuentemente es posible seleccionar rasgos de un estado que contribuyen a su evaluación heurística. La función de evaluación heurística puede ser construida como una combinación lineal de estos rasgos. Los rasgos pueden tener un peso que indique su importancia.

g) Otro tipo de modelo flexibilizado para un problema son los modelos analógicos. Aquí el modelo auxiliar flexibilizado extrae su potencia no de simplificar la estructura del problema a resolver sino de usar procesos de búsqueda que fueron empleados con éxito en problemas análogos. Esto se retoma posteriormente al estudiar el razonamiento por analogía.

Técnicas Heurísticas

Ascenso a Colina (Hill Climbing)

Es una variante del algoritmo de búsqueda de generación y prueba. Del procedimiento de prueba existe una realimentación que ayuda al generador a decidirse por cual dirección debe moverse en el espacio de búsqueda. En estos procesos se abandona la búsqueda si no existe un estado alternativo razonable al que se pueda mover.

Los algoritmos de ascenso a colina son típicamente locales, ya que deciden qué hacer, mirando únicamente a las consecuencias inmediatas de sus opciones. Puede que nunca lleguen a encontrar una solución, si son atrapados en estados que no son el objetivo, desde donde no se puede hallar mejores estados, por ejemplo:

1. Un máximo local, que es un estado mejor que sus vecinos pero no es mejor que otros que están algo más alejados.
2. Una meseta, es un espacio de búsqueda en el que todo un conjunto de estados vecinos tienen igual valor.
3. Un risco, que es un tipo especial de máximo local, imposible de atravesar con movimientos simples.

Hay algunas formas que pueden ayudar a solventar estos problemas, aunque no existe garantía:

1. Para evitar máximos locales, regresar a un estado anterior y explorar en una dirección diferente.
2. Para casos de mesetas, dar un salto grande en alguna dirección y tratar de encontrar una nueva sección del espacio de estado.
3. Para los riscos, aplicar dos o más reglas, antes de realizar una prueba del nuevo estado, esto equivale a moverse en varias direcciones a la vez.

Los algoritmos de ascenso a colina, a pesar de explorar sólo un paso adelante, al examinar el nuevo estado pueden incluir una cierta cantidad de información global codificada en la función objetivo o función heurística.

Recocido Simulado (Simulated Annealing)

Es una variación del ascenso a colina. Al inicio, este algoritmo, permite explorar una buena parte del espacio de estado, de tal forma que la solución final puede resultar insensible al estado inicial. En consecuencia, la probabilidad de quedar atrapado en un máximo local, en una meseta o en un risco, se hace mínima.

El procedimiento que se va a seguir para enfriar el sistema, se llama programa de recocido. Su forma óptima depende de cada tipo de problema y usualmente se lo descubre empíricamente.

El algoritmo para el recocido simulado, es ligeramente diferente del procedimiento simple de ascenso a colina. Las diferencias son:

Se debe respetar el programa de recocido.

Movimientos a estados peores que el actual, pueden ser aceptados.

Se debe mantener, a más del estado actual, el mejor estado encontrado hasta el momento. Así, si por alguna razón el estado final resulta peor que el mejor encontrado anteriormente, siempre será posible regresar a él.

Capitulo3. Búsqueda de funciones booleanas con fuertes propiedades criptográficas.

3.1 Propiedades criptográficas deseables en funciones booleanas

A continuación se enlistan varios de los principales criterios utilizados en la práctica profesional para diseñar cajas *S* con buenas propiedades criptográficas:

3.1.1 Balance

Esta propiedad es muy deseable para evitar ataques cripto-diferenciales tales como los introducidos por A. Shamir contra el algoritmo DES [21, 51-53].

Diremos que la función booleana f en V_n es balanceada si la tabla de verdad contiene tanto unos como ceros. Equivalentemente, f es balanceada si $wt(f) = |\Omega_f| = 2^{n-1}$. Usando el Corolario 2.4, podemos decir fácilmente que f es balanceada si y solo si $\Delta_f(\epsilon) = o(1)$

3.1.2 Alta no linealidad

Esta propiedad reduce el efecto de los ataques por criptoanálisis lineal. Como se discutió antes, la no linealidad de una función booleana puede ser calculada directamente de la transformada de Hadamard $\mathcal{N}_f = 2^{n-1} - \frac{1}{2} max_{u \in \mathrm{F}_2^n} |H(\hat{f})(u)|$

3.1.3 Autocorrelación

Este valor es proporcional al desbalance de todas las derivadas de primer orden de la función booleana. Valores pequeños son considerados como buenos mientras que un valor grande es considerado un símbolo de debilidad. Las funciones curvas, estudiadas en la subsección precedente, gozan de una autocorrelación mínima, por lo que optimizan esta propiedad.

La función de autocorrelación se define como $\hat{r}(s) = \sum_x f(x) f(x \oplus s)$ inversa. Es bien sabido [54] que la función de autocorrelación puede ser eficaz a partir de la transformada de Hadamard.

Teorema 3.1 (Wiener-Kintchine). Dada una función booleana$f(x)$tiene transformada de Hadamard $\widehat{H}(\omega)$ y la función de autocorrelación $\hat{r}(s)$.Para todo$\omega \in F_2^n$ es cierto que

$$\sum_{s \in F_2^n} \hat{r}(s)(-1)^{s \cdot \omega} = \left(\widehat{H}(\omega)\right)^2$$

3.1.4 Indicador absoluto

Indicador absoluto de una función booleana denotado por $M(f)$ está dado por $|r_{max}|$el máximo valor absoluto en $r_{\bar{f}}(s)$ (véase (12)). Se considera que una función booleana con un $M(f)$ pequeño es criptográficamente deseable. Nuevamente, las funciones curvas tienen una autocorrelación óptima, pues su indicador absoluto es cero [37, 55,56].

3.1.5 Efecto avalancha

Una función booleana $f(x)$ en n variables se dice que cumpleel Estricto Criterio Avalancha (*SAC* por sus siglas) si se cambia cualquiera de los n bits en la entrada x da lugar a la salida de la función de ser cambiado por exactamente la mitad de los 2^{n-1} vectores x con el bit de entrada cambiado. Por ejemplo, la siguiente función booleana con $n = 3$ se ve fácilmente que satisface la SAC.

Ejemplo 3.1 *(A-3 variables la función que cumple el SAC).*

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

El *SAC* es una característica útil para una función booleana en aplicaciones criptográficas porque al satisfacer el *SAC* significa que un ligero cambio en la entrada a la función lleva a un gran cambio en la salida (un efecto de avalancha), y de hecho un gran cambio de tipo uniforme (de ahí el nombre Estricto Criterio Avalancha). Este es un aspecto de hacer una función booleana cuya entrada es difícil deducir desde su salida, lo que veremos es esencial en un contexto de cifrado.

El *SAC* se definió por primera vez por Webster y Tavares [57] en un estudio de formas de buen diseño de S-*cajas.* Las funciones booleanas pueden ser vistos como piezas de la estructura de una S-*caja.* El diseño de S-cajas con funciones booleanas que satisfacen el *SAC* ha sido explorada en trabajos de Adams y Tavares [58],Kwangjo Kim [59] y Kim et al.[60] por mencionar algunos. Desde 1990, el *SAC* ha sido estudiado sobre todo en el contexto de las funciones booleanas, y que es el punto de vista que adoptamos aquí.

Es evidente que el lema 3.1 se desprende de nuestra definición del *SAC* y así proporciona una definición equivalente.

Lema 3.1: Una función Booleana $f: \mathrm{F}_2^n \to F_2$ satisface el SAC si y sólo si la función $f(x) \oplus f(x \oplus a)$ es balanceada para cada a en F_2^n con peso de Hamming 1.

Lema 3.1 proporciona una forma sencilla de verificar el*SAC* mediante cálculo, teniendo en cuenta los valores de salida de f.Para mayor brevedad, a partir de ahora a veces se dice que una función que cumple el *SAC* es una función *SAC,* o simplemente que la función es *SAC*.

Conteo de funciones SAC

Como con cualquier criterio de importancia criptográfica, es de interés contar las funciones que cumplen los SAC. Denotaremos S_n como el número de funciones booleanas en n variables las cuales son funciones SAC. Definimos $exp_2(x) = 2^x$, por lo que son exactamente $exp_2(2^n)$ las funciones booleanas en n variables. Una de las preguntas más simples que podemos hacer sobre el número S_n es el tamaño de la relación L_n definido por $L_n = 2^{-n} log_2 S_n$

Está claro que $L_n \leq 1$ y es natural conjeturar eso

$lim_{n\to\infty} L_n$ existe (3.1)

Supongamos que L *es* el límite en (3,1) Cusick [61] conjeturó que L existe, pero sólo demostró que $L_n \geq 1/4$ .En el mismo periódico, Cusick también conjeturó el resultado que se indica a continuación en el lema 3.3, que se comprobó de forma independiente por Cusick y Stanica [61] y Youssef y Tavares [62], utilizando diferentes métodos. La prueba que damos aquí aparece en los artículos de Youssef y Tavares.

Lema 3.2. Dado que cualquiera elección de los valores de$f(v_i), 0 \leq i \leq 2^{n-1}-1,$ para una función booleana en$n$variables, existe una opción de los restantes2^{n-1}los valores de$f(v_i)$, $2^{n-1} \leq i \leq 2^n-1$de modo que la función resultante$f(x)$satisface la*SAC*.

Identificamos la función booleana $f(x)$, con su tabla de verdad,

$$f(x) = \{f(v_0), f(v_1), \ldots, f(v_{2^n-1})\},$$

Donde $v_i = b(i)$. El caso $n = 1$ del lema es trivial, así que suponemos que $n \geq 2$. Dado que cualquier de elección de los valores de $f(v_i)$,$0 \leq i \leq 2^{n-1}-1$ nosotros definimos una función Booleana h_{n-1} en $n-1$ variables por

$$h_{n-1} = \{f(v_0), f(v_1), \ldots, f(v_{2^{n-1}-1})\}$$

g_{n-1} denota la función booleana $x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1} \oplus b$ en $n-1$ variables, donde$b$es fija e igual a 0 ó 1. Vamos a demostrar que si definimos $f_n(x)$

$$f_n(x) = \{h_{n-1}(x), h_{n-1}(x) \oplus g_{n-1}(x)\},$$

entonces la función $f_n(x)$ es una función SAC. Tenga en cuenta que esto demuestra un resultado ligeramente más fuerte que el lema, porque nos muestran que existen al menos dos opciones (para $b = 0\ y\ b = 1$) de $f_n(v_i)$, $2^{n-1} \leq i \leq 2^n - 1$ tal que $f_n(x)$ es SAC.

Sea a uno de los vectores en F_2^n con peso de Hamming 1. Sea a^* y v_i^* la composición de los $n-1$ bits menos significativos de a y v_i, respectivamente. Para la notación fácil que a veces denota por $C(u)$, en lugar de $\bar{u}$ el complemento de los bits u.Debido al Lema 3.1, los siguientes dos cálculos demuestran Lema 3.2.Caso 1. $a \neq (1,0,\dots,0)$

$$\sum_{i=0}^{2^n-1} f_n(v_i)\oplus f_n(v_i\oplus a) = \sum_{i=0}^{2^{n-1}-1} f_n(v_i)\oplus f_n(v_i\oplus a)\oplus \sum_{i=2^n-1}^{2^n-1} f_n(v_i)\oplus f_n(v_i\oplus a)$$

$$= \sum_{i=0}^{2^{n-1}-1} h_{n-1}({v_i}^*)\oplus h_{n-1}({v_i}^*\oplus a^*)\oplus \sum_{i=0}^{2^{n-1}-1} h_{n-1}({v_i}^*)\oplus h_{n-1}({v_i}^*\oplus a^*)\oplus g_{n-1}(v_i^*)\oplus g_{n-1}(v_i^*\oplus a^*)$$

$$= \sum_{i=0}^{2^{n-1}-1} h_{n-1}({v_i}^*)\oplus h_{n-1}({v_i}^*\oplus a^*)\oplus C\big(h_{n-1}({v_i}^*)\oplus h_{n-1}({v_i}^*\oplus a^*)\big) = 2^{n-1}$$

$g_{n-1}(v_i^*)\oplus g_{n-1}(v_i^*\oplus a^*) = 1$

que es válido para cualquier v_i^* y cualquier a^* de peso 1.

Caso 2. $a = (1,0\dots,0)$

$$\sum_{i=0}^{2^n-1} f_n(v_i)\oplus f_n(v_i\oplus a) = 2\sum_{i=0}^{2^{n-1}-1} f_n(v_i)\oplus f_n(v_i\oplus a) = 2\sum_{i=0}^{2^{n-1}-1} h_{n-1}({v_i}^*)\oplus h_{n-1}({v_i}^*)\oplus g_{n-1}(v_i^*)$$

$$= 2\sum_{i=0}^{2^{n-1}-1} g_{n-1}(v_i^*) = 2^{n-1}$$

Ahora hemos demostrado que $S_n = exp_2(2^{n-1}+1)$, así el corolario debajo sigue inmediatamente.

Corolario 3.1. Para cada$n$, $L_n \geq 1/2$.

Por un argumento mucho más complicado, D. Biss [63] logrado demostrar (3.1) y evaluó el límite 1.

Teorema 3.1. Tenemos $\lim_{n\to\infty} L_n = 1$.

Demostración

Ver [63]. La prueba consiste en un análisis detallado geométrico dentro de cubos de alta dimensión, entre otras cosas.

Conteo de funciones SAC balanceadas

Las funciones booleanas en aplicaciones criptográficas siempre tienen que ser balanceadas, o casi. Por lo tanto es de interés para ver si los resultados como los anteriores puede ser demostrado por el número de funciones booleanas SAC balanceadas en n variables. U_n denota este número. También estamos interesados en el tamaño de la relación B_n que se define por

$$B_n = 2^{-n} log_2 U_n$$

Como el anterior, es claro que $B_n \leq 1$ y es natural conjeturar que

$lim_{n\to\infty} B_n$ existe. (3.2)

No hay demostración de (3.2), pero lo siguiente es conocido a partir de [62].

Lema 3.3. Tenemos $lim\ inf_{n\to\infty} B_n \geq 1/2$.

Sea $x^* = (x_1, \dots, x_{n-1})$ y a^* ser cualquiera vector fijo de tamaño $n-1$ de peso impar. Al igual que en la demostración del lema 3.2, dado que cualquier elección de los valores de $f(v_i)$, $0 \leq i \leq 2^{n-1}-1$, nosotros definimos una función booleana h_{n-1} en $n-1$ variables por$h_{n-1} = \{f(v_0), f(v_1), \dots, f(v_{2^{n-1}-1})\}$, pero ahora se impone la condición adicional

$$\sum_{wt(v_i^*)\ impar} h_{n-1}(v_i^*) = 2^{n-3}$$

Desde b puede ser 0 o 1 en la definición de

$$g_{n-1} = x_1 \oplus x_2 \oplus \cdots \oplus x_{n-1} \oplus b$$

para cualquier vector a^* de peso impar tenemos $g_{n-1}(x^*) = g_{n-1}(x^* \oplus a^*) \oplus 1$,

si definimos la tabla de verdad de una función $F_n(x)$ en n variables

$$F_n(x) = \{h_{n-1}(x^*), h_{n-1}(x^* \oplus a^*) \oplus g_{n-1}(x^*)\}$$

(Similar a la definición de $f_n(x)$ en la prueba del lema 3.2), a continuación, por el Lema 3.2,$F_n(x)$ satisface la SAC. El cálculo siguiente se muestra que $F_n(x)$ es balanceada:

$$\begin{aligned}
\sum_{i=1}^{2^n-1} F_n(v_i) &= \sum_{i=1}^{2^{n-1}-1} h_{n-1}({v_i}^*) \oplus h_{n-1}({v_i}^* \oplus a^*) \oplus g_{n-1}(v_i^*) \\
&= \sum_{i=1}^{2^{n-1}-1} h_{n-1}({v_i}^*) \oplus h_{n-1}({v_i}^*) \oplus g_{n-1}(v_i^* \oplus a^*) \\
&= \sum_{i=1}^{2^{n-1}-1} h_{n-1}({v_i}^*) \oplus C\big(h_{n-1}({v_i}^*) \oplus g_{n-1}(v_i^*)\big) \\
&= \sum_{wt(v_i^*)par} \Big(h_{n-1}({v_i}^*) \oplus C\big(h_{n-1}({v_i}^*)\big)\Big) + 2 \sum_{wt(v_i^*)impar} h_{n-1}({v_i}^*) = 2^{n-2} + 2^{n-2} \\
&= 2^{n-1}
\end{aligned}$$

Así queda demostrado

$U_n \geq \binom{2^{n-2}}{2^{n-3}} exp_2(2^{n-2}+1)$.

Usando la fórmula de Stirling $n! \asymp (2\pi n)^{1/2}(n/e)^n$, nosotros encontramos

$$\binom{2^{n-2}}{2^{n-3}} exp_2(2^{n-2}+1) \asymp 2^{5/2}\pi^{-1/2} exp_2(2^{n-1}-n/2),$$

implica que $B_n \geq 1/2 - \epsilon(n)$, dónde $\epsilon(n) \to 0$ cuando $n \to \infty$. Esto da Lema 3.2.

Orden superior *SAC*

Ahora pasamos a una generalización de la *SAC* se definido en Forre [64], que llamó al *SAC* de orden superior.Una función booleana $f(x)$ en n variables se dice que cumplen los Estrictos Criterios Avalancha de orden k (SAC (k) para abreviar), si se fija cualquier k de los bits de n en la entrada x da lugar a una función booleana en $n-k$ variables restantes que satisface el SAC. Así, una función que cumple el *SAC* como se definió originalmente es una función $SAC\ (0)$. La definición de SAC (k) para funciones de n variable tiene sentido sólo para $0 \leq k \leq n-2$, ya que el *SAC* no está definido para funciones de 1-variable. Tenga en cuenta que la función en el ejemplo 3.1 se SAC (1).

Forre [64] no se dio cuenta que si una función es SAC (k) para $k > 0$, entonces también es SAC (j) para todo $j = 0,1,\ldots,k-1$. Esto fue señalado por Lloyd [65] y le damos la prueba en el siguiente lema.

Lema 3.4. Supongamos que$f(x)$es una función booleana en $\mathrm{n} > 2$ variables que satisface el SAC de orden$k, 1 \leq k \leq n-2$. Entonces$f(x)$también satisface el SAC de orden j para cualquier $j = 0,1,\ldots,k-1$.

Demostración. Probamos que si f satisface el *SAC* de orden k, entonces f también satisface el *SAC* de orden $k-1$. Entonces, nuestro lema será demostrado por inducción. Sea g una función en $n-k+1$ variables obtenidas por $k-1$ fijación de las variables en una f. Tenemos que demostrar que g es una función de *SAC,* por lo que por el Lema 3.2, es suficiente mostrar

$$S = \sum_{i=0}^{2^{n-k+1}-1} g(v_i) \oplus g(v_i \oplus a) = 2^{n-k} \qquad (3)$$

para todo $a \in F_2^{n-k+1}$ con peso de Hamming 1.Sin pérdida de generalidad, podemos tomar $a = (0,\ldots,0,1)$. Así, v_i y $v_i \oplus a$ tiene el mismo primero el pedazo, para que nosotros podemos hender la suma anterior S en dos sumas, primero dónde el primer pedazo de v_i es 0 y uno dónde que primero el pedazo es 1.Sea g_0 y g_1 denotan las funciones de g obtenidas mediante la fijación de la primera entrada de bit como 0 ó 1, respectivamente, y a^* denota el vector formado por los menos $n-k$ los pedazos significativos de a.

Entonces tenemos que

$$S = \sum_{i=0}^{2^{n-k}-1} g_0(v_i) \oplus g_0(v_i \oplus a^*) \oplus \sum_{i=0}^{2^{n-k}-1} g_1(v_i) \oplus g_1(v_i \oplus a^*)$$

Ambos g_0 y g_1 se obtienen a partir de f por k variables que se fijan, por lo que, por hipótesis, que son funciones *SAC*. Por consiguiente los dos de las sumas anteriores son 2^{n-k-1}.

Nuestro siguiente lema da el simple resultado de que en la prueba de si una función booleana f es SAC (k), podemos descartar los términos afines (si hay) en f.

Lema 3.5. Si una función booleana$f$ de$n$variables satisface$SAC(k)$para algúnk, $0 \leq k \leq n-2$ entonces también lo es $f \oplus g$ dónde g es cualquier función afin en n variables.

Demostración.

Esto se deduce inmediatamente del Lema 3.2.

Dos resultados fundamentales en funciones SAC (k) estuvieron a cargo de Preneel et al. [66]. El concepto clave que se utiliza en la prueba de estos resultados es la función de autocorrelación de una función booleana $f(x)$ en n variables, que se define para todo $a \in F_2^n$

$$r_f(a) = \sum_{i=0}^{2^n-1} f(v_i) \oplus f(v_i \oplus a) \; (3.4)$$

Por ejemplo, la función de autocorrelación de una función afín f es $r_f(a) = 2^n f(a)$, que es una de las constantes 0 o 2^n. Ahora podemos repetir el Lema 3.2 como:

Lema 3.6. Una función booleana$f$de$n$variables es SAC si y sólo si la función de autocorrelación$r_f(a)$es igual a2^{n-1}para todo $a \in F_2^n$ con peso de Hamming 1.

Necesitamos el siguiente lema para la prueba de los resultados de Preneel et al. [66].

Lema 3.7. Si f es una función booleana en $n > 2$ variables y $deg(f) = n$, entonces $r_f(a)$ no toma el valor de 2^{n-1} para cualquier$a \in F_2^n$.

Corolario 3.2. Si f es una función booleana en $n > 2$ variables y $deg(f) = n$, entonces f no satisface el *SAC*.

Demostración

Probamos que si $r_f(a) = 2^{n-1}$ para algún a, entonces el peso de Hamming $wt(f)$ es par. Esta es una contradicción, puesto que es evidente $deg(f) = n$ implica en peso $wt(f)$ es impar. Supongamos que $r_f(a) = 2^{n-1}$, a continuación,

$$wt(f) = \sum_{i=0}^{2^n-1} f(v_i) \equiv \sum_{i=0}^{2^n-1} f(v_i \oplus a) \equiv (1/2) \sum_{i=0}^{2^n-1} f(v_i) \oplus f(v_i \oplus a) \equiv r_f(a)/2 \equiv 2^{n-2} (mod\ 2).$$

Puesto que $n > 2$, tenemos $wt(f)$.

3.1.6 Grado algebraico

El grado algebraico de una función f, denotado como $deg(f)$, es el número de entradas más grande que aparece en cualquier producto de la forma normal algebraica. Esto es, $x_1 \oplus x_2$ tiene grado 1 (es decir, es lineal) mientras que $x_1 \oplus x_1x_2x_3$ tiene grado 3 [67-69].

Criterios de propagación

Una función booleana $f(x)$ en n variables se dice que satisface el criterio de propagación de grado k ($PC(k)$ para abreviar) si se cambia cualquier $i(1 \le i \le k)$ de los n bits de la entrada x resulta en la salida de la función ser cambiado por exactamente la mitad de los 2^n vectores x. También podemos decir simplemente que la función$f(x)$ es un $PC(k)$. Esto generaliza el concepto de SAC, que es claramente idéntica a la PC (1). Estos criterios fueron presentados por Preneel et al. [66] y aparecen en la tesis de Preneel [70, 71]. La función de tres variables en el ejemplo 3.1 cumple PC (2), pero no para PC (3).

Los criterios de propagación están estrechamente relacionados con las propiedades de la función de autocorrelación $r_f(a)$, porque tenemos

Lema 3.8. Una función booleana $f(x)$ en n variables satisface PC (k) si y sólo si todos los valores dados

$$r_f(a) = \sum_{x \in F_2^n} f(x) \oplus f(x \oplus a), \qquad 1 \le wt(a) \le k$$

de la función de autocorrelación son iguales a 2^{n-1}.

Demostración

De la definición de $r_f(a)$ tenemos

$$Pr\big(f(x) \ne f(x \oplus a)\big) = r_f(a)/2^n,$$

por lo que el lema se sigue inmediatamente de la definición de PC (k).

Podemos interpretar PC (k) en términos de teoría de la información. Una función f satisface PC (k) si y sólo si la información obtenida acerca de $f(x)$ dada $f(x \oplus a)$ para cualquier a con $1 \le wt(a) \le k$ es cero, es decir

$$I\big(f(x) \big| f(x \oplus a)\big) = 0, \qquad 1 \le wt(a) \le k \qquad (3.2)$$

Si$r_f(a)$ es simplemente la suma de todos los valores de la derivada direccional de $f(x) \oplus f(x \oplus a)$ cuando x se ejecuta a través de F_2^n. El siguiente lema reitera Lema 3.8 en términos de estas derivadas de dirección.

Lema 3.9. Una función booleana $f(x)$ en n variables satisface PC (k) si y sólo si todas las derivadas direccionales

$f_a(x) = f(x) \oplus f(x \oplus a), \qquad 1 \leq wt(a) \leq k,$

son funciones balanceadas.

Prueba.El lema sigue del Lema 3.8 y la definición de $PC\ (k)$. Alternativamente, podemos utilizar el hecho de que $f(x)$ satisface $PC\ (k)$ si y sólo si (3.2) se cumple.

Para funciones $PC\ (k)$, es un análogo del lema 3.8 para funciones SAC (k).

Lema 3.10. Si una función booleana f de n variables satisface PC (k) para algún k, $1 \leq k \leq n$ entonces también lo hace $f \oplus g$, donde g es una función afín en n variables.

Demostración

Esto se deduce inmediatamente del Lema 3.8.

Una función $f(x)$ en n variables PC(n) cumple si y sólo si $f(x)$ es no lineal perfecta, es decir, bent.

Orden superior PC (K)

La definición del SAC de orden superior que figura en la subsección 3.1.5 se puede generalizar para orden superior para PC (k).

Definición3.1. Una función booleana $f(x)$ en n variables se dice que satisface el criterio de propagación de grado k y orden m (PC (k) de orden m, para abreviar) si $k + m \leq n$, y si se fijan cualquier m de los n bits en la entrada x da lugar a una función booleana en los restantes $n - m$ variables que se ajuste a PC (k).

La condición de $k + m \leq n$ se impone porque cuando m bits son fijos, sólo hay $n - m$ variable bits a la izquierda que se puede cambiar, como la definición de PC (k) requiere. Si permitimos que m bits que se fije y, posteriormente, k bits a cambiar, esto tiene sentido, incluso si el $k + m \leq n$ condición es removida. Entonces podemos generalizar el concepto de SAC orden superior de una manera diferente.

Definición 3.2. Una función booleana $f(x)$ en n variables se dice que satisface el criterio de propagación prolongado de grado k y orden m (EPC (k) de orden m, para abreviar) si el conocimiento de m bits de x no da ninguna información acerca de $f(x) \oplus f(x \oplus a)$ para todo $\boldsymbol{a}$ con $1 \leq wt(a) \leq k$.

Estas definiciones fueron introducidas en[66]. Las funciones cuadráticas PC(k) satisface fueron estudiados en detalle en [72]. De ello se desprende de las definiciones y el Lema 3.9 que el PC(k), PC(k) de orden 0 y el EPC(k) de orden 0 todas significan lo mismo. Por supuesto que queremos una

función que satisfaga PC(k) o EPC(k) de orden m para satisfacer también el criterio correspondiente para todos los órdenes $< m$. Esto es cierto, como el siguiente lema.

Lema 3.11. Supongamos que $f(x)$ es una función booleana en n variables que se ajuste a PC (k) o EPC (k) de orden $m > 0$. Entonces $f(x)$ también satisface PC (k) o EPC (k), respectivamente, de orden j para cualquier $j < m$.

Demostración

La prueba del Lema 3.7, que da este resultado para $PC(1) = SAC$, se generaliza a probar el Lema 3.11.

Podemos expresar EPC (k) de orden $m > 0$ en términos de la inmunidad de correlación:

Lema 3.12. Una función booleana $f(x)$ en n variables satisface EPC (k) de orden $m > 0$ si y sólo si todas las derivadas direccionales

$f_a(x) = f(x) \oplus f(x \oplus a), \quad 1 \le wt(a) \le k,$

son balanceadas e inmunidad de correlación de orden m.

Demostración

El lema sigue inmediatamente de las definiciones de EPC (k) de orden m y la inmunidad de correlación de orden m, utilizando el lema 3.9.

Lema 3.12. Supongamos que $0 \le k \le n$. Una función booleana f de n variables satisface PC (k) si y sólo si para todo n-vector u con $wt(u) = k$ y cada n-vector v

$$\sum_{w \le \overline{u}} W(\hat{f})(w \oplus v)^2 = 2^{wt\,(\overline{u})+n}$$

La misma igualdad es válida para todos los u con peso $wt(u) \le k$.

Demostración

Por la definición de la Transformada de Hadamard, la parte izquierda de la ecuación en el lema es

$$\sum_{w \le \overline{u}} \left(\sum_{x \in F_2^n} (-1)^{f(x) \oplus x.(w \oplus v)} \right)^2 = \sum_{w \le \overline{u}} \sum_{x,y \in F_2^n} (-1)^{f(x) \oplus f(y) \oplus (x \oplus y).(w \oplus v)}$$

$$= \sum_{x,y \in F_2^n} (-1)^{f(x) \oplus f(y) \oplus (x \oplus y).v} \sum_{w \le \overline{u}} (-1)^{(x \oplus y).w} = 2^{wt(\overline{u})} \sum_{x,y \in F_2^n, x \oplus y \le u} (-1)^{f(x) \oplus f(y) \oplus (x \oplus y).v}$$

$$= 2^{wt(\overline{u})} \sum_{s \le u} (-1)^{s.v} \sum_{x \in F_2^n} (-1)^{f(x) \oplus f(x \oplus s)} = 2^{wt(\overline{u})+n}$$

La igualdad final sigue del Lema 3.8, lo que da

$$\sum_{x \in F_2^n} (-1)^{f(x) \oplus f(x \oplus s)} \text{ distinto de cero para } s \le u$$

3.1.7 Inmune correlación y funciones booleanas resistentes

Una función booleana $f(x)$ en n variables se dice que es inmune correlación de orden k, $1 \leq k \leq n$, si para cualquier subconjunto fijo de k variables la probabilidad de que, dado el valor de $f(x)$, las variables k tiene cualquier conjunto fijo de valores, es siempre 2^{-k}, no importa lo que la elección del conjunto fijo de valores de k, es decir, $f(x)$ es inmune correlación de orden k si sus valores son estadísticamente independientes de cualquier subconjunto k de variables de entrada. Si el subconjunto seleccionado de variables k es $\{x(i_1), \dots, x(i_k)\}$, entonces la definición de la inmunidad de correlación de orden k es equivalente a la condición de teoría de la información que la información obtenida acerca de los valores de $x(i_1), \dots, x(i_k)$ dado $f(x)$ es cero, es decir

$$I(x(i_1), \dots, x(i_k)|f(x)) = 0 \qquad (3.3)$$

en la notación habitual. Recordamos que la teoría de la información de base muestra que la función de información mutua $I(A|B)$ es simétrica, por lo que (3.3) es equivalente a

$$I\big(f(x)|x(i_1), \dots, x(i_k)\big) = 0.$$

Siegenthaler [73] define la inmunidad de correlación en primer lugar, y él afirma [73], que era el contenido intuitivo de (3.3), que lo llevó a su definición. Para dar un ejemplo, la siguiente función con $n = 3$ es fácil ver que la inmunidad de correlación de orden 1, pero no de orden 2.

Ejemplo 3.1 (una función 3-variable de la función, con inmunidad de correlación de orden 1).

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

Propiedades básicas de la inmunidad de correlación

Es útil para reunir varias condiciones equivalentes a la inmunidad de correlación (de orden 1) en el siguiente lema.

Lema 3.13. Una función $f(x)$en n variables es inmune correlación (orden 1) si y sólo si tienen alguna de las siguientes condiciones. (Condiciones equivalentes para la inmunidad de correlación de orden k, $1 \leq k \leq n$ tiene, pero se omiten por razones de brevedad).

a. Si $\Omega_f = \{x \in F_2^n : f(x) = 1\}$, entonces para cada $1 \leq i \leq n$, tenemos $|\{x \in \Omega_f : x_i = 1\}| = |\Omega_f|/2$.
b. Para cada $1 \leq i \leq n$, $f(x) \oplus x_i$ es una función balanceada.
c. Para cada $1 \leq i \leq n$, $Pr(x_i = 1|f(x) = 1) = 1/2 = Pr(x_i = 0|f(x) = 1)$.

d. Sea f_{0i} y f_{1i} denotan las funciones de $n-1$ variables obtenidas a partir de $f(x)$ mediante el establecimiento de $x_i = 0$ o 1, respectivamente. Entonces para cada $i = 1,2,\dots,n$, las funciones f_{0i} y f_{1i} tiene el mismo peso de Hamming.

e. Todas las Transformadas de Hadamard

$$H(\hat{f})(w) = \sum_{x \in F_2^n} (-1)^{f(x) \oplus x.w}, \qquad wt(w) = 1,$$

son iguales a cero.

f. Para cada$i = 1,2,\dots,n$, $Pr(f(x) = 1 | x_i = 1) = Pr(f(x) = 1 | x_i = 0) = wt(f)/2^n$.

Demostración

Todas las formas equivalentes seguir fácilmente de la definición de la inmunidad de correlación.

La extensión del lema 3.12 (e) a órdenes superiores de la inmunidad de correlación es lo suficientemente importante como para justificar un tratamiento distinto. Esto se hace en

Lema 3.13. Una función $f(x)$ en n variables es inmune de correlación de orden k, $1 \le k \le n$ si y sólo si todas las Transformadas de Hadamard

$$H(\hat{f})(w) = \sum_{x \in F_2^n} (-1)^{f(x) \oplus x.w}, \qquad 1 \le wt(w) \le k,$$

son iguales a cero.

Demostración

La demostración se basa en la observación de que la transformada de Hadamard $H(\hat{f})(w)$ es la correlación cruzada entre $f(x)$ y la función lineal $l_w(x) = w.x$. Definamos a y como el $k - vector$ por $y = (x(i_1), x(i_2), \dots, x(i_k))$,

donde $x(i_1), x(i_2), \dots, x(i_k)$ son las variables en $l_w(x)$. Luego nos fijamos en la transformada de Hadamard en las variables k de la probabilidad condicionada $Pr(y \mid z)$, donde z es un posiblevalor de $f(x)$. Vemos en la definición de la esperanza de que

$$\sum_y Pr(y|z)(-1)^{w.x} = E[(-1)^{w.x} | f(x) = z] = E[(-1)^{w.x}] = \sum_y Pr(y)(-1)^{w.x};$$

las dos sumas son iguales por nuestra hipótesis de la inmunidad de correlación. Así, $Pr(y|z)$ y $Pr(y)$ son idénticos, ya que su transformada de Hadamard son idénticas (cualquier función puede ser recuperado por la transformada inversa de Hadamard. Esto significa que la correlación cruzada entre $f(x)$ y $l_w(x)$ es cero, lo que da el lema.

Otra prueba del lema 3.13, basada en la combinatoria y el álgebra lineal en vez de la teoría de probabilidades, ha sido propuesta por Sarkar [74].

Vemos en la prueba del lema 3.13 que las funciones $f(x)$ y $l_w(x) = w.x$ son estadísticamente independientes si y sólo si la transformada de Hadamard $\mathrm{H}(\hat{f})(w) = 0$. De ello se deduce de la definición del valor de la correlación $c(f(x), l_w(x))$ que

$$c(f(x), l_w(x)) = 1 - 2Pr\left(f(x) \neq c(f(x), l_w(x))\right) = Pr\left(f(x) = c(f(x), l_w(x))\right) - Pr\left(f(x) \neq c(f(x), l_w(x))\right) = 2^{-n} \sum_{x \in F_2^n} (-1)^{f(x) \oplus w.x} = 2^{-n} H(\hat{f})(w). \qquad (4.2)$$

Así, el lema 3.13 dice que la inmunidad de correlación para el logro de $f(x)$ es lo mismo que recibir correlación cero de $f(x)$ con ciertas funciones lineales $l_w(x)$.

Es imposible garantizar que $f(x)$ no tienen una correlación distinta de cero con cualquier función lineal, es decir, no podemos lograr $c(f(x), l_w(x)) = 0$ para todo vector w, debido a la siguiente lema, que al parecer primero probado por Meier y Staffelbach [41].

Lema 3.14. Para cualquier función booleana $f(x)$, la correlación cuadrada total de $f(x)$ con el conjunto de todas las funciones lineal es igual a uno, es decir,

$$\sum_{w \in F_2^n} c(f(x), w.x)^2 = 1$$

Demostración

Por (3.12) tenemos

$$\sum_{w \in F_2^n} c(f(x), w.x)^2 = 2^{-2n} \sum_{w \in F_2^n} H(\hat{f})(w)^2,$$

y ahora el lema de la ecuación de Parseval.

Lema 3.14 da una limitación esencial de la cantidad de la inmunidad correlación que una función booleana puede tener. Si logramos obtener una correlación cero de $f(x)$ con varias funciones lineales, entonces necesariamente tendremos correlaciones cero con algunas otras funciones lineales. Esto está relacionado con el equilibrio entre el orden de la inmunidad de correlación y el grado de $f(x)$. Por ejemplo, si $f(x)$ en n variables ha pedido máximo de la inmunidad correlación $n-1$, entonces $f(x)$ debe ser $x_1 \oplus \ldots \oplus x_n$ o $x_1 \oplus \ldots \oplus x_n \oplus 1$, por lo que $f(x)$ tiene la máxima correlación posible a la suma de todas las variables.

Debido a Lema 3.14 y (3.12), es natural a buscar esas funciones booleanas $f(x)$ tal que el mayor valor posible de $|H(\hat{f})(h)|$ es tan pequeño como sea posible. Meier y Staffelbach [41] llamado a estas funciones perfectas no lineales.

3.2 Discusión de compromisos y conflictos en las propiedades de las funciones booleanas

De manera ingenua, uno podría plantearse buscar funciones booleanas que reúnan todas las propiedades criptográficas descritas en la subsección anterior. Así, podría ensayarse el vano intento de hallar funciones booleanas balanceadas, con máxima no linealidad, alto grado algebraico, alto orden de inmunidad de correlación y baja autocorrelación.

Sin embargo, es imposible que alguna función booleana pueda satisfacer al mismo tiempo todos esos criterios.

Quizás el ejemplo más socorrido para ilustrar esa realidad son las funciones curvas, son máximamente no lineales pero desbalanceadas. Si desistimos de las funciones curvas y nos concentramos en funciones balanceadas (esto es $H(0) = 0$), entonces, y como consecuencia del teorema de Parseval, algún otro coeficiente del espectro deberá necesariamente compensar ese faltante teniendo una magnitud mayor que $2^{n/2}$, lo cual reducirá la no linealidad de esa función. Otro conflicto más ocurre al intentar maximizar el orden de inmunidad, lo cual sólo puede llevarse a cabo en detrimento de la no linealidad [67]. Se conocen funciones booleanas curvas que exhiben máxima no linealidad y sin embargo tienen bajísimos grados algebraicos. Por otro lado, es posible hallar funciones con baja no linealidad pero con alto grado algebraico [58].

Debido a los conflictos existentes en las propiedades deseables para una función booleana, es necesario establecer compromisos. De esa manera, se ha ido adoptando más y más en la literatura especializada [58, 67-69, 75, 76] el perfil de una función booleana $f$balanceada, dado por la cuádrupla (n, m, d, nl), donde n denota el número de variables de entrada, m el orden de inmunidad, d el orden algebraico y nl la no linealidad de la función f.

3.3 Búsqueda de funciones booleanas por métodos heurísticos

Para poder realizar una búsqueda basada en técnicas heurísticas evolutivas, es indispensable contar con una *representación* que permita codificar las soluciones potenciales del problema en una población inicial de *individuos.* Enseguida es necesario definir operadores que, generación tras generación, alteren las características de los individuos, así que en cada generación, a los individuos con mejores características se les dé una mayor oportunidad de reproducirse, mejorando sus oportunidades de sobrevivir.

Los operadores que típicamente se utilizan en este tipo de heurísticas son la *mutación,* la *selección* y la *cruza.* En particular, para poder implementar el mecanismo de *selección,* resulta

indispensable contar con una función de *aptitud* que permita medir el desempeño de la solución representada en cada uno de los individuos de la población bajo análisis [54, 77-79].

En el caso de una búsqueda heurística de funciones booleanas, el problema de diseño más importante es decidir cuál será la función de aptitud que se utilizará para medir las bondades criptográficas de los individuos (funciones booleanas) que constituyen la población de cada generación [38, 77, 80]. En los últimos años se han propuesto diversas funciones de aptitud, de las cuales, en el resto de esta sección, se discutirán las siguientes tres: las tradicionales, las que se basan en inversión de espectro y en espacios restringidos.

Funciones de aptitud tradicionales

La abrumadora mayoría de los trabajos reportados antes del año 2000 [81, 82] enfilaban todos los cañones hacia la búsqueda de funciones altamente no lineales, sin reparar, ni poco ni mucho, en otras propiedades criptográficas. Así se propuso la medida de no linealidad de un individuo dado (esto es, alguna función booleana f) como su medida de aptitud:

$$Aptitud(f) = \frac{1}{2}(2^{\mathrm{n}} - |WH_{max}(f)|)$$

o visto como un problema de minimización, la función de aptitud se planteó también como:

$$costo(f) = |WH_{max}(f)| = \max_{\omega} \left|\widehat{H}(\omega)\right|$$

De manera similar, en los raros casos en que se fijó la baja autocorrelación como función objetivo, se utilizó una función de costo dada por:

$$costo(f) = \mathrm{AC(f)} = \max_{\mathrm{s}\neq 0} \left| \sum_{\mathrm{x}} \hat{f}(\mathrm{x})\hat{f}(\mathrm{x}\oplus\mathrm{s}) \right| = \max_{s\neq 0} |\hat{r}(s)| \ con\ s \in F_2^n$$

Funciones de aptitud basadas en inversión de espectro

Como se ha mencionado anteriormente, el espectro de Hadamard de una función booleana f permite evaluar rápidamente si los diferentes criterios de diseño han sido alcanzados o no. Es por ello que en años recientes se propuso desarrollar motores de búsqueda basados en las características que el espectro debiera tener en una buena función booleana. Esta estrategia realiza entonces una suerte de "ingeniería en reversa", en el sentido que la búsqueda se enfoca primero en diseñar el espectro con las características que se desean, para después, a través de la aplicación de la transformada inversa de Hadamard, hallar la función booleana a la que le corresponde tal espectro.

En concreto, supongamos que se cuenta con el espectro de Hadamard $\mathrm{H}(\omega) = \{H(0), H(1), \ldots, H(2^n - 1)\}$, de una función con perfil criptográfico (n, m, d, nl), esto es, el espectro correspondiente al de una función booleana balanceada de $n$variables de entrada con no

linealidad nl, grado algebraico d y orden de inmunidad m. Consideremos entonces el conjunto de espectros P dado por todas las posibles permutaciones del espectro original $H(\omega)$ tales que $P(\omega) = 0; 0 \leq H(\omega) \leq m$. Entonces cualquier espectro *G* incluido en el conjunto *P* disfruta de los mismos valores y propiedades criptográficos con los que cuenta el espectro original F.

Desafortunadamente, esta estrategia no garantiza que un espectro permutado Gen el conjunto Pcorresponderá a alguna función booleana legítima. En efecto, cuando se aplica la transformada inversa a G:

$$\hat{p}(x) = 2^{-n} \sum_{\omega} G(\omega)(-1)^{\omega x},$$

la función resultante $\hat{p}$tendrá, en general, coeficientes reales, en vez de tener todos sus coeficientes en {1,-1}, como corresponde a la representación polar de toda verdadera función booleana. Debido a ello, en [55]se propuso utilizar una asignación heurística para evaluar la *desviación* del espectro Ga un espectro legítimo. Se define la función booleana $\hat{b}$así que:

$$\hat{b}(x) = \begin{cases} +1 & si\ \hat{p}(x) > 0 \\ +1 & si\ \hat{p}(x) < 0 \\ +1\ o\ -1 & si\ \hat{p}(x) = 0 \end{cases}$$

Con lo que de manera natural surge como función de costo la ecuación que mide cuán lejos quedó la permutación espectral *G* de una verdadera función booleana, es decir [55]:

$$Costo(G) = \sum_{x=0}^{2^n-1} \left(\hat{p}(x) - \hat{b}(x)\right)^2 \qquad (16)$$

La función de costo en (16) tiene el defecto de hacer las evaluaciones en el dominio booleano abandonando el dominio de la frecuencia ω donde está definida la permutación espectral G. Es por ello que en [55] se definió una función de costo en el dominio de la frecuencia, fundamentada en el teorema de Titsworth, enunciado en la sección precedente (véase la ecuación (13)):

$$costo(G) = \sum_{s} \left(\left| \sum_{\omega \in F_2^n} G(\omega) G(\omega \oplus s) \right| \right) - 2^{2n} = 0 \qquad (17)$$

Con $s \in F_2^n$.

Utilizando la función de costo (17) se hicieron en[83] experimentos para hallar funciones booleanas con perfil criptográfico (7, 0, 6, 56), correspondiente a una función booleana de siete variables de entrada, balanceada, con orden de inmunidad 0, grado algebraico 6 y no linealidad 56.Se utilizó un algoritmo genético simple con porcentaje de mutación en el rango de$[1/100, 1/128]$ y porcentaje de cruza 0.7, obteniéndose resultados favorables en todas las corridas.

Tabla IV. Número de funciones booleanas de rotación simétrica en B_n

	2	3	4	S	6	7	8	9
B_n	2^4	2^8	2^{16}	2^{32}	2^{64}	2^{128}	2^{256}	2^{512}
$FBRS$	2^3	2^4	2^6	2^8	2^{14}	$\simeq 2^{19}$	$\simeq 2^{32}$	$\simeq 2^{57}$

Búsquedas en espacios restringidos

Aunque el método de búsqueda por inversión espectral ha dado en los últimos tres años excelentes resultados [55, 80, 84], sigue estando limitado por el hecho que el espacio de búsqueda $B_n$tiene un crecimiento doblemente exponencial con B_n. Por ello, en trabajos más recientes [81, 82,85] se ha utilizado un refinamiento del método de búsqueda por inversión espectral restringiendo el espacio de búsqueda al asociado a las funciones booleanas de rotación simétrica (FBRS).

Los FBRS son funciones booleanas que mantienen el mismo valor para todas las rotaciones cíclicas de sus entradas. Por ejemplo, para una FBRS de 5 variables de entrada se definen las siguientes 8 clases u orbitas [55, 81, 82,86-88] de rotación:

Órbita 1:$f(00000)$

Órbita 2:$f(00001) = f(00010) = f(00100) = f(01000) = f(10000)$

Órbita 3:$f(00011) = f(00110) = f(01100) = f(11000) = f(10001)$

Órbita 4:$f(00101) = f(01010) = f(10100) = f(01001) = f(10010)$

Órbita 5:$f(01011) = f(10110) = f(01101) = f(11010) = f(10101)$

Órbita 6:$f(00111) = f(01110) = f(11100) = f(11001) = f(10011)$

Órbita 7:$f(01111) = f(11110) = f(11101) = f(11011) = f(10111)$

Órbita 8:$f(11111)$

La tabla IV muestra el número de funciones booleanas de rotación simétrica en el universo total de B_n funciones booleanas para $n = 2,3, \dots ,9$.

Una propiedad muy útil de las FBRS es que el espectro de Hadamard toma el mismo valor para todos los elementos que pertenezcan a la misma órbita [85]. Además, a pesar de que el conjunto de FBRS representa sólo una pequeña fracción de todas las posibles funciones booleanas, el conjunto de funciones FBRS tiende a tener una muy rica no linealidad [81, 82].

Utilizando una búsqueda heurística del máximo gradiente restringido al subespacio de las funciones FBRS y empleando la técnica de inversión espectral, se reportó en diciembre de 2006 la siguiente función booleana de nueve variables con no linealidad 241 [82]:

977F 3FFA 0EFA AEC9 55F8 FACD CCA9 A083 7666 EBC0 FA88 E0B3 F4E0 8983 C845 915E 7F7C 2C29 FCCB A101 EA98 C085 E811 8B5E FE21 E911 8483 851E E195 2136 9716 76E9

En importante señalar que desde 1974 se había conjeturado que tal función podría existir, pero tuvieron que pasar más de 30 años para poder confirmar esa afirmación con evidencia experimental[82].

Conclusiones

- El presente trabajo aborda una nueva temática dentro de la Criptografía, la búsqueda de funciones booleanas criptográficamente deseables.
- Se estudió la teoría de las funciones booleanas y las propiedades criptográficamente deseables de las mismas.
- Se estudió y presentó de manera organizada la teoría matemática relacionada con la transformada de Hadamard y métodos heurísticos.
- Se explicaron las relaciones entre la transformada de Hadamard y las propiedades de las funciones booleanas.

Recomendaciones

El presente trabajo abre una temática de investigación dentro del Departamento de Matemática de la Facultad de Matemática Física y Computación de la Universidad Central Marta Abreu de Las Villas específicamente dentro del Seminario de Criptografía por lo que se recomienda:

1. Aplicar los métodos heurísticos para el estudio de funciones booleanas propiedades criptográficamente deseables.
2. Seguir incursionando en otros trabajos posteriores la aplicación de la Transformada de Hadamard ayuda a la búsqueda de estas funciones deseadas.
3. Implementar como asignatura optativa el estudio de las funciones booleanas dentro de la Carrera de Matemática.

Bibliografía

1. Matsui, M., *Linear Cryptanalysis Method for DES Cipher.* Advances in Cryptology - Eurocrypt '93, 1994. **volume 765**: p. pages 386–397.

2. Shamir, E.B.a.A., *Differential cryptanalysis of DES-like cryptosystems.* Lecture Notes in Computer Science, 1991. **volume 537**: p. 2–21.

3. Biham, E., *Observations on the relations between the bit-functions of manys-boxes.* Presentation at the 3rd NESSIE Conference, Nov 2002.

4. C. Carlet, J.S., and X.-M. Zhang, *A Construction of Resilient Functions with High Nonlinearity.* IEEE Transactions on Information Theory, 2003.

5. S. Chee, S.L., K. Kim, and D. Kim, *Correlation Immune Functions withControllable Nonlinearity.* ETRI Journal, December 1997: p. 389–401.

6. C. Carlet, J.S., and X.-M. Zhang, *Comments on "Generating andCounting Binary Bent Sequences".* IEEE Transactions on Information Theory, 1994.

7. Anubis, *Submission to the New European Schemes for Signatures.*Available athttp://cosic.esat.kuleuven.ac.be/nessie/workshop/submission/anubis.zip.

8. Carlet, C., *Partially-Bent Functions.* Advances in Cryptology - Crypto '92, 1993. **volume 740 of Lecture Notes in Computer Science**: p. pages 280–291.

9. Cusick, T.W., *Boolean Functions Satisfying a Higher Order Strict AvalancheCriterion.* Advances in Cryptology - Eurocrypt '93, 1994. **volume 765**: p. 102–117.

10. Carlet, C., *A Construction of Bent Functions.* Finite Fields and Applications (third conference), 1996. **London Mathematical Society, Lecture Series 233**: p. pages 47–58.

11. Carlet, C., *On cryptographic complexity of Boolean functions.* Proccedingsof the Sixth Conference on Finite Fields with Application to Coding Theory, 2002: p. 53-69.

12. MacWilliams, F.J.a.S., J.A, *The Theory of Error-CorrectingCodes.* 1997.

13. Kahn, D., *The Codebreakers, Scribner.* 1996.

14. Singh, S., *The code book.* 1999.

15. Diffie, W.a.H., M, *New directions in cryptography.* IEEE Transactionson Information Theory, 1976. **vol. IT-22**: p. 644-654.

16. Menezes, A.J., van Oorschot, P.C. and Vanstone,S, *Handbook of AppliedCryptography.* 1997.

17. *Data Encryption Standard.* FIPS Publ. 46-I, 1987.

18. Daemen, J.a.R., V, *The Design of Rijndael: The AES-AdvancedEncryption Standard.* Information Security and Cryptography, Texts andMonographs, 2001.

19. Rivest, R., Shamir, A. and Adleman L.N., *A method for obtainingdigital signature and public-key cryptosystems.* Communications of theACM, 1978: p. 120-126.

20. Koblitz, N., *Algebraic Aspects of Cryptography.* Algoritms and Computationsin Mathematics, 1998. **vol.3**.

21. Schneider, B., *Applied Cryptography.* 1996.

22. Stinson, D.R., *Cryptography: Theory and Practice.* 1995.

23. Lidl, R.a.N., H, *Finite Fields.* 1987.

24. Tapia-Recillas, H.a.V., G, *On the ZZ2k-Linear and QuaternaryCodes.*SIAM Journal on Discrete Mathematics, 2003. **Vol. 17, No.1**: p. pp. 103-113.

25. Rentería, C., Tapia-Recillas, H. and Velez, *Breve Introducción acódigos Detectores-Correctores de Errores.* Aportaciones Matemáticas,No.7, 1990.

26. Huffman, W.C.a.P., V, *Fundamentals of Error-Correcting Codes.*2003.

27. Hedayat, A., Wallis, W.D., *Hadamard matrices and their applications.* 1978: p. 1184–1238.

28. EW., W. *Hadamard matrix.* 2010; Available from: http://mathworld.wolfram.com/HadamardMatrix.html.

29. Yamada, J.S.a.M., *Hadamard matrices, sequences and block designs.*J. H. Dinitz and D. R. Stinson, 1992.

30. Marlon J. Luján Paredes, E.M.P.R., Klebes R. Arias Quispe, *IMPLEMENTACIÓN DE LA TRANSFORMADA BIDIMENSIONAL DE HADAMARDEN UN FPGA.*

31. Stanica, T.W.C.a.P., *Cryptographic Boolean Functions and Applications.*

32. Bernasconi A, C.B., Simon J., *On the Fourier analysis of Boolean functions.*1996: p. 1-24.

33. Henríquez, F.R., *De la búsqueda de funciones booleanas con buenas propiedades criptográficas.*

34. Institute for Studies in Theoretical Physics and Mathematics, I; Available from: http://math.ipm.ac.ir, 2010.

35. M. Hall, J., *Note on the Mathieu group M_{12}.* Arch. Math. 13 1962: p. 334-340.

36. Kantor, W.M., *Automorphism Groups of Hadamard Matrices.* 1967.

37. Carlet, C., D.K. Dalai, K.C. Gupta y S. Maitra, *Algebraic Immunity forCryptographically Significant Boolean Functions: Analysis and Construction.* IEEE Transactions on Information Theory 52(7), 2006.

38. Clark, J.A., *Evolving boolean functions satisfying multiple criteria.* In INDOCRYPT 2002, 2002: p. p. 246--259.

39. Sarkar, K.C.G.y.P., *Improved construction of nonlinear resilient s-boxes.* 8th International Conference on the Theory and Application of Cryptology and Information Security, 2002: p. 466-483.

40. Seberry J, Z.X.-M., Zheng Y., *Nonlinearity and propagation characteristics of balanced Boolean functions.* 1995.

41. Meier W, S.O., *Nonlinearity criteria for cryptographic functions.* 1990.

42. Carlet, C., *On the degree, nonlinearity, algebraic thickness, and nonnormality of Boolean functions, with developments on symmetric functions.* 2004.

43. Schneier, B., *Algorithms, and Source Code in Applied Cryptography. 2 ed.* 1996.

44. Carlet, C., *Recursive Lower Bounds on the Nonlinearity Profile of Boolean Functions and Their Applications.* 2008.

45. K., M., *Spectral analysis of Boolean functions under nonuniformity of arguments.* 2002.

46. Seberry J, Z.X.-M., *Highly nonlinear 0–1 balanced Boolean functions satisfying strict avalanche criterion.* 1993.

47. Beauchamp, K.G.a.L.D., *Walsh Functions and Their Applications.* 1979.

48. Sloane, N., *A library of Hadamard matrices.* Available from: http://www.research.att.com/~njas/hadamard/. 2010.

49. MacWilliams FJ, S.N., *The theory of error-correcting codes.* North-Holland Publishing Company, 1978.

50. S., W., *Cryptography with Cellular Automata.* 1986.

51. Rodríguez-Henríquez, F., N.A. Saqib, A. Díaz Pérez y Ç.K. Koç, *Cryptographic Algorithms on Reconfigurable Hardware.* noviembre 2006.

52. Singh, S., *The Code Book.*Disponible en:http://www.simonsingh.net/, junio 2000.

53. Singh, S., *Los códigos secretos.*2000.

54. Hernández-Luna, E., C.A. Coello Coello y A. Hernández-Aguirre, *On the use of a population-based particle swarm optimizer to design combinationallogic circuits.* junio 2004.

55. Clark, J.A., J.L. Jacob, S. Maitra y P. Stnic, *Almost boolean functions: Thedesign of boolean functions by spectral inversion.* 2004.

56. Gupta, K.C.y.P.S., *Improved construction of nonlinear resilient s-boxes.* En Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, 2002: p. 466–483.

57. Webster AF, T.S., *On the design of S-boxes.* 1986.

58. Adams CM, T.S., *The structured design of cryptographically good S-boxes.* 1990.

59. K, K., *Construction of DES-like S-boxes based on Boolean functions satisfying the SAC.* 1993.

60. Kim K, M.T., Imai H, *A recursive construction method of S-boxes satisfying strict avalanche criteria.* 1991.

61. TW, C., *Bounds on the number of functions satisfying the strict avalanche criterion.* 1996.

62. Youssef AM, T.S., *Comment on Bounds on the number of functions satisfying the strictavalanche criterion.* 1996.

63. DK, B., *A lower bound on the number of functions satisfying the strict avalanche criterion.* 1998.

64. R, F., *The strict avalanche criterion: spectral properties of Boolean functions and an extendeddefinition.* 1990.

65. S, L., *Counting functions satisfying a higher order strict avalanche criterion.* 1990.

66. Preneel B, V.L.W., Van Linden L, Govaerts R, Vandewalle J, *Propagation characteristics of Boolean functions.* 1991.

67. Adams CM, T.S., *Generating bent sequences.* 1992.

68. S, B., *On the relevance of the strict avalanche criterion.* 1990.

69. Bakhtiari S, S.-N.R., Pieprzyk JP, *Cryptographic hash functions: a survey.* 1995.

70. B, P., *Analysis and design of cryptographic hash functions.* 1993.

71. *Data cryptographic techniques – data integrity mechanism using a cryptographic check function employing a block cipher algorithm.* 1989.

72. Preneel B, G.R., Vandewalle J, *Boolean functions satisfying higher order propagationcriteria.* 1992.

73. T, S., *Correlation immunity of nonlinear combining functions for cryptographicapplications.* . IEEE Trans Inform Theory 1984.

74. P, S., *A note on the spectral characterization of Boolean functions.* 2000.

75. *Advanced Encryption Standard. Third conference.*http://csrc.nist.gov/encryption/aes/round2/conf3/AES3FeedbackForm-summary.pdf.

76. *Advanced Encryption Standard.*http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf;, 2001.

77. Hernández-Luna, E., *Documento de propuesta doctoral.* enero 2005.

78. Hernández-Luna, E., *Criterio de avalancha estricto en funciones booleanas.* mayo 2005.

79. Kennedy, J.y.R.C.E., *Swarm Intelligence.* 2001.

80. Clark, J.A., J.L. Jacob y S. Stepney, *The design of S-boxes by simulatedannealing.* 2004.

81. Kavut, S., S. Maitra, S. Sarkar y M.D. Yücel, *Enumeration of 9-VariableRotation Symmetric Boolean Functions Having Nonlinearity* Progress in Cryptology - INDOCRYPT 2006, 7thInternational Conference on Cryptology in India, Proceedings. Lecture Notesin Computer Science, 2006. **vol. 4329**: p. 266-279.

82. Kavut, S., S. Maitra y S. Sarkar, *There exist Boolean functions on n (odd)variables having nonlinearity > 2n−1 − 2n−1/2 if and only if n > 7.*Disponible en: http://eprint.iacr.org, 2006.

83. Cruz-Cortés, N., *Comunicación personal inédita.* diciembre 2006.

84. Clark, J.A., J. L. Jacob, S. Stepney, S. Maitra y W. Millan, *Evolving Booleanfunctions satisfying multiple criteria.*Proceedings of the Third InternationalConference on Cryptology, 2002: p. 246-259.

85. Saber, Z., M.F. Uddin y A. Youssef, *On the existence of (9, 3, 5, 240) resilient functions.* . IEEE Transactions on Information Theory 52(5), 2006.

86. Qu, P.J.y.C.X., *Fast hashing and rotation-symmetric functions.* Journal of Universal Computer Science, 1999.

87. Stănică, P., S. Maitra y J. Clark, *Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions.* 2004.

88. Stănică, P.y.S.M., *Rotation symmetric Boolean functions – Count andcryptographic properties.* Disponible en:http://www.isical.ac.in/~crg/tech_reports.html., December 2002.

89. W, M., *How to Improve the Non-linearity of Bijective S-boxes.* Lecture Notes in Computer Science, 1998.

90. Millan W., B.L., Carter G., Clark A., Dawson E, *Evolutionary Heuristics for Finding Cryptographically Strong S Boxes.* 1999.

91. Clark J. A., J.J.L., Stepney S., *The Design of S-Boxes by Simulated Annealing.* 2005.

92. Nedjah N., d.M.M.L., *Designing Substitution Boxes for Secure Ciphers.* International Journal Innovative Computing and Application 2007.