Differential Cryptanalysis and Boolean Functions

T-79.5501 Cryptology

Lecture 6
March 4, 2008

Kaisa Nyberg

Differential Cryptanalysis

Differential Cryptanalysis

- Presented by E. Biham and A. Shamir in 1989.
- A chosen plaintext attack on block ciphers.
- A difference in the plaintext is transferred over several nonlinear S-boxes to some output difference on the second last round.
- Makes use of differentials with large probabilities. Suppose that $f : \{0,1\}^n \to \{0,1\}^m$ is an S-box. Given $a' \in \{0,1\}^n$, $a' \neq 0$, we set:

$$N_D(a', b') = \#\{x \in \{0,1\}^n | f(x \oplus a') \oplus f(x) = b'\}$$

- Differential are chained similarily as linear approximations to form a differential characteristic. The probability of a characteristic is the product of the probabilities of the differentials it consists of.

The Table N_D of the S-box π_S

a'	b' 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

FIGURE 3.4
Difference distribution table: values of $N_D(a', b')$

Possible Output Differences b' for $a' = B = 1011$

x	$x \oplus a'$	y	$y*$	b'
0000	1011	1110	1100	**0010**
0001	1010	0100	0110	**0010**
0010	1001	1101	1010	0111
0011	1000	0001	0011	**0010**
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	**0010**
0111	1100	1000	0101	1101
1000	0011	0011	0001	**0010**
1001	0010	1010	1101	0111
1010	0001	0110	0100	**0010**
1011	0000	1100	1110	**0010**
1100	0111	0101	1000	1101
1101	0110	1001	1011	**0010**
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

$\pi_S(x \oplus 1011) \oplus \pi_S(x)$ takes only on 5 different values b'. One of them, $b' = 0010$ has probability $\frac{1}{2}$.

Iterative and Impossible Characteristic

- An iterative characteristic (resp. linear approximations) is such that its input difference (resp. linear combination at input) is equal to the output difference (resp. linear combination at the output). This property makes it possible to use them repeatedly in an iterated block cipher such as SPN (AES) or Feistel Network (DES).
- An impossible characteristic is an efficient tool for Feistel networks with bijective round function. If some candidate key leads to a situation which is known to never happen, then such a candidate is wrong.

Iterative Characteristics for Feistel Network

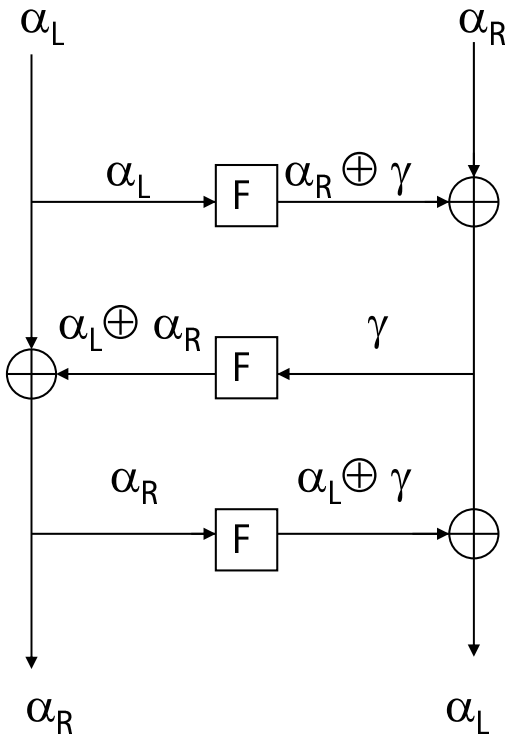


Figure 1: 3-round iterative characteristic

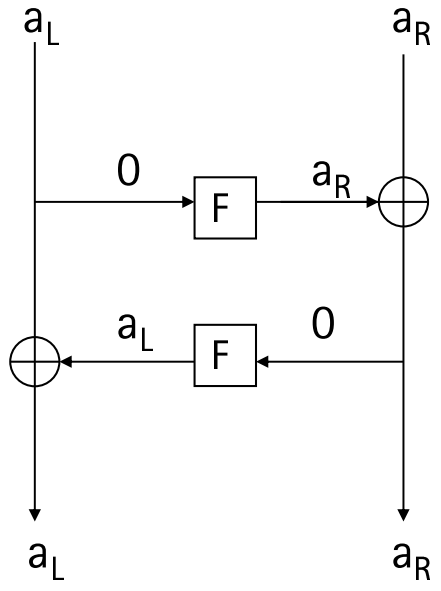


Figure 2: 2-round iterative linear relation

Impossible Differential Characteristic

F bijection

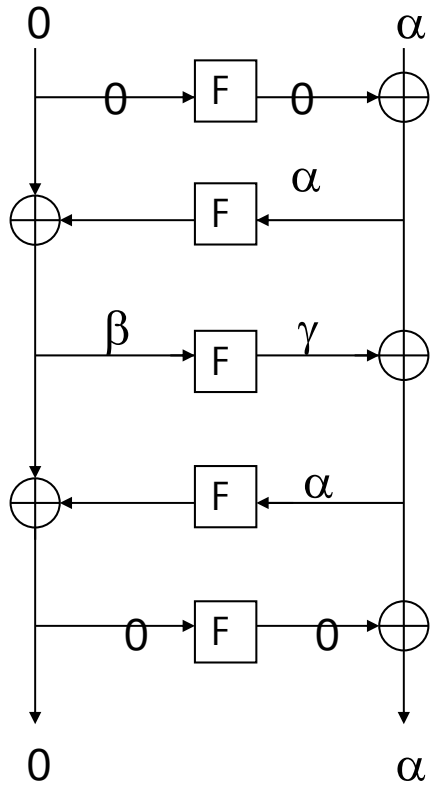
$\alpha \neq 0$

$\Downarrow$

$\beta \neq 0$

$\Downarrow$

$\gamma \neq 0$



Boolean Algebras and Rings

Background information

Boolean Algebra – Example

- **Example 1.** For a set E, we denote by $\mathcal{P}(E)$ the set algebra of E, that is,

$$\mathcal{P}(E) = \{x \,|\, x \subset E\}.$$

- For $x, y \in \mathcal{P}(E)$, let us define

addition	$x+y$	the union of x and y
with neutral element	0	the empty set
multiplication	xy	the intersection of x and y
with neutral element	1	the entire set E

Boolean Algebra – Operations

- $\mathcal{P}(E)$ equipped with these operations has the following properties:
  - (i) Addition is commutative and associative, and $x+0=x$, $1+x=1$, for all $x \in \mathcal{P}(E)$.
  - (ii) Multiplication is commutative and associative and $1x=x$, $x0=0$, for all $x \in \mathcal{P}(E)$.
  - (iii) The distributive law: $x(y+z)=xy+xz$, for all $x, y, z \in \mathcal{P}(E)$.
  - (iv) Each $x \in \mathcal{P}(E)$ has a unique complement $x' \in \mathcal{P}(E)$ such that $x+x'=1$ and $xx'=0$.
- **Definition 1.** Boolean algebra is a set $B=\{0,1,x,y,\ldots\}$ with three

operations:	addition	$x,y \mapsto x+y$	with the properties (i) - (iv)
	multiplication	$x,y \mapsto xy$	
	complementation	$x \mapsto x'$	

listed above.

Finite Boolean Rings

- Let B be a Boolean algebra. As defined above, there is no inverse with respect to addition. Define a new addition, the exclusive-or addition or xor-addition

$$x \oplus y = xy' + x'y, \text{ for } x, y \in B.$$

- **Fact 1.** xor-addition satisfies properties (i) - (iv), except that, instead of $1 + x = 1$, we have $1 \oplus x = x'$.
- **Fact 2.** xor-addition has neutral element 0 and inverses. Indeed, each $x \in B$ is its own inverse, since $x \oplus x = xx' + xx' = 0 + 0 = 0$.
- Let B be a Boolean algebra. Then B with xor-addition and its algebra-multiplication is a ring with unit 1.
- **Definition 2.** Boolean ring is a ring with the property that $xx = x$ for all elements x.

An Example and Boolean Polynomials

- **Example 2.** $E = \{a\}$ a set of one element. Then $\mathcal{P}(E) = \{0, 1\} = \mathbb{Z}_2$. Equipped with multiplication and or-addition (1+1 = 1), $\mathcal{P}(E)$ is a Boolean algebra. Equipped with multiplication and xor-addition ($1 \oplus 1 = 0$), $\mathcal{P}(E)$ is a Boolean ring.
- **Definition** Let B be a Boolean algebra. A Boolean polynomial in B is a string which results from a finite number of Boolean operations on a finite number of elements in B.

Representations of Boolean Polynomials

- **Example 3.** Boolean polynomials can be represented in different equivalent ways. Polynomials $x+yz$ and $(x+y)(x+z)$ are two different representations of the same polynomial. Similarily, $x(y+z)$ is the same as $xy+xz$ (distributivity law).
- Boolean algebra has a partial ordering defined as follows:

$$x \geq y \Leftrightarrow xy = y.$$

As usual, we denote $x > y$ in case $x \geq y$ and $x \neq y$. An element $x \in B$ is said to be a minimal element or atom, if $0 < x$ and there is no $y \in B$ such that $0 < y < x$. Similarily, $x \in B$ is said to be a maximal element, if $x < 1$, and there is no $y \in B$ such that $x < y < 1$. Clearly, complements of atoms are maximal elements and vice versa.

Disjunctive and Conjunctive Normal Forms

- Assume now that the Boolean algebra B is finite. Then for any given $x \in B$ the set of atoms contained by x is uniquely determined, and moreover, x has a unique representation as a sum of the atoms contained by x. Such a representation is called the *disjunctive normal form*.
- Similarily, any given $x \in B$ has a unique representation as the product of the maximal elements that are larger than or equal to x. This representation is the *conjunctive normal form*.

Algebraic Normal Form

- Let B be a Boolean algebra, and consider the associated Boolean ring. Then $\bigcup_n B[x_1, x_2, \ldots, x_n]$ is the set of all finite multivariate polynomials over B. Such a polynomial has a unique representation as an xor-sum of monomials of the form:

$$\bigoplus_{J \subset \{1,2,\ldots,n\}} a_J \prod_{j \in J} x_j$$

  where $a_J \in B$ are uniquely determined. This representation is called the *algebraic normal form*.

- Consider the Boolean algebra $B = \mathbb{Z}_2 = \{0, 1\}$. The the algebraic normal form of a Boolean polynomial of n indeterminates $x_1, \ldots, x_n$ over $B = \mathbb{Z}_2$ is

$$\begin{aligned} g(x_1, \ldots, x_n) \quad = \quad & a_0 \oplus a_1 x_1 \oplus \ldots \oplus a_n x_n \oplus a_{12} x_1 x_2 \oplus \ldots \\ & \ldots \oplus a_{(n-1)n} x_{n-1} x_n \oplus a_{123} x_1 x_2 x_3 \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \cdots x_n. \end{aligned}$$

  with coefficients $a_{i_1,\ldots,i_k} \in B = \mathbb{Z}_2$.

Boolean Functions

Algebraic Normal Form Algorithm

- Let us now consider a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$. Such a function is called a Boolean function of n variables. We can always associate with it a Boolean polynomial by deriving an algebraic normal form representation using the following algorithm:

- **ANF Algorithm.**

  1. Set $g(x_1, \ldots, x_n) = f(0, 0, \ldots .0)$
  2. For $k = 1$ to $2^n - 1$, do
  3. compute the binary representation of the integer k,
     $$k = b_1 + b_2 2 + b_3 2^2 + \cdots + b_n 2^{n-1}$$
  4. if $g(b_1, b_2, \ldots, b_n) \neq f(b_1, b_2, \ldots, b_n)$ then
     set $g(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \oplus \prod_{i=1}^{n} (x_i)^{b_i}$
  5. $\mathrm{ANF}(f) = g(x_1, \ldots, x_n)$

Algebraic Normal Form Algorithm – Example

- **Example 4.**

x_1	x_2	x_3	$f(x_1,x_2,x_3)$	k	$g(x_1,x_2,x_3)$
0	0	0	0		0
1	0	0	0	1	0
0	1	0	1	2	x_2
1	1	0	0	3	$x_2 \oplus x_1x_2$
0	0	1	1	4	$x_2 \oplus x_1x_2 \oplus x_3$
1	0	1	1	5	$x_2 \oplus x_1x_2 \oplus x_3$
0	1	1	0	6	$x_2 \oplus x_1x_2 \oplus x_3$
1	1	1	1	7	$x_2 \oplus x_1x_2 \oplus x_3$

Hamming Weight and Hamming Distance

- Let $x = (x_1, \ldots, x_m) \in \mathbb{Z}_2^m$. The *Hamming weight* of x is defined as

$$H_W(x) = |\{i \in \{1, 2, \ldots, m\} \,|\, x_i = 1\}|.$$

- For two vectors $x = (x_1, \ldots, x_m) \in \mathbb{Z}_2^m$ and $y = (y_1, \ldots, y_m) \in \mathbb{Z}_2^m$ the *Hamming distance* is defined as

$$d_H(x, y) = H_W(x \oplus y) = |\{i \in \{1, 2, \ldots, m\} \,|\, x_i \neq y_i\}|.$$

- Given two Boolean functions $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and $g : \mathbb{Z}_2^n \to \mathbb{Z}_2$ the *Hamming weight* of f is defined as

$$H_W(f) = |\{x \in \mathbb{Z}_2^n \,|\, f(x) = 1\}|,$$

and the *Hamming distance* between f and g is

$$d_H(f, g) = |\{x \in \mathbb{Z}_2^n \,|\, f(x) \neq g(x)\}|.$$

Balanced Boolean Functions

- A Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is *balanced* if $H_W(f) = 2^{n-1}$, which happens if and only if

$$|\{x \in \mathbb{Z}_2^n \mid f(x) = 1\}| = |\{x \in \mathbb{Z}_2^n \mid f(x) = 0\}|.$$

- **Example 5.** Let $f_{00} : \mathbb{Z}_2^4 \to \mathbb{Z}_2$ be the Boolean function defined as the first outputbit of the s-box S_1 of the DES, when the first and the last (sixth) input bits are set equal to zero. Then f_{00} has the following values

$$f_{00} = (1,0,1,0,0,1,1,1,0,1,0,1,0,1,0,0)$$

arranged in the *lexicographical order* with respect to the input (x_2, x_3, x_4, x_5). Clearly, f_{00} is balanced, that is, $H_W(f_{00}) = 8$. Further we see that

$$d_H(f_{00}, s_5) = 6, \text{ and } d_H(f_{00}, s_2) = 10,$$

where we have denoted by s_i the ith input bit to S_1 as a Boolean function of

Correlation and Linear Functions

- Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and $g : \mathbb{Z}_2^n \to \mathbb{Z}_2$ be two Boolean functions. The *correlation* between f and g is defined as

$$\begin{aligned} c(f,g) &= 2^{-n}(|\{x \in \mathbb{Z}_2^n \mid f(x) = g(x)\}| - |\{x \in \mathbb{Z}_2^n \mid f(x) \neq g(x)\}|) \\ &= 2^{-n}(2^n - 2|\{x \in \mathbb{Z}_2^n \mid f(x) \neq g(x)\}|) = 1 - 2^{1-n} d_H(f,g). \end{aligned}$$

- A Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is *linear* if it has an ANF of the form

$$f(x) = a \cdot x = a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n$$

for some $a = (a_1, a_2, \ldots, a_n) \in \mathbb{Z}_2^n$. Then f is just a linear combination of its input bits. In such a case we denote $f = L_a$. A Boolean function is *affine* if it has an ANF of the form $f(x) = a \cdot x \oplus 1$.

Nonlinearity of Boolean Functions

- *Nonlinearity* of a Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is defined as its minimum distance from the set consisting all affine and linear Boolean functions

$$\mathcal{N}(f) = \min_{L \text{ linear}} \{\min\{d_H(f, L), d_H(f, L \oplus 1)\}\}.$$

- **Example 5**(continued)
  From $d_H(f_{00}, s_5) = 6$ and $d_H(f_{00}, s_2) = 10$, it follows that the nonlinearity of f is at most 6. Further we see that

$$\begin{aligned} c(f_{00}, s_5) &= 1 - \frac{1}{8} \cdot 6 = \frac{1}{4}, \text{ and} \\ c(f_{00}, s_2) &= 1 - \frac{10}{8} = -\frac{1}{4}. \end{aligned}$$

Walsh-Hadamard Transform

- Recall the definition of *Walsh-Hadamard Transform*: Given an integer-valued function $f : \mathbb{Z}_2^n \to \mathbb{Z}$ the Walsh-Hadamard transform is defined as

$$F(w) = \sum_{x \in \mathbb{Z}_2^n} f(x)(-1)^{w \cdot x}, w \in \mathbb{Z}_2^n,$$

  where the sum is taken over integers.

- The Walsh-Hadamard Transform is its own inverse upto a constant multiplier. Given the Walsh-Hadamard transform $F(w)$, $w \in \mathbb{Z}_2^n$, of an integer valued function f we can compute the values of f as

$$f(x) \quad = \quad 2^{-n} \sum_{w \in \mathbb{Z}_2^n} F(w)(-1)^{w \cdot x} \text{ , for all } x \in \mathbb{Z}_2^n.$$

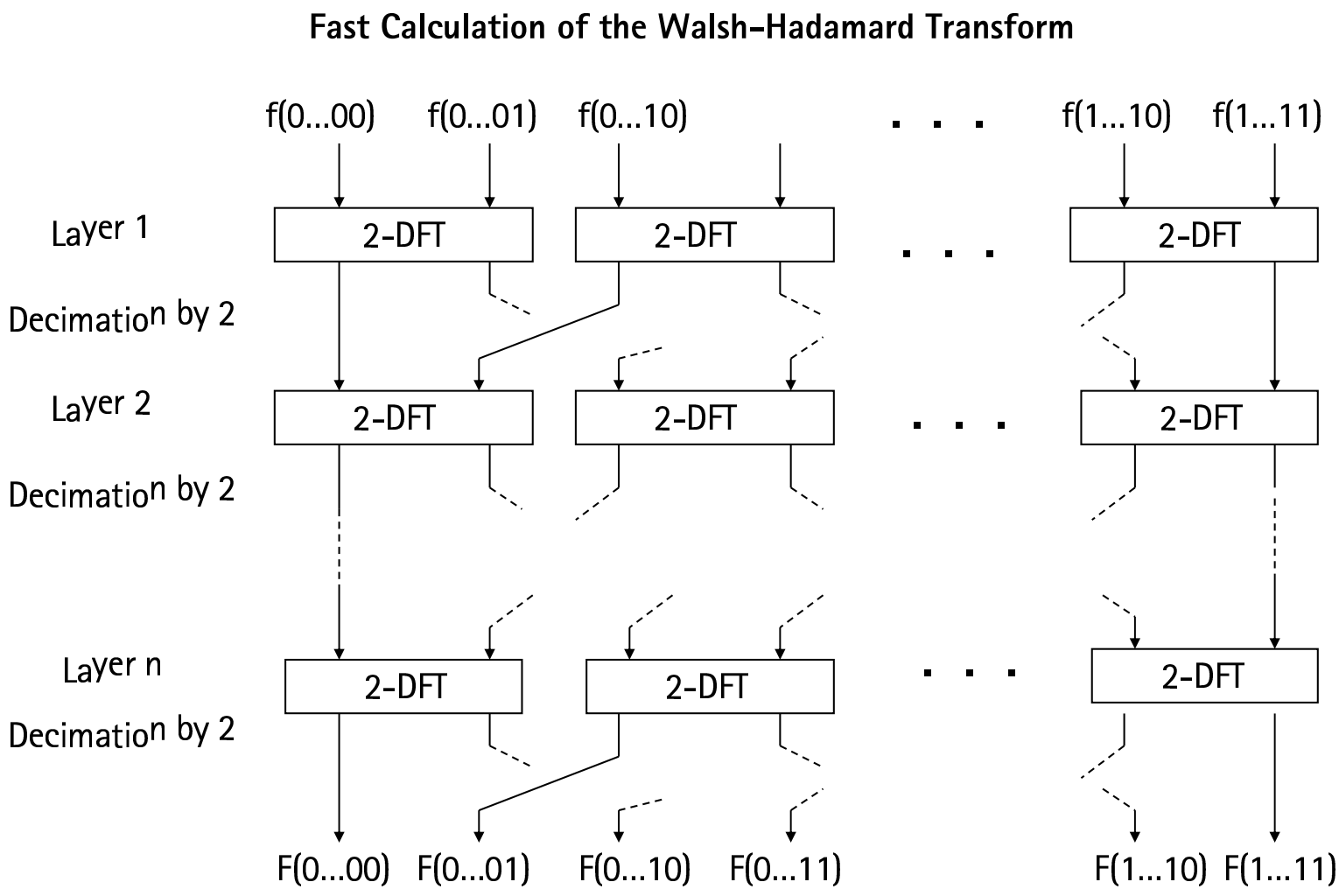
Fast Computation of Walsh-Hadamard Transform

- A fast algorithm for calculating the Walsh-Hadamard transform is depicted on the next slide. It takes n layers of 2^{n-1} parallel “2-DFT" operations followed by “decimation by 2". This is a permutation, which skips every second entry in the row, and after that takes the skipped elements without changing their order. If the number N of the entries is even, the permutation goes as follows:

$$(1,2,3,4,5,\ldots,N-1,N) \to (1,3,5,\ldots,N-1,2,4,\ldots,N).$$

- The [2-DFT] operation is the Discrete Fourier Transform of two inputs, defined as follows: [2-DFT]$(m,n) = (m+n, m-n)$, for integers m and n.
- Hence it takes $n2^n$ additions and subtractions to compute the Walsh-Hadamard Transform for a function of n Boolean variables.

Computation of the Walsh-Hadamard Transform



Two Transforms of Boolean Functions

- Given a Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ there are two possible ways of interpret it as an integer-valued function. The first approach is to take f as it is, and compute its Walsh-Hadamard transform as above

$$F(w) \quad = \quad 2^{-n} \sum_{x \in \mathbb{Z}_2^n} f(x)(-1)^{w \cdot x} \text{ , for all } w \in \mathbb{Z}_2^n.$$

- The second approach is to consider a related $\{-1, 1\}$-valued function $\hat{f}$ defined as follows:

$$\hat{f} : \mathbb{Z}_2^n \to \mathbb{Z},\ \hat{f}(x) = (-1)^{f(x)}.$$

- Applying the Walsh-Hadamard transform on $\hat{f}$, we get a transform $\hat{F} : \mathbb{Z}_2^n \to \mathbb{Z}$ defined as

$$\hat{F}(w) \quad = \quad \sum_{x \in \mathbb{Z}_2^n} \hat{f}(x)(-1)^{w \cdot x} = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) \oplus w \cdot x} \text{ , } w \in \mathbb{Z}_2^n.$$

The Walsh Transform

- The latter transform $\hat{F}$ is called the *Walsh transform* of the Boolean function f. There exists an easy conversion rule from Walsh-Hadamard Transform to Walsh Transform:

$$\hat{F}(w) = -2F(w) + 2^n \cdot \delta(w),$$

where δ is the *Kronecker symbol*:

$$\begin{aligned} \delta(0) &= 1, \\ \delta(w) &= 0, \text{ for } w \neq 0. \end{aligned}$$

- Hence, the Walsh transform of a Boolean function can be computed by computing first its Walsh-Hadamard Transform and then converting it to the Walsh Transform.

Correlation and the Walsh Transform

- Next we show that given a Boolean function its correlations with all linear functions can be computed simultaneously using the Walsh Transform. This is due to the fact that there is a close connection between the Walsh Transform and the correlations between f and linear functions. Indeed

$$\begin{aligned}\hat{F}(w) &= |\{x \in \mathbb{Z}_2^n \mid f(x) \oplus w \cdot x = 0\}| - |\{x \in \mathbb{Z}_2^n \mid f(x) \oplus w \cdot x = 1\}| \\ &= 2^n \cdot c(f, L_w)\end{aligned}$$

  where we used the notation for a linear function $L_w : L_w(x) = w \cdot x$.

- The values of the Walsh Transform are called the *spectral coefficients*, which are up to a constant, the same as the correlation coefficients between f and the linear functions.

Parseval's Theorem

- The next theorem states that linear approximations with non-zero correlation cannot be avoided. Every Boolean function contains some nonzero terms in its *Walsh spectrum* $\hat{F}(w)$, $w \in \mathbb{Z}_2^n$.
- **Theorem 1.** *Parseval's Theorem* Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ be a Boolean function. Then

$$\sum_{w \in \mathbb{Z}_2^n} c(f, L_w)^2 = 1 \, ,$$

or what is the same,

$$\sum_{w \in \mathbb{Z}_2^n} \hat{F}(w)^2 = 2^{2n}.$$

Parseval's Theorem – Proof

$$\begin{aligned}
\sum_{w\in\mathbb{Z}_2^n} \hat{F}(w)^2 &= \sum_{w\in\mathbb{Z}_2^n} \hat{F}(w)\hat{F}(w) \\
&= \sum_{w\in\mathbb{Z}_2^n} \Big(\sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)\oplus w\cdot x} \Big) \cdot \Big(\sum_{y\in\mathbb{Z}_2^n} (-1)^{f(y)\oplus w\cdot y} \Big) \\
&= \sum_{x,y\in\mathbb{Z}_2^n} (-1)^{f(x)\oplus f(y)} \sum_{w\in\mathbb{Z}_2^n} (-1)^{(x\oplus y)\cdot w} \\
&= 2^n \sum_{x\in\mathbb{Z}_2^n} (-1)^{f(x)\oplus f(x)} = 2^{2n},
\end{aligned}$$

where we used the following property for $u = x \oplus y$

$$\sum_{w\in\mathbb{Z}_2^n} (-1)^{u\cdot w} = \begin{cases} 2^n, \text{ for } u = 0 \\ 0, \text{ for } u \neq 0 \end{cases}$$

Example.

- **Example 6.** The standard hash-function SHA-1 makes use of the following two functions for combining three 32-bit blocks X_i, $i = 0, 1, 2$.

$$\begin{aligned} G(X_0, X_1, X_2) &= (X_0 \wedge X_1) \vee (\neg X_0 \wedge X_2) \\ T(X_0, X_1, X_2) &= (X_0 \wedge X_1) \vee (X_0 \wedge X_2) \vee (X_1 \wedge X_2) \end{aligned}$$

  where

  - $\wedge$ bitwise “and" multiplication
  - $\vee$ bitwise “or" addition
  - $\neg$ bitwise complementation

- The bitwise operations are the Boolean algebra operations. Let us now consider one bit component of G, and denote it by g. Using the Boolean algebra notation we have

$$g(x_0, x_1, x_2) = x_0 x_1 + x_0' x_2.$$

The Normal Forms of g

- The disjunctive normal form of g is

$$g(x_0, x_1, x_2) = x_0' x_1' x_2 + x_0' x_1 x_2 + x_0 x_1 x_2' + x_0 x_1 x_2.$$

- To determine the ANF of g we have two possibilities: either
  a) by direct algebraic manipulation,or
  b) make the value table of g and use the ANF algorithm.
- The representation of g in its ANF form is

$$g(x_0, x_1, x_2) = x_0 x_1 \oplus x_0 x_2 \oplus x_2.$$

- We calculate the distance between g and the linear function x_2, and get

$$d_H(g, x_2) = H_W(g \oplus x_2) = 2 \text{ and } c(g, x_2) = 1 - \frac{1}{4} \cdot 2 = \frac{1}{2}.$$

Correlations of g

- The correlations between g and the linear functions are calculated using the Walsh Transform:

	000	001	010	011	100	101	110	111
f:	0	1	0	1	0	0	1	1
	1	-1	1	-1	0	0	2	0
	1	1	0	2	-1	-1	0	0
	2	0	2	-2	-2	0	0	0
	2	2	-2	0	0	-2	0	0
	4	0	-2	-2	-2	2	0	0
F:	4	-2	-2	0	0	-2	2	0
$\hat{F}$:	0	4	4	0	0	4	-4	0

Differential Properties of Boolean Functions

- The differential properties of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ can be investigated by the *difference distribution table* DDT, also denoted by $N_D(a', b')$. The DDT is a $(2^n - 1) \times 2^m$ table, where the input difference indicates the row and the output difference indicates the column of the table. The entry in the row labeled by $a \in \mathbb{Z}_2^n$, $a \neq 0$, and in the column labeled by $b \in \mathbb{Z}_2^m$ is denoted by $\delta(a, b)$ and it is defined as

$$\delta(a, b) = \#\{x \in \mathbb{Z}_2^n \mid f(x) \oplus f(x \oplus a) = b\}.$$

- These values can be calculated as

$$\delta(a, b) = 2^{-(m+n)} \sum_{(c,w) \in \mathbb{Z}_2^{m+n}} \hat{F}_c(w)^2 (-1)^{(w,c)\cdot(a,b)}.$$

where $\hat{F}_c$ the Walsh Transform of the Boolean function $c \cdot f = c_1 f_1 \oplus \cdots \oplus c_m f_m$.

Perfect and Almost Perfect Nonlinear Functions

- Typically the entries in the DDT vary a lot, and such non-uniformity can be exploited in differential cryptanalysis.
- There exist S-boxes for which all values of $\delta(a,b)$, $a \neq 0$ are equal. Such functions are called perfect nonlinear, and they exist if and only if the number of input bits is even and, moreover, at least twice as large as the number of output bits.
- The S-boxes of the AES block cipher are *almost perfect nonlinear*, that is, $\delta(a,b) = 2$ or $\delta(a,b) = 0$, for all $a \neq 0$ and b.