Boolean Functions in Cryptography

Zülfükar Saygı

TOBB ETU

Winter 2016
Table of contents

1. Introduction

2. Representations

3. Criteria for Boolean functions

4. Bent Functions

5. Vectorial Boolean Functions
Outline

1. Introduction
2. Representations
3. Criteria for Boolean functions
4. Bent Functions
5. Vectorial Boolean Functions
Boolean functions play a central role in the design of many symmetric cryptosystems and in their security.

- **In Stream Ciphers (Combination Generator, Filter Generators,...)**
  - Combine the outputs of several linear feedback shift registers, or they filter (and combine) the contents of a single one.
  - Their output produces then the pseudo-random sequence which is used in a Vernam-like cipher (i.e., which is bitwise added to the plaintext to produce the ciphertext).

- **In Block Ciphers**
  - The S-boxes are designed by appropriate composition of nonlinear Boolean functions.
A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state.

LFSRs can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence.

**Motivation**
A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state.

LFSRs can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence.

long periods, and very uniformly distributed output streams.
Motivation

- A linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state.
- LFSRs can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence.
- Long periods, and very uniformly distributed output streams.
  - But LFSR is a linear system,
  - Berlekamp-Massey algorithm: finds the shortest LFSR for a given binary output sequence
    - Idea: Solve the set of linear equations
Three general methods are employed to reduce this problem in LFSR-based stream ciphers:
- Non-linear combination of several bits from the LFSR state;
- Non-linear combination of the output bits of two or more LFSRs.
- Irregular clocking of the LFSR (as in the alternating step generator).

Usage of NFSR.
Motivation

Combiner model:

\[
\begin{align*}
LFSR_1 & \quad x_1 \\
LFSR_2 & \quad x_2 \\
\vdots & \\
LFSR_n & \quad x_n \\
\end{align*}
\]

\[
f \quad \text{keystream } s_i
\]
Motivation

Filter model

LFSR

\( f \)

keystream \( s_i \)
Motivation

- DES Cipher
Motivation

- **DES S-box: S5**

<table>
<thead>
<tr>
<th>Outer bits</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0010</td>
<td>1100</td>
<td>0100</td>
<td>0001</td>
</tr>
<tr>
<td>0001</td>
<td>1110</td>
<td>1011</td>
<td>0010</td>
<td>1100</td>
</tr>
<tr>
<td>0010</td>
<td>0100</td>
<td>0010</td>
<td>0001</td>
<td>1011</td>
</tr>
<tr>
<td>0011</td>
<td>1011</td>
<td>1000</td>
<td>1100</td>
<td>0111</td>
</tr>
<tr>
<td>0100</td>
<td>1011</td>
<td>1100</td>
<td>0110</td>
<td>0001</td>
</tr>
<tr>
<td>0101</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>0110</td>
<td>1111</td>
<td>1101</td>
<td>0010</td>
<td>1100</td>
</tr>
<tr>
<td>0111</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1000</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1001</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1010</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1011</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1100</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
<tr>
<td>1101</td>
<td>1111</td>
<td>0000</td>
<td>1110</td>
<td>1101</td>
</tr>
</tbody>
</table>

- Given a 6-bit input, the 4-bit output is found by selecting the row using the outer two bits (the first and last bits), and the column using the inner four bits.
- For example, an input “011011” has outer bits “01” and inner bits “1101”; the corresponding output would be “1001”.
The Advanced Encryption Standard (AES), also known as Rijndael, is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.
Some Notations

- $\mathbb{F}_2 = \{0, 1\}$ finite field having 2 elements.
- $\mathbb{F}_{2^n}$ finite field having $2^n$ elements.
- $\mathbb{F}_2^n$ $n$-dimensional vector space over $\mathbb{F}_2$.
- $BF(n)$ the set of all Boolean functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$.
- $w_H(f) = w(f)$ Hamming weight of $f$,
  - $w(f) = |\{x \in \mathbb{F}_2^n : f(x) \neq 0\}|$.
- $d(f, g)$ distance between $f$ and $g$,
  - $d(f, g) = |\{x \in \mathbb{F}_2^n : f(x) \neq g(x)\}| = w(f + g)$.
A Remark

- Boolean functions are currently used in cryptography have low numbers of variables.
- Determining and studying those Boolean functions satisfying the desired conditions is not feasible through an exhaustive computer investigation.
  - $|BF(n)| = 2^{2^n}$ and it is too large when $n \geq 6$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>BF(n)</td>
<td>$</td>
<td>$2^{16}$</td>
<td>$2^{32}$</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>$\approx$</td>
<td>$6 \cdot 10^4$</td>
<td>$4 \cdot 10^9$</td>
<td>$10^{19}$</td>
<td>$10^{38}$</td>
<td>$10^{77}$</td>
</tr>
</tbody>
</table>
1. Introduction

2. Representations

3. Criteria for Boolean functions

4. Bent Functions

5. Vectorial Boolean Functions
Algebraic Normal Form (ANF)

- $n$-variable polynomial representation over $\mathbb{F}_2$

$$f(x) = \sum_{I \in P(N)} a_I \left( \prod_{i \in I} x_i \right) = \sum_{I \in P(N)} a_I x^I,$$

where $P(N)$ denotes the power set of $N = \{1, 2, \ldots, n\}$.

- This representation belongs to $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 + x_1, \ldots, x_n^2 + x_n)$.

- ANF is unique.

- There exists a simple divide-and-conquer butterfly algorithm to compute the ANF from the truth-table (or vice-versa).

- The degree of the ANF (algebraic degree)

$$deg(f) = \max\{|I| : a_I \neq 0\},$$

where $|I|$ is the size of $I$.

- $f$ is called affine function if $deg(f) = 1$. (also if $a_0 = 0$ then $f$ is called linear).
An example for ANF

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Note that a Boolean function $f$ has algebraic degree $n$ if and only if $\text{w}_H(f)$ is odd.
### An example for ANF

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\Rightarrow f = (1 + x_1)(1 + x_2)x_3 + x_1(1 + x_2)x_3 + x_1x_2x_3 = x_1x_2x_3 + x_2x_3 + x_3$

Note that a Boolean function $f$ has algebraic degree $n$ if and only if $w_H(f)$ is odd.
Representations

An example for ANF

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\Rightarrow f = (1 + x_1)(1 + x_2)x_3 + x_1(1 + x_2)x_3 + x_1x_2x_3 = x_1x_2x_3 + x_2x_3 + x_3$

- $\deg(f) = 3$
- Note that a Boolean function $f$ has algebraic degree $n$ if and only if $w_H(f)$ is odd.
Butterfly Algorithm

**Algorithm**

For every \( u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n \), the coefficient of \( x^u \) in the ANF of \( f \) equals

\[
\bigoplus_{(x_1, \ldots, x_{n-1}) \leq (u_1, \ldots, u_{n-1})} \left[ f(x_1, \ldots, x_{n-1}, 0) \right] \text{ if } u_n = 0 \text{ and } \\
\bigoplus_{(x_1, \ldots, x_{n-1}) \leq (u_1, \ldots, u_{n-1})} \left[ f(x_1, \ldots, x_{n-1}, 0) \oplus f(x_1, \ldots, x_{n-1}, 1) \right] \text{ if } u_n = 1.
\]

Hence if, in the truth-table of \( f \), the binary vectors are ordered in lexicographic order, then the table of the ANF equals the concatenation of the ANFs of the \((n - 1)\)-variable functions.
1. Write the truth-table of $f$, in which the binary vectors of length $n$ are in lexicographic order,

2. let $f_0$ and $f_1$ be the restrictions of $f$ to $\mathbb{F}_2^{n-1} \times \{0\}$ and $\mathbb{F}_2^{n-1} \times \{1\}$, respectively; replace the values of $f_1$ by those of $f_0 \oplus f_1$,

3. apply recursively step 2, separately to the functions now obtained in the places of $f_0$ and $f_1$.

- When the algorithm ends the global table gives the values of the ANF of $f$.
- The complexity of this algorithm is of $n \cdot 2^n$ XORs.
### ANF from Butterfly Algorithm

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
<th>$f(x_1, x_2, x_3) + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
</tr>
</tbody>
</table>
### ANF from Butterfly Algorithm

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
<th>1. $0 + 0 = 0$</th>
<th>2. $0 + 1 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
<td>$0 + 1 = 1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>$1 + 1 = 0$</td>
</tr>
</tbody>
</table>
### ANF from Butterfly Algorithm

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
<th>1.</th>
<th>2.</th>
<th>3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0 + 1 = 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0 + 0 = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0 + 0 = 0</td>
<td>0 + 1 = 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 + 0 = 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0 + 1 = 1</td>
<td>1</td>
<td>1 + 1 = 0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0 + 0 = 0</td>
<td>0</td>
<td>0 + 0 = 0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0 + 1 = 1</td>
<td>1 + 1 = 0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Representations

ANF from Butterfly Algorithm

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
<th>1. $0 + 0 = 0$</th>
<th>2. $0 + 1 = 1$</th>
<th>3. $1 + 1 = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
<td>$0 + 1 = 1$</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>1</td>
<td>1 + 1 = 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$0 + 0 = 0$</td>
<td>0 + 0 = 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$0 + 1 = 1$</td>
<td>$1 + 1 = 0$</td>
<td>$1 + 0 = 1$</td>
</tr>
</tbody>
</table>

$\Rightarrow f = x_1x_2x_3 + x_2x_3 + x_3$

- Complexity $n \cdot 2^n$ XOR operation.
Representations

Trace representation(s)

- Note that $\mathbb{F}_2^n \cong \mathbb{F}_{2^n}$.
- Every mapping from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^n}$ (Vectorial Boolean Functions) has a (unique) representation as a univariate polynomial

$$f(x) = \sum_{i=0}^{2^n-1} c_i x^i \quad c_i, x \in \mathbb{F}_{2^n}$$

over $\mathbb{F}_{2^n}$ in one variable and of degree at most $2^n - 1$.

- $f$ is Boolean if and only if $(f(x))^2 = f(x) \mod x^{2^n} + x$, that is, $c_0, c_{2^n-1} \in \mathbb{F}_2$ and $c_{2j} = (c_j)^2$ (where $2j$ is taken mod $2^n - 1$).
Trace representation(s)

- Trace function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$: $\text{Tr}_1^n(x) = x + x^2 + x^{2^2} + \cdots + x^{2^{n-1}}$
- Every Boolean function can be represented in the form

$$\text{Tr}_1^n \left( \sum_{i=0}^{2^n-1} c_i x^i \right) \quad c_i, x \in \mathbb{F}_{2^n},$$

but such a representation is not unique.
Trace representation(s)

- Trace function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$: $Tr_1^n(x) = x + x^2 + x^{2^2} + \cdots + x^{2^{n-1}}$
- Every Boolean function can be represented in the form

\[
Tr_1^n \left( \sum_{i=0}^{2^n-1} c_i x^i \right) \quad c_i, x \in \mathbb{F}_{2^n},
\]

but such a representation is not unique.
- $f$ can be represented uniquely in polynomial form as

\[
f(x) = \sum_{r \in R} Tr_1^{o(r)}(a_r x^r) + \epsilon(1 + x^{2^{n-1}}), \forall x \in \mathbb{F}_{2^n}, a_r \in \mathbb{F}_{2^{o(r)}}
\]

- $R$ is the set of cyclotomic coset leaders $r$,
- $o(r)$ is the size of the coset that contains $r$,
- $\epsilon$ is the modulo 2 value of $wt(f)$
1 Introduction

2 Representations

3 Criteria for Boolean functions

4 Bent Functions

5 Vectorial Boolean Functions
Criteria for cryptographic Boolean functions

Characteristics depend on the choice of the cryptosystem

- Balanced
  - output must be uniformly distributed for avoiding statistical dependence between the input and the output.

- High algebraic degree
  - Berlekamp-Massey attack for stream ciphers.
  - Higher differential attack for block ciphers.

- m-th order correlation-immune
  - output distribution probability is unaltered when any m of its input bits are kept constant.
  - m-resilient: $= m$-th order correlation-immune + balanced
  - if $m$ is small enough, a divide-and-conquer attack due to Siegenthaler (Correlation Attack for Stream Ciphers) and later improved (Fast Correlation Attack)
  - $m \leq n - 1 - \deg(f)$
The correlation between output of $f$ and linear functions should be small.

- Nonlinearity of $f$ ($N(f)$): the minimum Hamming distance between $f$ and all affine functions must be high.
- Walsh transform: $W_f(\omega) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{f(x) + Tr_1^n(\omega x)}$.

Then $N(f) = 2^{n-1} - \frac{1}{2} \max_{\omega \in \mathbb{F}_{2^n}} |W_f(\omega)|$.

$N(f) \leq 2^{n-1} - 2^{n/2 - 1}$. The functions achieving this upper bound are called bent.
The Propagation Criterion (PC)

- generalization of the Strict Avalanche Criterion (SAC)
- quantifies the level of diffusion put in a cipher by a Boolean function.
- It is more relevant to block ciphers.
- $f$ is $PC(l)$ if, $\forall a$ with $w_H(a) \leq l$
  
  \[ D_a f(x) = f(x) + f(x + a) \]
  
  is balanced

- SAC is $PC(1)$
- The bent functions are $PC(n)$. 
Criteria for cryptographic Boolean functions

- **Algebraic immunity** \((AI(f))\)
  - A function \(g\) such that \(fg = 0\) is called an *annihilator*.
  - \(AI(f)\) is the minimum degree of the nonzero annihilators of \(f\) or \(f + 1\).
  - \(AI(f) \leq \deg(f)\) and \(AI(f) \leq \lceil \frac{n}{2} \rceil\)
  - In practical situation, \(AI(f)\) must be greater than or equal to 7. Hence \(n \geq 13\) and in fact \(n \approx 20\).
  - A variant of algebraic attacks (fast algebraic attack) needs the existence of \(g \neq 0\) and \(h\) such that \(fg = h\), where only \(g\) has low degree and \(h\) has reasonable degree.
1. Introduction
2. Representations
3. Criteria for Boolean functions
4. Bent Functions
5. Vectorial Boolean Functions
A Boolean function $f$ on $\mathbb{F}_2^n$ ($n$ even) is called bent if its Hamming distance to the set of all $n$-variable affine functions (the nonlinearity of $f$) equals $2^{n-1} - 2^{n/2 - 1}$.

Equivalently, $f$ is bent iff $W_f(\omega)$ takes on values $\pm 2^{n/2}$ only.

Walsh transform: $W_f(\omega) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{f(x)} + Tr_1^n(\omega x)$.

Nonlinearity $N(f) = 2^{n-1} - \frac{1}{2} \max_{\omega \in \mathbb{F}_{2^n}} |W_f(\omega)|$. 
Any $n$-variable Boolean function ($n$ even) $f$ is bent iff

- the nonlinearity of $f$ is $2^{n-1} - 2^{n/2-1}$.
- for any nonzero vector $a$, $D_a f(x) = f(x) + f(x + a)$ is balanced.
- $f$ satisfies $PC(n)$.
- the $2^n \times 2^n$ matrix $H = \left[ (-1)^{f(x+y)} \right]_{x,y \in \mathbb{F}_2^n}$ is a Hadamard matrix (i.e. $H \times H^T = 2^n I_n$).
Properties of Bent Functions

- Any $n$-variable Boolean function ($n$ even) $f$ is bent iff
  - the nonlinearity of $f$ is $2^{n-1} - 2^{n/2-1}$.
  - for any nonzero vector $a$, $D_a f(x) = f(x) + f(x + a)$ is balanced.
  - $f$ satisfies $PC(n)$.
  - the $2^n \times 2^n$ matrix $H = \left[ (-1)^{f(x+y)} \right]_{x,y \in \mathbb{F}_2^n}$ is a Hadamard matrix (i.e. $H \times H^T = 2^n I_n$).

- **Open problem**: Characterize the bent functions of algebraic degrees at least 3 for $n \geq 10$.

- Note that the algebraic degree of any bent function on $\mathbb{F}_2^n$ ($n \geq 4$) is at most $n/2$. 
Example (Maiorana-McFarland Class)

\[ f(x, y) = x \cdot \pi(y) \oplus g(y) \]

where \( \mathbb{F}_2^n = \{(x, y)|x, y \in \mathbb{F}_2^{n/2}\} \), \( \pi \) is any permutation on \( \mathbb{F}_2^{n/2} \) and \( g \) is any Boolean function on \( \mathbb{F}_2^{n/2} \).
Example (Maiorana-McFarland Class)

\[ f(x, y) = x \cdot \pi(y) \oplus g(y) \]

where \( F_{2}^{n} = \{ (x, y) | x, y \in F_{2}^{n/2} \} \), \( \pi \) is any permutation on \( F_{2}^{n/2} \) and \( g \) is any Boolean function on \( F_{2}^{n/2} \).

- \( f = x_{1}x_{2} + x_{3}x_{4} + \cdots + x_{n-1}x_{n} \)
- \( f = x_{1}x_{2} + x_{3}x_{4} + \cdots + x_{n-1}x_{n} + L(x) \), where \( L(x) \) is any affine function.
Monomial Bent Functions

Definition

\( f \) is called a monomial function if \( f(x) = \text{Tr}_1^n(ax^s), \forall x \in \mathbb{F}_{2^n} \).

In order \( f \) to be bent, the following two conditions should be satisfied:

- \( \gcd(s, 2^n - 1) \neq 1 \).
- either \( \gcd(s, 2^{n/2} + 1) = 1 \) or \( \gcd(s, 2^{n/2} - 1) = 1 \).

Definition

If, for \( s > 0 \), \( \exists a \in \mathbb{F}_{2^n}^* \) such that \( \text{Tr}_1^n(ax^s) \) is bent, then \( s \) is called a bent exponent.
**Table:** All known bent exponents, $s, o(s) = n$

<table>
<thead>
<tr>
<th>$s$</th>
<th><strong>Condition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^i + 1$</td>
<td>$\frac{n}{\gcd(n,i)}$ even, $1 \leq i \leq \frac{n}{2}$</td>
</tr>
<tr>
<td>$r \cdot (2^{n/2} - 1)$</td>
<td>$\gcd(r, 2^{n/2} + 1) = 1$</td>
</tr>
<tr>
<td>$2^{2i} - 2^i + 1$</td>
<td>$\gcd(n, i) = 1$</td>
</tr>
<tr>
<td>$(2^{n/4} + 1)^2$</td>
<td>$n = 4r$, $r$ odd</td>
</tr>
<tr>
<td>$2^{n/3} + 2^{n/6} + 1$</td>
<td>$n = 0 \mod 6$</td>
</tr>
</tbody>
</table>

- For $o(s) < n$, the only bent Boolean functions are of the form $Tr_1^{n/2}(ax^{2^{n/2}+1})$ for some $a \in \mathbb{F}_{2^n}[5]$.
- Note: $o(s)$ is the size of the cyclotomic coset of 2 modulo $2^n - 1$ that contains $s$. 
Binomial Bent Functions

Definition

Functions of the form $Tr_1^n(ax^{s_1}) + Tr_1^k(bx^{s_2})$ are called binomial functions.

- **Binomial Bent Functions**
  - **Niho Bent Functions**[6]: $Tr_1^n(a_1x^{s_1} + a_2x^{s_2})$
    - $s_1 = (2^m - 1)^{\frac{1}{2}} + 1$ and $s_2 = (2^m - 1)^{3} + 1$;
    - $s_1 = (2^m - 1)^{\frac{1}{4}} + 1$ and $s_2 = (2^m - 1)^{\frac{1}{4}} + 1$ ($m$ odd);
    - $s_1 = (2^m - 1)^{\frac{1}{6}} + 1$ and $s_2 = (2^m - 1)^{\frac{1}{6}} + 1$ ($m$ even);
  - **Mesnager**[7] $Tr_1^n(ax^{(2^m-1)}) + Tr_1^2(bx^{\frac{2n-1}{3}})$
    - $m > 3$, $m$ odd
    - $K_m(a) = 4$ where Kloosterman sum $K_m(a) = \sum_{x \in \mathbb{F}_2^m} (-1)^{Tr_1^m(ax + \frac{1}{x})}$.
  - **Wang et al.**[8] $Tr_1^n(ax^{(2^m-1)}) + Tr_1^4(bx^{\frac{2n-1}{5}})$
    - $m \equiv 2$ mod 4
  - **Note:** A general characterization has not been achieved yet.
1 Introduction

2 Representations

3 Criteria for Boolean functions

4 Bent Functions

5 Vectorial Boolean Functions
**Definition**

Let \( n \) and \( m \) be two positive integers. The functions from \( \mathbb{F}_2^n \) to \( \mathbb{F}_2^m \) are called vectorial Boolean functions.

- \( F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \) where the Boolean functions \( f_i(x) \) on \( \mathbb{F}_2^n \) are called the coordinate functions of \( F \).
- In block ciphers such functions are also called S-boxes.
Let $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$.

- Walsh transform of $F$ maps any ordered pair $(u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ to the value at $u$ of the Walsh transform of the component $v \cdot F$.
  
  $$W_F(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x}.$$

- Nonlinearity of $F$ ($N(F)$): the minimum Hamming distance between all $v \cdot F$ and all affine functions.
  
  $$N(F) = 2^{n-1} - \frac{1}{2} \max_{(u, v) \in \mathbb{F}_2^n \times (\mathbb{F}_2^m \setminus \{0\})} |W_F(u, v)|.$$

- Note that $N(F) \leq 2^{n-1} - 2^{n/2-1}$. 

- Criteria for Vectorial Boolean Functions
A vectorial functions $F$ from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called bent if $N(F) = 2^{n-1} - 2^{n/2-1}$.

$F$ is bent iff all of its derivatives $D_aF(x) = F(x) + F(x + a)$, $a \in \mathbb{F}_2^n \setminus \{0\}$ are balanced (if it takes every value of $\mathbb{F}_2^m$ the same number $2^{n-m}$ of times).
A vectorial function $F$ from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called bent if $N(F) = 2^{n-1} - 2^{n/2-1}$.

- $F$ is bent iff all of its derivatives $D_aF(x) = F(x) + F(x + a)$, $a \in \mathbb{F}_2^n \setminus \{0\}$ are balanced (if it takes every value of $\mathbb{F}_2^m$ the same number $2^{n-m}$ of times).

- Bent functions exist only if $n$ is even and $m \leq n/2$. 

\[ N(F) \leq 2^{n-1} - 1 \left( \frac{2n}{2} \right) \]
A vectorial functions $F$ from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called bent if
\[ N(F) = 2^{n-1} - 2^{n/2-1}. \]

- $F$ is bent iff all of its derivatives $D_a F(x) = F(x) + F(x + a)$, $a \in \mathbb{F}_2^n \setminus \{0\}$ are balanced (if it takes every value of $\mathbb{F}_2^m$ the same number $2^{n-m}$ of times).

- Bent functions exist only if $n$ is even and $m \leq n/2$.

- The Sidelnikov-Chabaud-Vaudenay bound
  - Let $m \geq n - 1$. Then
  \[ N(F) \leq 2^{n-1} - \frac{1}{2} \sqrt{3 \times 2^n - 2 - 2 \frac{(2^n - 1)(2^{n-1} - 1)}{2^m - 1}}. \]
Special Vectorial Boolean functions ($n = m$ Case)

- $F$ is called almost bent (AB) if $N(F)$ achieves the Sidelnikov-Chabaud-Vaudenay bound with equality, that is, $N(F) = 2^{n-1} - 2^{(n-1)/2}$ ($n$ odd).

- If $F$ is AB, then the algebraic degree of $F$ is less than or equal to $(n + 1)/2$.

- $F$ is called almost perfect nonlinear (APN) if for every $a \in \mathbb{F}_2^n \setminus \{0\}$ and every $b \in \mathbb{F}_2^n$ the equation

$$F(x) + F(x + a) = b$$

has 0 or 2 solutions.
  - Every AB function is APN.
Open problem: Is there any APN permutations when \( n \) is even and \( n \geq 8 \)?

An example of APN permutation in 6 variables has been given by J. Dillon at the conference Fq 9 [9].

The existence of infinite classes of APN permutations when \( n \) is even also remains open.
Different characterization of APN Functions

- $F$ is APN if any of the following is satisfied:
  - $x \mapsto F(x) + F(x + a)$ is 2-to-1 for all $a \neq 0$.
  - For all distinct $a, b, c, d$
    \[ a + b + c + d = 0 \implies F(a) + F(b) + F(c) + F(d) \neq 0. \]
  - If $F(0) = 0$ the binary code with parity check matrix
    \[
    H_F = \begin{bmatrix}
    0 & \cdots & x & \cdots & 1 \\
    F(0) & \cdots & F(x) & \cdots & F(1)
    \end{bmatrix}
    \]
    is double-error-correcting (no fewer than 5 columns sum to 0).
Different characterization of APN Functions

- $F$ is APN if any of the following is satisfied:
  - $x \mapsto F(x) + F(x + a)$ is 2-to-1 for all $a \neq 0$.
  - For all distinct $a, b, c, d$
    
    $$a + b + c + d = 0 \implies F(a) + F(b) + F(c) + F(d) \neq 0.$$ 
  - If $F(0) = 0$ the binary code with parity check matrix
    
    $$H_F = \begin{bmatrix} 0 & \cdots & x & \cdots & 1 \\ F(0) & \cdots & F(x) & \cdots & F(1) \end{bmatrix}$$
    
    is double-error-correcting (no fewer than 5 columns sum to 0).

**Example**

The $F(x) = x^3$ on $\mathbb{F}_2^n$ is APN for all dimensions $n$. 
Differential cryptanalysis of block ciphers exploits the existence of 
\((a, b)\) such that \(F(x) + F(x + a) = b\) for many values of \(x\).

The differential uniformity of \(F\) is defined as

\[
\max_{a \in \mathbb{F}_2^n \setminus \{0\}, b \in \mathbb{F}_2^n} \left| \{x \mid F(x) + F(x + a) = b\} \right|
\]

Differential uniformity of APN functions are 2.
Differential uniformity of APN functions are 2.
Differential uniformity of $F(x) = x^{-1}$ is 2 if $n$ is odd and 4 if $n$ is even. Also note that $F$ is a permutation.
**AES S-Box**

- Differential uniformity of APN functions are 2.
- Differential uniformity of $F(x) = x^{-1}$ is 2 if $n$ is odd and 4 if $n$ is even. Also note that $F$ is a permutation.
- Note that the S-box of AES is $S(x) = Ax^{-1} + B$ over $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$, where

$$
A = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
1 \\
1 \\
0 \\
0
\end{bmatrix}
$$

* The value of 0x25 is 0x3F

C. Carlet, "Vectorial Boolean Functions for Cryptography".


S. Mesnager, Recent Results on Bent and Hyper-bent Functions and Their Link With Some Exponential Sums. IEEE Information Theory Workshop (ITW 2010), Dublin, August-September 2010.


