Applications of Boolean Functions in Cryptography

Chunhui Wu
Guangdong University of Finance
Department of Computer Science
510521 Guangzhou, People’s Republic of China
email: chunhuiwu@163.com

Bernd Steinbach
Freiberg University of Mining and Technology
Institute of Computer Science
D-09596 Freiberg, Germany
email: steinb@informatik.tu-freiberg.de

Abstract
Nonlinear Boolean functions are considered for a long time to construct symmetric cryptosystems. In order to resist the known attacks, many properties of Boolean functions must be utilized. In this paper we analyze some major properties according to different attacks and list some research topics. We also analyze the performance of the S-box in classical algorithms such as DES and AES, and the distribution of all Boolean functions of four variables with regard to their weight and nonlinearity.

1 Introduction
There are two types of cryptosystems, called public key cryptosystem and symmetric key cryptosystem, due to the nature of the keys. In a public key cryptosystem, every user has a pair of keys, called public key and secret key, where the public key can be made public and the secret key is kept secret. The security of public key cryptosystem is based on the hardness of inverting a one-way function. The idea of the public key cryptosystem was first proposed in the groundbreaking work of Diffie and Hellman [5] in 1976, and the first practical scheme was proposed by Rivest, Shamir, and Adleman [9] in 1978.

Differently, one user shares the same secret key with his partner to communicate secretly in symmetric key cryptosystems. The symmetric key cryptosystem can be further divided into two subclasses, called stream cyphers (see Figure 1) and block cyphers (see Figure 2). As the name emphasizes, the messages are encrypted bit-by-bit in stream cyphers while in block cyphers the messages are encrypted by blocks. The Data Encryption Standard (DES) is the first and most significant modern symmetric encryption algorithm. It was published by the United States’ National Bureau of Standards in 1977 and is widely used all over the world. In 1997, the United States’ National Institute of Standards and Technology (NIST) announced an open call for a
new symmetric key block cypher algorithm as the new encryption standard to replace the DES. The new algorithm would be named the Advanced Encryption Standard (AES). In October, 2000, NIST announced that Rijndael has been selected as proposal for the AES.

Public key cryptosystem and symmetric key cryptosystem have different advantages and disadvantages, and we usually combine both of them to build a practical cryptosystem. Public key cryptosystem provides a convenient technique to distribute the keys, while symmetric key cryptosystem can efficiently be implemented, and it is suitable to encrypt large data. Hence, we usually use in a practical cryptosystem the public key technique to distribute the keys between two users; after that the keys are used to encrypt messages using the symmetric key technique.

The most important part of a stream cypher is the key stream generator, which provides the overall security for stream cyphers. Nonlinear Boolean functions were preferred for a long time to construct the key stream generator. In order to resist several know attacks, many requirements have been proposed on the Boolean functions [6]. Attacks against the cryptosystems have forced deep research on Boolean function to allow us a more secure encryption.

The block encryption maps the plaintext \((p_1, p_2, \ldots, p_n)\) to the cyphertext \((c_1, c_2, \ldots, c_m)\), which can be treated as a multiple-output Boolean function from \(B^n\) to \(B^m\) (also called \(n \times m\) S-box). Furthermore, if \(n = m\) and if the map is bijective, then it is a Boolean permutation, which is composed by a group of Boolean functions with certain relations. Hence, the research of block cyphers also requires the research of Boolean permutations. Of course, the research of multiple-output Boolean functions is significantly more difficult than the research of (single-output) Boolean functions; consequently, the results for multiple-output Boolean functions are not so deep and systematic as in the case of simple Boolean functions.

Furthermore, the security of symmetric cryptosystems is strongly influenced by Boolean functions. Hence, the knowledge about cryptographic properties of Boolean functions is very important. The research fields of Boolean functions regarding the cryptography include:

- the design and implementation of Boolean functions;
- the properties of Boolean functions;
- the existence, the distribution, the construction and the counting of Boolean functions with certain properties;
- the relationships such as equivalence, elimination, compatibility and conflicts between different properties, especially in quantity form;
- the tradeoff between different properties in order to improve the performance of cryptosystems;
- the study of new properties according to new attacks.

The rest of the paper is organized as follows: Some major properties of Boolean functions regarding their cryptographic application are defined in Section 2. The construction of stream cyphers using Boolean functions and their properties according to major attacks are summarized in Section 3. The applications of Boolean functions in block cyphers and the properties according to some major attacks are given in Section 4, where the performance of S-boxes in two classical block cyphers of DES and AES is also analyzed. The experimental results of the distribution of all Boolean functions of \(B^4\) with regard to their weights and their quantities of nonlinearity are explored in Section 5. Some further research topics are given in Section 6 before we conclude this paper in Section 7.

2 Preliminaries

A Boolean function \(f\) of \(n\) variables is a unique mapping \(B^n \to B\), where \(B = \{0, 1\}\). \(B\) can also be seen as the Galois field \(GF(2)\) which contains the same two elements, and addition and multiplication are calculated module 2, which are denoted by \(x_1 \oplus x_2\) and \(x_1 x_2\) for short.

The Walsh transformation [7, 13] is useful to study certain properties, especially the nonlinearity and the correlation immunity of a Boolean function.
Their row sum divided by 4 results in the exponent $f$.

Example 2 (Inverse Walsh Transformation)

Table 2 shows the detailed steps of the inverse upper right part of Table 1. The results of $f$ of $w_0$ part of Table 1. The linear functions are calculated as sum of the four integer values of the lastly calculated columns. The coefficients either 1 or -1 as shown in the lower right part of Table 1. The wanted four Walsh coefficients are calculated as sum of the four integer values of the lastly calculated columns. The coefficients of the Walsh Spectrum are: $S(f)(00) = 2$, $S(f)(10) = 2$, $S(f)(01) = -2$, and $S(f)(11) = 2$.

Example 2 (Inverse Walsh Transformation)

Table 2 shows the detailed steps of the inverse Walsh transformation based on (2) for the Walsh Spectrum calculated in Example 1. The product values of $(-1)^{w \cdot x}$ and the values of the given Walsh Spectrum are labeled by $h(w, x)$ in Table 2. Their row sum divided by 4 results in the exponent $f(x)$ in $(-1)^{f(x)}$ so that the original Boolean function $f(x_1, x_2) = x_1 \land \overline{x}_2$ is finally found.

### Table 1: Detailed calculation of the Walsh Spectrum of the function $f(x_1, x_2) = x_1 \land \overline{x}_2$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$f(x)$</th>
<th>$w \cdot x$</th>
<th>$f(x) \oplus w \cdot x$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$w=(00)$</td>
<td>$(10)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$(-1)^{f(x)} \oplus w \cdot x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>-1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S(f)(w)=\sum_{w \in \mathbb{GF}(2)^n}h(w, x)\cdot (-1)^{w \cdot x} \cdot \frac{1}{4} \sum_{w}h(w, x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

### Definition 1 (Walsh Transformation)

Let $x = (x_1, \ldots, x_n), w = (w_1, \ldots, w_n) \in \mathbb{GF}(2)^n$, and $w \cdot x = w_1 x_1 \oplus \cdots \oplus w_n x_n \in \mathbb{GF}(2)$. The Walsh transformation of a function $f$ on $\mathbb{GF}(2)^n$ is defined by the mapping $S(f) : \mathbb{GF}(2)^n \to \mathbb{R}$:

$$S(f)(w) = \sum_{x \in \mathbb{GF}(2)^n}(-1)^{f(x)}(-1)^{w \cdot x} = \sum_{x \in \mathbb{GF}(2)^n}(-1)^{f(x) \oplus w \cdot x}.$$  

(1)

$f(x)$ can be recovered by the inverse Walsh transformation

$$(-1)^{f(x)} = 2^{-n} \sum_{w \in \mathbb{GF}(2)^n}S(f)(w)\cdot (-1)^{w \cdot x}.$$  

(2)

The Walsh spectrum of $f$ is the list of the $2^n$ Walsh coefficients given by (1).

This connects the Boolean function with a real vector of $2^n$ elements. The Walsh spectrum at a point $w$ counts the number of coincides of function $f(x)$ with the linear function $l(x) = w \cdot x$ minus the number of points with $f(x) \neq l(x)$.

### Example 1 (Walsh Transformation)

The Walsh Spectrum $S(f)(w)$ of the Boolean function $f(x_1, x_2) = x_1 \land \overline{x}_2$ must be calculated. We start with the associated function table in the top left part of Table 1. The linear functions 0, $x_1$, $x_2$, and $x_1 \oplus x_2$ are calculated by $w \cdot x$ for all values of $w$ and are shown in the middle part of Table 1. The next step is the calculation of the addition modulo 2 between $f(x)$ and each of the four linear functions. These results are shown in the upper right part of Table 1. The results of $(-1)$ to the power of the last intermediate values are either 1 or -1 as shown in the lower right part of Table 1. The wanted four Walsh coefficients are calculated as sum of the four integer values of the lastly calculated columns. The coefficients of the Walsh Spectrum are: $S(f)(00) = 2$, $S(f)(10) = 2$, $S(f)(01) = -2$, and $S(f)(11) = 2$.

### Example 2 (Inverse Walsh Transformation)

Table 2 shows the detailed steps of the inverse Walsh transformation based on (2) for the Walsh Spectrum calculated in Example 1. The product values of $(-1)^{w \cdot x}$ and the values of the given Walsh Spectrum are labeled by $h(w, x)$ in Table 2. Their row sum divided by 4 results in the exponent $f(x)$ in $(-1)^{f(x)}$ so that the original Boolean function $f(x_1, x_2) = x_1 \land \overline{x}_2$ is finally found.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$w \cdot x = $</th>
<th>$(-1)^{w \cdot x} = h(w, x)$</th>
<th>$S(f)(w)$ = $\sum_{w}h(w, x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$w=(00)$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$w=(10)$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$w=(01)$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$w=(11)$</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

$S(f)(w)$ = 2, 2, -2, 2
Cryptographic applications of Boolean functions are influenced by the following properties.

1. **Balance.** A Boolean function $f(x)$ is a balanced function if its weight\(^1\) $w(f(x)) = 2^{n-1}$, which means that half of the values of $f(x)$ are equal to 1 and half of the values are equal to 0.

2. **Nonlinearity.** The nonlinearity describes the smallest distance between $f$ and all linear functions $l(x) \in L_n(x)$. It is defined as

$$N_f = \min_{l(x) \in L_n(x)} d(f(x), l(x)) = \min_{l(x) \in L_n(x)} w(f(x) \oplus l(x)),$$

where $L_n(x)$ is the set of all $n$-ary linear functions (including affine functions).

We have the following relation between the nonlinearity and the Walsh spectrum.

$$N_f = \frac{2^n - \max_{w \in GF(2)^n} |S_f(w)|}{2}.$$  \hspace{1cm} (4)

3. **Correlation Immunity (CI).** Correlation immunity is a statistical property of a Boolean function. $f(x)$ is correlation immune of order $m$ (denoted by CI($m$)), $1 \leq m \leq n$, if and only if its values are statistically independent of any subset $\{x_{i_1}, \ldots, x_{i_m}\}$ of $m$ input variables, that means, for any $(a_{i_1}, \ldots, a_{i_m}) \in GF(2)^m$,

$$w(f(x)|x_{i_1} = a_{i_1}, \ldots, x_{i_m} = a_{i_m}) = \frac{w(f(x))}{2^m}.$$  \hspace{1cm} (5)

Another definition of correlation immunity is: $f(x)$ is CI($m$) if and only if $S_f(w) = 0$, for all $w$, $1 \leq w(w) \leq m$.

4. **Strict Avalanche Criterion (SAC).** A Boolean function $f(x)$ of $\mathbb{B}^n$ satisfies the Strict Avalanche Criterion (SAC for short) if and only if the function $f(x) \oplus f(x \oplus a)$ is balanced for every $a$ in $\mathbb{B}^n$ with the Hamming weight 1.

5. **Propagation Criterion (PC).** A Boolean function $f(x)$ of $\mathbb{B}^n$ satisfies the Propagation Criterion of the degree $k$ (PC($k$) for short) if and only if the function $f(x) \oplus f(x \oplus a)$ is balanced for every $a$ with $1 \leq w(a) \leq k$. It generalizes the notion of SAC, which is clearly identical to PC(1).

6. **Algebraic Immunity.** Algebraic attack gains much concerns since 2002 [3, 4, 8]. An over-defined system of high degree equations between the original status and the key stream is established and solved using linearization methods. In order to resist an algebraic attack, the Boolean function $f(x)$ should have the property that there is no non-zero Boolean function $g(x)$ such that $f(x)g(x) = h(x)$ (or $\overline{f(x)}g(x) = h(x)$) and $h(x)$ has a low algebraic degree\(^2\). It has been proven that the lowest degree of all the multiples of $f(x)$ and $\overline{f(x)}$ equals to the lowest degree of all the annihilators of $f(x)$ and $\overline{f(x)}$ (if $f(x)g(x) = 0$ and $g(x) \neq 0$, then $g(x)$ is called an annihilator of $f(x)$). This lowest degree is defined as the algebraic immunity of $f(x)$ (or $\overline{f(x)}$), which can be described as:

$$AI(f(x)) = AI(\overline{f(x)}) = \min \{deg(g(x)) : f(x)g(x) = 0 \text{ or } \overline{f(x)}g(x) = 0\}.$$  \hspace{1cm} (6)

The strict avalanche criterion (SAC) and the propagation criterion (PC) are sometimes collectively referred to as nonlinearity, since we have the fact that function $f(x)$ is PC($n$) if and only if $f(x)$ has the largest nonlinearity, i.e., $f(x)$ is a bent function. Bent functions are Boolean functions which have the largest distance to all linear Boolean functions. The drawback of bent functions for cryptographic applications is that they are not balanced and their algebraic degrees are low. In practice, functions with a high degree of propagation will be more valuable than bent functions which are the best cases in nonlinearity.

\(^1\)The (Hamming) weight of $f(x) \in GF(2)^n$ is the number of values 1 in the truth table of $f(x)$, denoted by $w(f) = |\{x|f(x) = 1, x \in GF(2)^n\}|$.

\(^2\)A Boolean function $f(x_1, x_2, \ldots, x_n) : GF(2)^n \rightarrow GF(2)$ can be written as a multi-variable polynomial, which is called the Algebraic Normal Form (ANF): $f(x_1, x_2, \ldots, x_n) = a_0 \oplus a_1 x_1 \oplus \sum_{1 \leq i \leq n} a_i x_i \oplus \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j \oplus \cdots \oplus a_{12 \ldots n} x_1 x_2 \cdots x_n$, where $a_0, a_1, \ldots, a_{12 \ldots n} \in GF(2)$. The algebraic degree of $f(x)$ is defined as the maximum degree of the monomials in its ANF.
3 Boolean Functions in Stream Ciphers

3.1 Constructions of Stream Ciphers Using Boolean Functions

Stream ciphers can be divided into two parts (see Figure 1), the first part is the key stream generator and the second part is the encryption transformation. The encryption transformation can simply be the XOR of the plaintext with the key stream. Hence, the security of the stream cipher depends on the security of the key stream.

In order to design a secure key stream generator, a nonlinear transformation should be used. R. Rueppel [10] divided the key stream generator into two parts, which are the driven part and the nonlinear combination part, for the convenience of theoretical analysis. The driven part controls the states of the generator; it also provides a sequence with long period and good statistical properties to the nonlinear combination part. The nonlinear combination part combines the sequences from the driven part into a sequence with good cryptographic properties.

A Linear Feedback Shift Register (LFSR) is used as the driven part for a long time, because the structure of LFSR is simple, is easy to implement, and it can run very fast. The design of the driven part is relatively simple, since the theory of LFSR is mature, especially that we have a maximum length sequence with good pseudo-randomness. Hence, the design of nonlinear combination part is the key point in the design of key stream generator.

Nonlinear Boolean functions were preferred for a long time for the nonlinear combination part. Such nonlinear Boolean functions are called filter functions (one LFSR is used as the driven part, see Figure 3 (a)) or combination functions (several LFSRs are used as the driven part, see Figure 3 (b)). The corresponding key stream generators are called nonlinear filter sequence generators (see Figure 3 (a)) and nonlinear combination sequence generators (see Figure 3 (b)). The subscript $j$ in Figure 3 denotes the time clock and we have one bit key output at each time clock.

3.2 Attacks and Required Properties

Two classical attacks against stream ciphers are the correlation attack and the linear attack.

We take the nonlinear combination sequence generator for example. The idea of the correlation attack is to make use of the correlation between the output sequence $\{k_j\}$ and every input sequence $\{a_{ij}\}$ ($1 \leq i \leq n$) to recover the initial status and the feedback function of LFSR using statistical methods. So we require the Boolean function $f$ that satisfies correlation immunity to a certain extend to resist correlation attack.

The essence of linear attack is to use a sequence with a low linear complexity to approach a sequence with a high linear complexity. The linear attack can be efficient for a big value of $|\max S_f(w)|$ [13]. Consequently, we require a high nonlinearity of the Boolean function to resist linear attacks. Other basic requirements of Boolean functions include the balance and the high algebraic degree, where the balance affects the statistic property of the key stream and the algebraic degree directly affects the linear complexity of key stream.

The performance of Boolean functions against algebraic attacks gained much concern since 2003 [2, 3] in LFSR-based stream ciphers. An algebraic attack works as follows. Take the
such that

Boolean function $f$ equations with degree less than

does the details of one round. Figure 4 (c) summarizes some basic information about DES.

has the advantage that encryption and decryption operations are very similar. Therefore, the

internal function called a Feistel function (F function, or round function). The Feistel structure

structure is named after the German physicist and cryptographer Horst Feistel to honor his pio-

The Feistel structure is a symmetric structure used in many block ciphers including DES. This

DES has 16 rounds of iterations. Even though DES is being replaced by AES because both

structure of iteration, which iterates a simple function

every bit in the plaintext and key should have influence on every bit in the cyphertext. A block

such that the encryption transformation should have enough nonlinearity. Diffusion means that

means that the statistical features between the plaintext and the cyphertext should be complex,

functions which can totally resist fast algebraic attacks, but we can choose some special Boolean

Table 3: Attacks on symmetric cryptosystems and required properties on Boolean functions

<table>
<thead>
<tr>
<th>(a) stream ciphers</th>
<th>required properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistic analysis</td>
<td>balance</td>
</tr>
<tr>
<td>linear attack</td>
<td>nonlinearity</td>
</tr>
<tr>
<td>correlation attack</td>
<td>correlation immunity</td>
</tr>
<tr>
<td>algebraic attack</td>
<td>algebraic immunity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(b) block ciphers</th>
<th>required properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistic analysis</td>
<td>orthogonality (balance)</td>
</tr>
<tr>
<td>linear attack</td>
<td>nonlinearity</td>
</tr>
<tr>
<td>differential attack</td>
<td>differential uniformity and robustness</td>
</tr>
<tr>
<td>algebraic attack</td>
<td>algebraic immunity</td>
</tr>
</tbody>
</table>

nonlinear filter sequence generator of Figure 3 (a) for example (the analysis of nonlinear combi-
nation sequence generator is similar), denote $a=(a_0, a_1, \ldots, a_{n-1})$ the initial status of LFSR,
and $L$ the linear state transition function of LFSR, then the output key can be denoted by
$f(L'(a))=k_t$ at time $t$. If the attacker knows $L$, $f$ and some key stream $k_t$, then the initial
status $a$ can be revealed by solving a system of nonlinear equations. If $f$ has high algebraic
degree, it is difficult to solve this system of equations. However, if $f$ has low degree multiples
$h=fg$, then we get as a result of the multiplication of both sides of $f(L'(a))=k_t$ with $g(L'(a))$:

$$f(L'(a))g(L'(a)) = k_t g(L'(a)) .$$

If $k_t = 0$, then $f(L'(a))g(L'(a)) = h(L'(a)) = 0$. If $k_t = 1$, then $g(L'(a)) = h(L'(a)) = 0$. If $g$
has a low algebraic degree, then $g(L'(a)) \oplus h(L'(a))$ also has a low algebraic degree. Hence, the
cryptanalysis can be converted to solving an over-defined system of equations with relative low
degree. If a Boolean function has big algebraic immunity then it can resists algebraic attacks
to a certain extend. However, even a Boolean function with a big algebraic immunity may not
resist a fast algebraic attack [2], which is an improved algebraic attack.

A fast algebraic attack does not require the Boolean function $f(x)$ to have a low degree
multiple. If there exists a low degree (less than $\text{AI}(f)$) of the function $g(x)$ such that the degree
of $h(x) = f(x)g(x)$ is not too big, then the fast algebraic attack will still be efficient. If the
degree of $g(x)$ is $e$ ($e < \frac{n}{2}$), then fast algebraic attack can be converted to solve a system of
equations with degree less than $e$. The research in fast algebraic attack shows that for any
Boolean function $f(x)$ and integers $e, d$ such that $e + d \geq n$, there exists Boolean function $g$
such that $h(x) = f(x)g(x)$, with $\text{deg}(g(x)) \leq e$ and $\text{deg}(h(x)) \leq d$. So there are no Boolean
functions which can totally resist fast algebraic attacks, but we can choose some special Boolean
functions which have high complexity against a fast algebraic attack.

Table 3 summarizes the main attacks and required properties of Boolean functions.

4 Boolean Functions in Block Cyphers

Confusion and diffusion are two general design principals proposed by Shannon [11]. Confusion
means that the statistical features between the plaintext and the cyphertext should be complex,
such that the encryption transformation should have enough nonlinearity. Diffusion means that
every bit in the plaintext and key should have influence on every bit in the cyphertext. A block
cypher should be hard to analyze and easy to implement at the same time. Hence, we use the
structure of iteration, which iterates a simple function $F$ (easy to implement) many times; e.g.,
the DES has 16 rounds of iterations. Even though DES is being replaced by AES because both
its block and its key are not long enough for brute force attack nowadays, it is still valuable for
its basic theory and designing ideas. Figure 4 briefly describes the DES algorithm.

Figure 4 (a) shows the overall Feistel structure of DES, which has 16 rounds of iterations.
The Feistel structure is a symmetric structure used in many block ciphers including DES. This
structure is named after the German physicist and cryptographer Horst Feistel to honor his pio-
near research in cryptography. The Feistel structure is an iteratively repeated structure with an
internal function called a Feistel function (F function, or round function). The Feistel structure
has the advantage that encryption and decryption operations are very similar. Therefore, the
size of the code or circuit required to implement such a cipher is nearly halved. Figure 4 (b)
shows the details of one round. Figure 4 (c) summarizes some basic information about DES.
Ciphertext (64 bits) for 16 rounds

Plaintext (64 bits) → Initial Permutation IP → subkey1 → F → subkey2 → F → ··· → subkey16 → F → Inverse Initial Permutation IP\(^{-1}\) → Ciphertext (64 bits)

Half Block (32 bits) → Extension → S1 S2 S3 S4 S5 S6 S7 S8 → Permutation → Subkey (48 bits)

(a) The overall Feistel structure of DES

(b) The Feistel function (F function) of DES

General Designers: IBM
First published: 1977 (standardized in January 1979)
Successors: Triple DES

Cipher Details
Key sizes: 56 bits
Block sizes: 64 bits
Structure: Feistel network
Rounds: 16

(c) Summarize of DES

The key point of DES is the design of the S-box. It is a map from \(\mathbb{B}^6\) to \(\mathbb{B}^4\), i.e., a function group composed by 4 Boolean functions which depend on 6 variables. Table 4 shows for example the box \(S_1\); the 6 input bits are split into the two most significant bits assigned to the left column and the four least significant bits of the top row. Each of the four bits of the binary encoded values of Table 4 represents one associated function value of Boolean function \(f_1(x_1, \ldots, x_6), \ldots, f_4(x_1, \ldots, x_6)\). We omit further details here.

There are also important attacks in block ciphers, such as the linear attack, the differential attack, and the algebraic attack (see Table 3 (b)). The linear attack makes use of the linear relationship of certain bits between the plaintext, the ciphertext and the key in every round. The XOR of two plaintexts and the XOR of two corresponding ciphertexts are compared in a differential attack. The idea of the algebraic attack is the same as in stream ciphers. It tries to break the system by solving an over-defined system of algebraic equations which describes

<table>
<thead>
<tr>
<th>Table 4: DES S-box (S_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7</td>
</tr>
<tr>
<td>01 0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8</td>
</tr>
<tr>
<td>10 4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0</td>
</tr>
<tr>
<td>11 15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13</td>
</tr>
</tbody>
</table>
Table 5: Properties of DES S-box and AES S-box

<table>
<thead>
<tr>
<th></th>
<th>bound ((n \times m))</th>
<th>DES S-box ((6 \times 4))</th>
<th>AES S-box ((8 \times 8))</th>
</tr>
</thead>
<tbody>
<tr>
<td>orthogonality</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>algebraic degree of</td>
<td>(\leq n - 1)</td>
<td>5 (5)</td>
<td>7 (7)</td>
</tr>
<tr>
<td>every function</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nonlinearity of</td>
<td>(\leq 2^{n-1} - 2^{\frac{n}{2}-1})</td>
<td>14 (\sim) 22 (28)</td>
<td>112 (120)</td>
</tr>
<tr>
<td>every function</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>differential uniformity</td>
<td>(\geq 2^{n-m+1})</td>
<td>16 (8)</td>
<td>4 (2)</td>
</tr>
<tr>
<td>robustness</td>
<td>(\leq (2^{-1} + 2^{m-n-1})(1 - 2^{-m+1}))</td>
<td>0.316 (\sim) 0.469</td>
<td>0.984375</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.546875)</td>
<td>(0.9921875)</td>
</tr>
</tbody>
</table>

The S-box. The algebraic attack is deterministic and thus the security does not have to grow exponentially in the number of rounds, while classical attacks (e.g., the linear attack or the differential attack) are based on probabilistic characteristics, which makes their security grow exponentially with the number of rounds.

The security of block ciphers mainly depend on the properties of the S-box (multiple-output Boolean function), which is the only nonlinear part. In order to resist known attacks, we also require the properties of the orthogonality (corresponding to the balance of a single-output Boolean function), a high algebraic degree, a high nonlinearity, and a high algebraic immunity. Additionally, we have to evaluate two more properties called differential uniformity and robustness [13] which indicate the ability to resist differential attacks.

A detailed analysis of the DES shows that all the eight S-boxes are orthogonal, all 32 Boolean functions have the maximum algebraic degree and even have the maximum algebraic immunity. However, the S-boxes have some weakness in their nonlinearity, their differential uniformity, and their robustness are not so good. It partially answers the question why DES suffers from linear attacks and differential attacks.

The AES S-box is constructed based on the multiply-inverse operation on \(GF(2^8)\). A detailed analysis shows that the AES S-box is orthogonal, i.e., it is a Boolean permutation on \(GF(2^8)\). Not only all the eight Boolean functions, but the S-box as a whole has the maximum algebraic degree. The low differential uniformity and the high robustness (close to its upper bound) guarantee the ability of AES to resist differential attack. Both the eight Boolean functions and the S-box as a whole also have a high nonlinearity which is closed to totally nonlinear functions, i.e., bent functions.

See Table 5 for the detailed numerical results of the DES S-box and the AES S-box, where the numbers in brackets are the upper bounds (lower bound for differential uniformity). It may be more appropriate to study the multi-output Boolean functions (S-box) as a whole. This leads to stronger definitions of the algebraic degree and the nonlinearity [13], which considers any linear combinations of the Boolean functions in an S-box. The AES S-box reaches the upper bounds for the algebraic degree and the nonlinearity considering this stronger definition, while it is not the case in DES S-boxes. The numerical results need further calculations.

5 Experimental Results Regarding the Nonlinearity

One result of the analysis of several attacks against cryptosystems is that balanced Boolean functions with a high nonlinearity are needed in cryptography. How these properties are distributed over all Boolean functions of a certain Boolean space? We explored all \(2^{2^4} = 65536\) Boolean functions regarding these two properties. Figure 5 shows the result of this experiment and gives an interesting insight into the distribution of all Boolean functions of \(\mathbb{B}^4\) regarding these properties.

Due to the definition (3) of the nonlinearity, the 32 linear functions of \(\mathbb{B}^4\) have nonlinearities of 0. Italic numbers in the bottom row within the frame of Figure 5 indicate these linear functions.
Almost all linear functions are balanced (weight: 8) except the functions $f = 0$ (weight: 0) and $f = 1$ (weight: 16). These linear functions determine the nonlinearity of all other functions.

Each additional value 1 which is added to the Boolean function $f = 0$ increases the nonlinearity up to the value 4. Similarly, each removed value 1 from a not constant linear function (weight: 8) also increases the nonlinearity up to the value 4 of the nonlinearity. Bent functions with the maximal nonlinearity of 6 are indicated by bold numbers in Figure 5. Balanced functions are emphasized by light gray background in Figure 5. It can be seen in Figure 5 that there is no balanced bent function and the constant functions $f = 0$ and $f = 1$ cause a mirror symmetry around the balanced functions in the distribution of the explored properties.

### 6 Further Research on Cryptographic Boolean Functions

There are many research topics on cryptographic Boolean functions. We list some topics from the application side.

1. A basic aim is the construction and enumeration of Boolean functions with certain properties, e.g., the construction of balanced and correlation immune functions (called resilient functions) with a high nonlinearity [14].

2. Generalized bent functions are studied in [15]. Researchers generalized the definition of bent functions by relaxing the propagation criterion (i.e., requiring that $f(x) \oplus f(x \oplus a)$ is nearly balanced) or by limiting the Walsh spectrum. These generalized bent functions are more suitable in applications but systematic construction methods must be found.

3. The algebraic immunity should also be considered together with other cryptographic properties, such as the balance, the algebraic degree and the nonlinearity [1, 12]. There are only few results on the relationship between the algebraic immunity and the correlation immunity or the construction of Boolean functions resisting fast algebraic attacks.

4. Further aims are the realization and improvement of algorithms to evaluate these cryptographic indicators. One can even choose a Boolean function randomly, but comprehensive tests are needed to show its security. It will be a good work to develop a toolkit which can do comprehensive tests.

### 7 Conclusions

Boolean functions are the most important part in symmetric cryptosystems. In this paper, we summarized some major properties of Boolean functions to resist several attacks, and analyzed the performance of the S-box in classical algorithms such as DES and AES. Moreover, we gave the experimental results of the distribution of all Boolean functions of $\mathbb{F}_4$ regarding the weight and nonlinearity. One may want to find a Boolean function that holds all the security properties, but our experimental results had shown that this is impossible since some properties conflict with each other. Hence, a tradeoff is necessary between the required properties. The construction of Boolean functions with certain properties is the main research subject of cryptographic Boolean
functions. We may also require new properties because attacks never stop. The quantitative evaluation of Boolean functions regarding security applications and the relation between different properties remain an important research topic.

References


