Analysis and Design of Block Ciphers

by

Amr Mohamed Youssef

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for the degree of
Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
December 1997

Copyright © 1997 Amr Mohamed Youssef
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-27864-6
Abstract

In this thesis we study various cryptographic properties of boolean mappings from \( n \) bits to \( m \) bits. In particular, we derive expressions for the expected size of the maximum XOR table entry and the maximum Linear Approximation Table entry for some combinatorial structures of interest such as regular (balanced) mappings, and injective mappings. We derive similar expressions for the expected value of different forms of information leakage and relate different forms of information leakage to the spectral properties of the function. We also extend the definitions of many cryptographic criteria to multi-output boolean functions and study the relationship between the Walsh-Hadamard transform and various types of information leakage.

A new construction method for highly nonlinear injective s-boxes is presented. It is shown that the resistance of CAST-like encryption algorithms (based on randomly selected substitution boxes) to the basic linear cryptanalysis was underestimated in previous work.

We introduce a new class of Substitution Permutation Networks (SPNs) with the advantage that the same network can be used to perform both the encryption and the decryption operations. Different cryptographic properties of this class such as resistance to both linear and differential cryptanalysis are examined. We also present two construction methods for involution linear transformations for SPNs based on Maximum Distance Separable codes.

An analytical model for the avalanche characteristics of SPNs with different linear transformation layers is developed. We also prove a conjecture by Cusick regarding the number of functions satisfying the Strict Avalanche Criterion.
Dedicated to my parents

For their eternal love, encouragement, and support, I am grateful.
Acknowledgments

This work would never have been possible without the guidance, encouragement, and suggestions of my supervisor, Dr. Stafford Tavares. It is my pleasure to thank him for all of his help.

I wish to thank Dr. Moustafa Fahmy who introduced me to Dr. Tavares.

I gratefully acknowledge the financial support of the Ontario Ministry of Education, Telecommunication Research Institute of Ontario (TRIO), and Queen’s University.

Special thanks and appreciation go to my wife, Ayda, for her many years of patient support and encouragement.
# List of Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{Z}_2$</td>
<td>The set of binary numbers</td>
</tr>
<tr>
<td>$\mathbb{Z}_2^n$</td>
<td>The set of binary $n$-tuples</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>XOR operation</td>
</tr>
<tr>
<td>$x$</td>
<td>A variable in $\mathbb{Z}_2^n$ or $GF(2^n)$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>A scalar variable</td>
</tr>
<tr>
<td>$X$</td>
<td>A random variable in $\mathbb{Z}_2^n$ or $GF(2^n)$</td>
</tr>
<tr>
<td>$#{\cdot}$</td>
<td>The cardinality of the enclosed set</td>
</tr>
<tr>
<td>$f(\cdot)$</td>
<td>A mapping from $\mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$</td>
</tr>
<tr>
<td>$wt(a)$</td>
<td>The Hamming weight of $a$</td>
</tr>
<tr>
<td>$wt(f)$</td>
<td>$#{x</td>
</tr>
<tr>
<td>$a \cdot x$</td>
<td>The dot product of $a$ and $x$ over $\mathbb{Z}_2$</td>
</tr>
<tr>
<td>$ax$</td>
<td>The multiplication of $a$ and $x$ over $GF(2^n)$</td>
</tr>
<tr>
<td>$F_c(w)$</td>
<td>The Walsh transform of $(-1)^{c \cdot f(x)}$</td>
</tr>
<tr>
<td>$LAT(a, b)$</td>
<td>Linear Approximation Table entry</td>
</tr>
<tr>
<td>$N_{\Delta x \Delta y}$</td>
<td>XOR distribution table entry</td>
</tr>
<tr>
<td>$LAT^*$</td>
<td>$\max_{a \neq 0, b}</td>
</tr>
<tr>
<td>$XOR^*$</td>
<td>$\max_{\Delta x \neq 0, \Delta y} N_{\Delta x \Delta y}$</td>
</tr>
<tr>
<td>$\mathcal{N}L_f$</td>
<td>Nonlinearity of $f$</td>
</tr>
<tr>
<td>$In(n, m)$</td>
<td>Number of injective functions $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$B(n, m)$</td>
<td>Number of balanced functions $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
<td>The entropy function</td>
</tr>
<tr>
<td>$h(\cdot)$</td>
<td>The binary entropy function</td>
</tr>
<tr>
<td>$I(\cdot, \cdot)$</td>
<td>The mutual information</td>
</tr>
<tr>
<td>$SL(\cdot, \cdot)$</td>
<td>Static information leakage</td>
</tr>
<tr>
<td>$SSL(\cdot)$</td>
<td>Self static information leakage</td>
</tr>
<tr>
<td>$DL(\cdot, \cdot)$</td>
<td>Dynamic information leakage</td>
</tr>
<tr>
<td>$Tr(\cdot)$</td>
<td>Trace function</td>
</tr>
</tbody>
</table>
# Table of Contents

Abstract .............................................................................................................. ii
 List of Notation ................................................................................................. v
 List of Figures .................................................................................................... xi
 List of Tables .................................................................................................... xiii

Chapter 1 Introduction ...................................................................................... 1
  1.1 Motivation for this Research ................................................................. 2
  1.2 Outline of Remainder of Thesis ............................................................ 3

Chapter 2 Previous Research ......................................................................... 4
  2.1 Architectures ............................................................................................ 4
  2.1.1 Substitution-Permutation Networks ................................................. 5
  2.1.2 DES-like Structure (Feistel Networks) ............................................. 7
  2.1.3 Other Structures ................................................................................ 9
  2.2 S-box versus Non S-box Approaches ................................................... 9
  2.3 Cryptanalysis .......................................................................................... 10
  2.3.1 Exhaustive Search ............................................................................ 11
  2.3.2 Differential Cryptanalysis ............................................................... 12
  2.3.3 Linear Cryptanalysis ....................................................................... 14
  2.3.4 Extensions to Linear and Differential Cryptanalysis ...................... 15
  2.3.5 Other Attacks .................................................................................... 16
  2.3.6 Implementation Dependent Attacks ................................................. 17
  2.3.6.1 Timing Attacks ........................................................................... 17
  2.3.6.2 Differential Fault Cryptanalysis ............................................... 17
Chapter 3
3.4 Advanced Encryption Standard (AES) Requirements 19
2.5 Mathematical Background and Definitions 21
2.5.1 Cryptographic Criteria for Boolean Functions 21
2.5.2 The Inclusion-Exclusion Principle 25
2.5.3 The Walsh Transform of Boolean Functions 26
Chapter 3
3.2 Linear Approximation Table and XOR Distribution Table of Balanced S-boxes 29
3.3 Linear Approximation Table of Balanced S-boxes 30
3.3 XOR Distribution Table of Balanced S-boxes 32
3.4 Counting the Number of Nonlinear Balanced S-boxes 38
3.5 Conclusion 41
Chapter 4
4.1 Linear Approximation of Injective S-boxes 43
4.2 Definitions and Notation 43
4.3 Linear Approximation Table of Injective Mappings 44
4.4 Construction of Highly Nonlinear Injective S-boxes 47
4.3.1 Definitions 47
4.3.2 Construction Method I 49
4.3.3 Construction Method II 50
4.3.4 Comments on the Security of the CAST Encryption Algorithm 53
Chapter 5
5.1 Information Leakage and Spectral Properties of Boolean Functions 55
5.2 Introduction 55
5.2 Definitions 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Information Leakage of a Randomly Selected Boolean Function</td>
<td>60</td>
</tr>
<tr>
<td>5.4</td>
<td>Information Leakage of a Randomly Selected Balanced Boolean Function</td>
<td>64</td>
</tr>
<tr>
<td>5.5</td>
<td>Information Leakage of a Randomly Selected Injective Boolean Function</td>
<td>67</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Relation Between the Walsh Transform and Different Forms of Information Leakage</td>
<td>69</td>
</tr>
<tr>
<td>5.6.1.1</td>
<td>Extended Definitions</td>
<td>73</td>
</tr>
<tr>
<td>5.7</td>
<td>Conclusion</td>
<td>75</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>A New Class of Substitution-Permutation Networks</td>
<td>77</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>77</td>
</tr>
<tr>
<td>6.2</td>
<td>S-boxes</td>
<td>79</td>
</tr>
<tr>
<td>6.3</td>
<td>S-box Interconnection Layer</td>
<td>87</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Proposed Linear Transformation</td>
<td>88</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Involution Linear Transformations based on MDS codes</td>
<td>89</td>
</tr>
<tr>
<td>6.4</td>
<td>Resistance to Differential and Linear Cryptanalysis</td>
<td>93</td>
</tr>
<tr>
<td>6.5</td>
<td>Cyclic Properties of the Proposed SPN</td>
<td>96</td>
</tr>
<tr>
<td>6.6</td>
<td>Key Scheduling Algorithm</td>
<td>97</td>
</tr>
<tr>
<td>6.7</td>
<td>Performance</td>
<td>100</td>
</tr>
<tr>
<td>6.8</td>
<td>Conclusion</td>
<td>100</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Modelling Avalanche Characteristics of Substitution-</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>Permutation Networks Using Markov Chains</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>102</td>
</tr>
<tr>
<td>7.2</td>
<td>Convergence in Markov Chains</td>
<td>105</td>
</tr>
</tbody>
</table>
## List of Figures

| Figure 2.1 | General Classification of Block Cipher Architectures | 5 |
| Figure 2.2 | SPN with $N = 16$, $n = 4$, and $R = 3$. | 6 |
| Figure 2.3 | Round $i$ of DES-like Cipher | 8 |
| Figure 4.1 | CAST Round Function | 53 |
| Figure 5.1 | Lower Bound for the Binary Entropy Function | 61 |
| Figure 5.2 | Expected Value of $SSL(Y)$ for a Randomly Selected Boolean Function | 63 |
| Figure 5.3 | Expected Value of $SSL(Y)$ and Expected Value of $SL(Y | X_{n/2})$ for a Randomly Selected Boolean Function | 63 |
| Figure 5.4 | Expected Value of $DL(\Delta Y | \Delta X)$ for an $n \times n$ Random Mapping and an $n \times n$ Random Bijective Mapping | 66 |
| Figure 6.1 | SPN with $N = 16$, $n = 4$, and $R = 3$. | 77 |
| Figure 6.2 | Average Nonlinearity and Average Maximum XOR Table Entry Versus the Number of Fixed Points ($n = 8$) | 85 |
| Figure 6.3 | Involution Linear Transformation Based on MDS Codes ($M = n = 8$, Irreducible Polynomial = 11d) | 93 |
| Figure 6.4 | Normalized Distribution of Cycle Length for all $2^{16}$ Starting Points | 97 |
| Figure 6.5 | A Typical Expanded Segment of the Cycle Length Distribution of SPNs with Random Involution S-boxes | 98 |
| Figure 6.6 | Key Scheduling Algorithm | 99 |
| Figure 7.1 | SPN with $N = 16$, $n = 4$, and $R = 3$. | 102 |
| Figure 7.2 | Theoretical and Experimental Avalanche for SPN with $N = 64$, $n = M = 8$. | 115 |
| Figure 7.3 | The Transition Matrices for Model A and B ($M = n = 8$) | 117 |
Figure 7.4 The Transition Matrices for Model C and D ($M = n = 8$) . 118
Figure 7.5 The Limiting Distribution and the Eigenvalues for Model A, B, C and D ($M = n = 8$) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
# List of Tables

| Table 3.1 | Experimental Results versus Theoretical Bounds for $10^4$
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Randomly Chosen 8-bit Bijective Mappings</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Distribution of the Maximum XOR Table Entry and the</td>
</tr>
<tr>
<td></td>
<td>Maximum LAT Entry (Experimental Results for $10^4$</td>
</tr>
<tr>
<td></td>
<td>Randomly Chosen 8-bit Bijective Mappings)</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Experimental Results for $8 \times m$ Injective S-boxes</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>$\mathcal{NL}, XOR^*$ for $8 \times 16$ S-boxes Obtained Using</td>
</tr>
<tr>
<td></td>
<td>Construction Method II</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>$\mathcal{NL}, XOR^*$ for $8 \times 24$ S-boxes Obtained Using</td>
</tr>
<tr>
<td></td>
<td>Construction Method II</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Best S-box Nonlinearity Obtained By Different</td>
</tr>
<tr>
<td></td>
<td>Construction Methods</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Minimum Distance for $10^5$ Randomly Selected Involution</td>
</tr>
<tr>
<td></td>
<td>Linear Transformations ($n = M = 8$)</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Relative Speed of the Proposed SPN and Q-CAST</td>
</tr>
<tr>
<td>Table 7.1</td>
<td>The Second Largest Eigenvalues for SPNs with $N = 64$,</td>
</tr>
<tr>
<td></td>
<td>$M = n = 8$.</td>
</tr>
<tr>
<td>Table A.1</td>
<td>A parity circuit $C(n,d)$ with $n = 10$, and $d = 3$</td>
</tr>
<tr>
<td>Table C.1</td>
<td>Exact Number of Functions Satisfying SAC versus the</td>
</tr>
<tr>
<td></td>
<td>Derived Lower Bounds.</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

Until recently, cryptography has been of interest primarily to the military and diplomatic communities. Today, however, several factors have combined to stimulate great interest in commercial applications. It is becoming increasingly apparent that a primary focus of society is on the generation, transmission, processing, and storage of information. Consequently, the focus on data vulnerability and data crime in recent years has emphasized the need for techniques to protect information, especially when it is in electronic form. Among these techniques is cryptography: transformations of data intended to make the data useless to one's opponents [40]. Such transformations provide solutions to two basic problems of data security: the privacy problem, preventing opponent from extracting information from a communication channel, and the authentication problem, preventing an opponent from injecting false data into the channel or altering messages so that their meaning is changed. The field of cryptology encompasses both cryptography (cipher making) and cryptanalysis (cipher breaking).

Cryptographic systems can be classified into two groups: private-key cryptosystems, and public-key cryptosystems. In a private-key cryptosystem, the enciphering and deciphering keys are the same (or easily determined from each other), and the key is kept secret, while in public-key cryptosystems, the two keys differ in such a way that one key is computationally infeasible to determine from the other. While public-key algorithms have been suggested as a solution to the key distribution problem for private-key cryptosystems, private-key cryptosystems are still the only practical solution for cryptographic applications that require high data rates and low power consumption. Many cryptographic systems implement a public-key algorithm for key distribution, and a private-key algorithm for the data encryption.

Private-key cryptosystems can be classified into block and stream ciphers. A block cipher divides the plaintext message into successive blocks and encipher each block with the
same key. A stream cipher breaks the plaintext message into successive characters or bits and enciphers each character or bit with one element of the key stream.

1.1 Motivation for this Research

Private-key cryptosystems, with their attractive features such as high data rates and low power consumption, provide a practical solution for a variety of applications. While the most recognizable private-key cryptosystem is the Data Encryption Standard (DES) [102], DES controversy was born at the same time as DES. Since DES was proposed as a standard, there was considerable criticism. The small key size and unpublished s-box design criteria are the main issues in this criticism. Today, the cryptographic community is quite aware of the fact that breaking DES is getting a lot more feasible than it was previously thought [139], [134]. The use of triple-DES, while having good security, is excessively inefficient especially for software implementations. While the need for a serious alternative to DES increases, the dramatic failure of some other proposed alternatives, such as FEAL [124], and the “Nonlinear Parity Circuits proposed at Crypto ’90” [153] (see appendix A) illustrates the difficulty of achieving this.

Boolean functions, because they are the basic components of cryptographic transformations, attract great attention in the cryptographic field. Despite the fact that almost all cryptographic functions are multi-output boolean functions, the area of multi-output boolean functions is still almost untouched. Moreover, most of the successful cryptanalytic techniques on cryptographic functions deal with several or all output components simultaneously instead of dealing with single components separately. This work explores different cryptographic properties of multi-output boolean functions. While there are currently many block cipher proposals, almost all of them are software oriented ciphers and most of them are optimized around the 32 or 64-bit processors. The purpose of this work is not only to design a hardware efficient block cipher that is both

---

Appendix B shows an example of a dramatic failure of a key agreement scheme
secure and also software efficient on most platforms, but also to provide some design principles and analytical tools that can help in the design process.

1.2 Outline of Remainder of Thesis

Chapter two gives a review of the previous research related to our work and some of the mathematical background and the necessary definitions required throughout the thesis.

In chapter three, we study the Linear Approximation Table (LAT) and the XOR distribution table of balanced s-boxes. We also calculate the probability that any nonzero linear combination of the output coordinates of a regular s-box is an affine function.

Chapter four contains similar results for injective s-boxes. It also contains two algebraic construction methods for highly nonlinear s-boxes.

In chapter five we study different forms of the information leakage through a randomly selected boolean function. We also present the relation between the Walsh transform and different forms of information leakage.

In chapter six we introduce a new class of Substitution Permutation Networks (SPNs) and study its resistance to both linear and differential cryptanalysis.

In chapter seven we study the avalanche characteristics of four different classes SPNs.

Finally in chapter eight we give a summary of our results and directions for future work.

We note that some of the material in chapter three appeared in [141], [145]. Part of chapter four appeared in [154], [144]. Part of chapter five appeared in [146],[149]. Part of chapter six appeared in [153]. Some of the material of chapter seven appeared in [150], [152].
Chapter 2 Previous Research

A good introduction to cryptography can be found in the following books [21,38,67,85,86,120,129] and survey articles [40,41,44,80,114,115,127]. In this chapter we focus on topics relevant to our work.

2.1 Architectures

In his landmark paper, Shannon [123] presented the principles of confusion and diffusion. Because these principles are so successful in capturing the essence of the desired attributes of a block cipher, they have become the cornerstone of block cipher design.

Confusion is described as "the use of enciphering transformations that complicate the determination of how the statistics of the ciphertext depend on the statistics of the plaintext" [128] or, more briefly, to make the relation between the key and the ciphertext as complex as possible, thereby hiding the statistical features of the plaintext.

On the other hand, diffusion spreads the influence of individual plaintext characters over as much of the ciphertext as possible, thereby hiding the statistical features of the plaintext. Methods of achieving good diffusion and confusion lie at the heart of block cipher design. A general classification of block cipher design architectures is shown in Figure 2.1.
2.1.1 Substitution-Permutation Networks

Feistel [44] and Feistel, Notz, and Smith [45] were the first to suggest that a Substitution Permutation Network (SPN) architecture (Figure 2.2) consisting of iterative rounds of nonlinear substitutions on small sub-blocks referred to as substitutions (s-boxes) connected by bit permutations was a simple, effective implementation of Shannon's principle of mixing transformation based on the concepts of "confusion" and "diffusion". Keying the network can be accomplished by XORing the key bits with the data bits before each round of substitution and after the last round or by choosing a different set of s-boxes for each key.

Karn and Davida [66] introduced the concept of completeness. A technique for designing a complete SPN was given. Completeness guarantees that every ciphertext bit of the SPN depends on all input plaintext bits, and this should hold for every possible key value.
Chosen plaintext and known plaintext attacks against Tree-structured SPNs introduced by Kam and Davida are given in [57], [59].

![Diagram of SPN with N = 16, n = 4, and R = 3.](image)

**Figure 2.2** SPN with $N = 16$, $n = 4$, and $R = 3$.

In [9] Ayoub presented a variant of SPNs which incorporates random permutations and retains, with a high probability, the completeness criterion after a small number of rounds. Ayoub suggested that the s-boxes can be randomly chosen as proof of freedom from an intentional trapdoor.

Heys [55] and Heys and Tavares [60] introduced a new type of SPNs in which they replaced permutations between rounds by diffusive linear transformations to improve the avalanche characteristics of the cipher and to increase resistance to differential and linear cryptanalysis. The s-box design criteria used in their proposed SPN are diffusion and nonlinearity [60]. The regular structure used in the Heys and Tavares SPN allows both careful design and careful analysis of the network which can be considered as a good step towards designing a provably secure block cipher\(^\dagger\). The main contribution of Heys and Tavares is that they showed that the s-box interconnection layer (despite being a

\(^\dagger\) Throughout this thesis, the term "provably secure" means provably secure against a prespecified set of cryptanalytic attacks.
linear layer) plays an important role in improving the cipher's resistance to both linear and differential cryptanalysis.

SAFER [79], SHARK [108], and SQUARE [34] are some examples of block ciphers based on the SPN architecture.

2.1.2 DES-like Structure (Feistel Networks)

The National Bureau of Standards (May 1973) published a solicitation for cryptosystems in the Federal Register. This lead to the development of the Data Encryption Standard, or DES, which is believed to be the most widely used cryptosystem in the world (for unclassified information). DES was developed at IBM as a modification of an earlier system known as LUCIFER [131]. On March 17, 1975, DES was first published in the Federal Register. DES was adopted as a standard [102] for unclassified data on January 15, 1977. DES has been reviewed by the National Bureau of Standards every five years. Its most recent renewal was in January 1994, when it was renewed until 1998. It is anticipated that it will not remain a standard past 1998. On January 1997, the National Institute of Standards and Technology (NIST\textsuperscript{2}) announced the development of a Federal Information Processing Standard for Advanced Encryption Standard (AES). A summary of the proposed minimum requirements for the AES is listed in section 4.

The idea behind DES is first described by Feistel, Notz, and Smith [45] who described a block cipher construction which became the network structure for DES. As in an SPN, the DES-like network also uses s-boxes and permutations to achieve Shannon's mixing transformation but performs these operations on only half the block at a time. For round $i$, the mixing (a substitution layer followed by a permutation layer) is executed by the round function $f : Z_2^{N/2} \rightarrow Z_2^{N/2}$ on the $N/2$ bits of the right half block, $r_i$. The parameters of the function are determined by the key bits associated with round $i$, denoted by $k_i$. Each bit of the output of $f$ is then XORed with the corresponding bit.

\textsuperscript{2} NIST is a new name for the National Bureau of Standards
of the left half block, $l_i$, then the left and right half blocks are swapped. Equivalently, the cipher can be viewed as follows:

$$ (l_{i+1}, r_{i+1}) = (r_i, f(r_i, k_i) \oplus l_i). $$  \hspace{1cm} (2.1)

Schneier and Kelsey examined a generalization of the concept of Feistel (DES-like) networks [121], which they called Unbalanced Feistel Networks (UFNs). Like conventional Feistel networks, UFNs consist of a series of rounds in which one part of the block operates on the rest of the block. However, in a UFN the two parts need not be of equal size. Removing this limitation has many interesting implications for designing ciphers which are secure against linear and differential cryptanalysis. The authors also presented some UFN constructions and made some initial observations about their security.

Matsui [83] introduced a methodology for designing block ciphers, with provable security against differential and linear cryptanalysis. This methodology is based on three new principles: change of the location of the round functions, round functions with recursive structure, and s-boxes of different sizes. Changing the location of the round function realizes parallel computation of the round functions without losing provable security. The recursive structure reduces the size of the s-boxes. The use of s-boxes of different sizes
is expected to make algebraic attacks difficult. Matsui also gave specific examples of practical block ciphers that are provably secure under an independent subkey assumption and are reasonably fast in hardware as well as in software.

Some other examples for block ciphers based on DES-like structures are CAST [2][3], FEAL [124], Blowfish [119], LOKI [22][23], TEA [138], and RC5 [112].

2.1.3 Other Structures

There are many other block cipher architectures available. Many of these architectures are based on some interesting theoretical foundation. IDEA is based on the principle of mixing operations from different algebraic groups.

BEAR and LION [8] are 3-round unbalanced Feistel networks, motivated by the earlier construction of Luby and Rackoff [77] which provides a provably secure (under suitable assumptions) block cipher from pseudorandom functions using a 3-round Feistel structure.

A class of cryptosystems based on the use of interconnection networks was introduced in [104]. Another class of cryptosystems based on nonlinear finite automata was proposed in [52].

2.2 S-box versus Non S-box Approaches

Until recently, most of the block cipher proposals used a set of s-boxes in the construction of the round function to achieve the confusion effect. Some of these s-boxes are fixed, such as DES s-boxes. Others are generated dynamically as a function of the key and kept fixed throughout the encryption process. An example of this latter approach is Blowfish [119]. A more complex approach is the dynamic substitution scheme. In this case, after each substitution the s-box is re-ordered. In most of the cases, the just-used substitution value is exchanged with some other entry in the s-box selected at random. This dynamic substitution scheme, while currently applied in stream cipher design (e.g., RC4 [111],[119] and other commercial ciphers by Ritter Software Engineering [110]) has not yet been widely applied to block cipher design.
According to Nyberg [93], the most common methods for constructing s-boxes are based on: random generation, testing against a set of design criteria, algebraic construction having good properties, or a combination of these.

The advantage of the s-box based approach is that the theory of s-box design is mature enough and many techniques are available to construct cryptographically strong s-boxes. The disadvantage of the s-box approach is that it may require a large amount of memory to store the s-boxes. This disadvantage becomes very clear for large s-boxes with non-simple algebraic description (i.e., when they can only be implemented using look-up tables).

To overcome this problem, one proposal is to use key dependent operations for the s-boxes. For example, IDEA [73] achieves the required confusion effect by mixing operations from different algebraic groups (XOR, addition modulo $2^{16}$, and multiplications modulo $2^{16} + 1$). The disadvantage of this approach is that multiplication, which is the most nonlinear operation among the three operations above, usually requires a large number of logic gates in hardware and is relatively slow in software. Moreover, the scaling process, to wider block length, is not straightforward (for example IDEA cannot be scaled to 128-bit block size because $2^{31} + 1$ is not a prime [120]).

Two elegant non s-box based proposals with minimum memory requirements are TEA[138] and the RC5 family[112]. The TEA round function is constructed with simple fixed logical shift operations. In RC5 the round function is constructed with data dependent rotations. The only disadvantage of these two proposals (assuming the most common block sizes of 64 bits or 128 bits) is that rotating or shifting large blocks are not very efficient operations on 8-bit platforms.

2.3 Cryptanalysis

In this section we give a brief review of some of the cryptanalysis techniques that have been applied to block ciphers. Cryptanalytic attacks can be categorized into the following
types: (i) ciphertext only, (ii) known plaintext, (iii) chosen plaintext, and (iv) chosen ciphertext. A ciphertext attack assumes that the cryptanalyst has access to the ciphertext only. A known plaintext attack uses knowledge of the plaintext values corresponding to the available set of ciphertexts; a chosen plaintext attack assumes that the cryptanalyst can select specific values of the plaintexts and acquire the corresponding ciphertexts while a chosen ciphertext attack assumes that the cryptanalyst can select specific values of the ciphertexts and acquire the corresponding plaintexts.

Among the attacks that are applicable to different block cipher structures are: Hellman's time-memory trade-off attack [54], meet-in-the-middle attack [64], key degeneracy attack [27,39,107], maximum likelihood estimation [7], method of formal coding (MFC) [118], and related-keys attack [13].

Here we focus on the following attacks: the Wiener exhaustive key search machine [139], differential cryptanalysis [12], linear cryptanalysis [81] and some other recently proposed implementation dependent attacks.

### 2.3.1 Exhaustive Search

Despite recent improvements in analytic techniques for attacking block ciphers, exhaustive key search remains the most practical and efficient attack for ciphers with relatively short keys (i.e., \( \leq 56 \) bits). In [139] Wiener described, in detail, an exhaustive DES key search machine that costs about US$1 million and can find a key in 3.5 hours on average. The basic machine design can be adapted to attack the standard DES modes of operation for a small penalty in running time.

Wiener designed his DES key search machine by himself in about five months, which raises the question of what a well funded government agency could do. This machine, even though it was not built, is a statement that breaking DES is getting a lot more feasible than it was previously thought.
More recently, responding to a challenge, including a prize of $10,000, offered by RSA Data Security, Inc., the DESCHALL group [134] successfully found a DES key by exhaustive search over the internet. The search takes less than 3 months. On April 8, 1997 the group announced finding the key after searching almost 25% of the total key space. Full details about the project statistics can be found in [134].

2.3.2 Differential Cryptanalysis

Differential cryptanalysis [15] is a statistical attack that was developed and popularized by Biham and Shamir and has been applied to a wide range of cryptosystems including LUCIFER, DES, FEAL, REDOC, and Khafre [16]. Differential cryptanalysis is a chosen plaintext attack which examines changes in the output of the cipher in response to controlled changes in the input. In particular, the attack exploits the existence of a highly probable $(R - 1)$-round characteristic to determine a portion of the key bits, where an $r$-round characteristic is defined to be a sequence of $r$ pairs of input and output XOR differences corresponding to each round. The existence of a highly probable characteristic depends on: the XOR distribution table and the bit changes within the network [15].

The main results of applying this attack to DES reported by Biham and Shamir in [15] are: DES with 6 rounds was broken in less than 0.3 seconds on a PC using 240 ciphertexts, DES with 8 rounds was broken in less than 2 minutes by analyzing 15000 ciphertexts chosen from a pool of 50000 candidate ciphertexts, and DES with 15 rounds can be broken in about $2^{52}$ steps. More importantly, they showed that modifying the key scheduling algorithm can not make DES much stronger.

One interesting result reported is that FEAL-4 can be broken with just eight ciphertexts and one of their plaintexts.

At Crypto '92, Biham, and Shamir [17] presented a slightly modified version of their original differential cryptanalysis attack on the full 16–round DES. This attack requires
chosen plaintexts and at that time, this attack was the first known attack capable of "breaking" the full 16-round DES in less than the $2^{55}$ complexity of exhaustive search\(^\text{3}\).

In [29] Coppersmith claimed that differential cryptanalysis was known to the DES designers at IBM (within IBM, the attack was formally known as the "T attack"). Furthermore, he showed how the design criteria for the s-boxes and permutations were developed to thwart such attacks.

A closer look at differential attacks shows that for an $r$-round characteristic, only the plaintext difference and the ciphertext difference need to be fixed. That is, the intermediate differences can be arbitrarily selected. The notion of differential was introduced by Lai and Massey [73] to account for this observation. For DES-like ciphers, sequences of differences can be modeled as a Markov chain when the subkeys are assumed to be independent. In order to make a successful attack on an iterated cipher by differential cryptanalysis, the existence of good characteristics is sufficient. On the other hand, to prove security against differential attacks for such ciphers, one has to ensure that there is no differential with a probability high enough to enable successful attacks.

Knudsen and Nyberg [94] showed that for DES-like ciphers, with independent round keys, the probability of an $r$-round differential, $r \geq 4$, is less than or equal to $2p_{max}^2$ where $p_{max}$ is the highest probability for a non trivial one round characteristic. They also gave some (far from being practical) examples for block ciphers with provable security against differential cryptanalysis. Some of these examples were later broken using higher order differential cryptanalysis [72], [68].

O'Connor [97] showed that, for a randomly selected $m$-bit bijective mapping, the expected size of the largest entry in the XOR distribution table is bounded by $2^m$ while the fraction of the XOR table that is zero approaches $e^{-0.5} = 0.60653$. O'Conner [96] also showed that the above two quantities are tending to optimal values in composite permutations.

\[^{3}\text{This 1 bit reduction in exhaustive search is due to the symmetric property of DES: } DES(m, k) = \overline{DES(m, k)}\]
2.3.3 Linear Cryptanalysis

In 1986, Rueppel [117] presented the idea of the "closest linear approximation" to a boolean function $f$. The Walsh transform coefficient of largest magnitude indicates a linear boolean function of the input variables whose output agrees with $f$ more often than any other linear function (this is the closest linear approximation to $f$). Matsui extended this idea by finding the closest linear approximation to a nonzero linear combination of the s-box outputs. In [82] Matsui introduced linear cryptanalysis, which is a known plaintext attack, on DES. The purpose of this method is to obtain a linear expression

$$(a.p) \oplus (b.c) = (d.k),$$

where $p$ is the plaintext, $c$ is the ciphertext, $k$ is the key, and $(a, b, d)$ are the attack parameters. This expression holds with probability $PL \neq \frac{1}{2}$ over all keys, such that $|PL - \frac{1}{2}|$ is maximal for a given cipher algorithm. To achieve this, a statistical linear path between input and output of each s-box is constructed. Then the path is extended to the entire DES algorithm, and finally a linear approximation expression without any intermediate values is reached. The main results of this attack are:

- 8-round DES is breakable with $2^{21}$ known plaintexts in 40 seconds,
- 12-round DES is breakable with $2^{33}$ known plaintexts in 50 hours,
- 16-round DES is breakable with $2^{47}$ known plaintexts (faster than an exhaustive search).

In [82] Matsui reported the first experimental attack of the full 16-round DES with $2^{43}$ known plaintexts, using 12 workstations (HP9735/PA-RISC 99MHZ), and using a program, written in C, and assembly language, implementing the linear cryptanalysis algorithm. The program, consisting of a total of 1000 lines, takes 50 days to find the key of which 40 days were spent for generating plaintexts and the corresponding ciphertexts and only 10 days were spent for the actual key search.

One should note that while linear cryptanalysis can be considered as the most successful theoretical attack on DES, this attack is by no means more practical than exhaustive

---

4 The experiments were implemented with the C-language on a workstation PA-RISC/66MHZ
search: one can not neglect the time needed to obtain the information about the plaintext. Also, the cryptanalyst is free to invest as much, in technology, as he can afford to make the search more efficient when doing exhaustive key search. In a known plaintext attack the cryptanalyst is restricted to the technology of the legitimate owner of the key, and to the frequency with which the key is used. In almost every practical application, a single DES key will be applied to much less than $2^{43}$ blocks, even in its entire life time.

Immunity to linear cryptanalysis can be proved using similar concepts to that used to prove security against differential cryptanalysis [99]. In [92] Nyberg showed how to achieve provable security against linear cryptanalysis.

### 2.3.4 Extensions to Linear and Differential Cryptanalysis

In [74] Langford and Hellman presented a new chosen text attack on iterated cryptosystems. This attack, which is a mixed version of both the differential and linear cryptanalysis attacks, is very efficient for 8-round DES, recovering 10 bits of key with 95% probability of success using only 768 chosen plaintexts. More keys can be recovered with less probability of success. This 8-round attack, while comparable in speed to existing attacks, represents an order of magnitude improvement in the amount of required text. The authors reported that they are currently working to extend this attack to the full 16-round DES.

In [65] Kaliski and Robshaw presented a technique which aids linear cryptanalysis of block ciphers and achieves a reduction of data required for successful attack. The basic idea is to deduce several linear approximations using the same set of plaintext/ciphertext pairs. Finally, the authors noted that the use of larger s-boxes, which is sometimes recommended as a way of increasing the security of DES-like block ciphers, might in certain circumstances facilitate the use of linear cryptanalysis with multiple linear approximations.
In [72] higher order derivatives of discrete functions were considered and the concept of higher order differentials was introduced. The cryptographic relevance of higher order differentials was discussed, but no application was given.

The concept of truncated differentials is introduced in [68] by Knudsen. Truncated differentials are differentials where only a part of the difference in the ciphertext, after a number of rounds, can be predicted. Knudsen also gave examples of Feistel block ciphers secure against differential cryptanalysis using first order differentials, but vulnerable to a differential attack using truncated differentials and higher order differentials.

In [53], linear cryptanalysis is extended to an attack called partition cryptanalysis which considers a partition of the plaintext space and a partition of the last round input space. Partitioning cryptanalysis exploits a potential weakness of the cipher, namely, that the last round inputs are non-uniformly distributed over the blocks of the second partition when the plaintexts are taken from a particular block of the first partition. An example of a cipher for which partition cryptanalysis performs better than linear and differential cryptanalysis was contrived. The success probability of partition cryptanalysis was given and a procedure for finding a pair of partitions that yields a successful attack was analyzed.

### 2.3.5 Other Attacks

Rijmen and Preneel [109] proposed a new attack on Feistel ciphers with a non-surjective round function. CAST and LOKI are examples of such ciphers. They also extended the attack to ciphers that use a non-uniformly distributed round function and applied the attack to a six round weak version of CAST which uses round keys with 16 bit entropy per round.

In [63] Jakobsen and Knudsen introduced a new attack on block ciphers, the interpolation attack. This attack is useful for breaking ciphers using simple algebraic functions (in particular quadratic functions) as s-boxes.
2.3.6 Implementation Dependent Attacks

2.3.6.1 Timing Attacks
This attack was introduced by Kocher in [70]. The basic idea is that cryptosystems often take slightly different amount of time to process different inputs. Reasons include performance optimization to bypass unnecessary operations, branching and conditional statements, RAM cache hits, processor instructions (such as multiplication and division) that run in non-fixed time, and many other causes. Performance characteristics typically depend on both the encryption key and the input data.

By carefully measuring the amount of time required to perform private key operations, attackers may be able to find fixed Diffie-Hellman exponents or factor RSA keys.

Timing attacks can potentially be used against other cryptosystems, including symmetric functions. For example, in software the 28-bit C and D values in DES key schedule are often rotated using a condition which tests whether a one-bit must be wrapped around.

The additional time required to move nonzero bits could slightly degrade the cipher’s throughput or key setup time. The cipher’s performance can thus reveal the Hamming weight of the key, which provides an average of $\sum_{n=0}^{56} \frac{16}{2^6} \log_2 \left( \frac{56}{2^{56}} \right) \approx 3.95$ bits of key information. IDEA uses multiplication modulo $(2^{16} + 1)$ operation, which will usually run in non-constant time. RC5 is at risk on platforms where rotates run in non-constant time. RAM cache hits can produce timing characteristics in many ciphers if tables in memory are not used identically in every encryption.

Additional research is needed to determine whether specific implementations are at risk and, if so, the degree of their vulnerability. So far, only a few specific systems have been studied in detail and the attacks against them are theoretical.

2.3.6.2 Differential Fault Cryptanalysis

This attack, developed by Biham and Shamir, follows the Bellcore fundamental

---

E. Biham and A. Shamir. Research announcement: A new cryptanalytic attack on DES (posted to cypherpunks@toad.com, 18 October 1996)
assumption:

"By exposing a sealed tamperproof device such as a smart card to certain physical effects (e.g., ionizing or microwave radiation), one can induce with reasonable probability a fault at a random bit location in one of the registers at some random intermediate stage in the cryptographic computation. Both the bit location and the round number are unknown to the attacker"

Another assumption is that the attacker is in physical possession of the tamperproof device, so that he can repeat the experiment with the same cleartext and key but without applying the external physical effects. As a result, he obtains two ciphertexts derived from the same (unknown) cleartext and key, where one of the ciphertexts is correct and the other is the result of a computation corrupted by a single bit error during the computation.

The theoretical part of the attack can be applied to DES, triple DES (with 168 bit keys), and DES with independent subkeys (with 768 bit keys). This attack still works even with more general assumptions on the fault locations, such as faults inside the round function, or even faults in the key scheduling algorithm. Differential Fault Analysis can break many additional secret key cryptosystems, including IDEA, RC5 and FEAL. Some ciphers, such as Khufu, Khafre and Blowfish compute their s-boxes from the key material. In such ciphers, it may be even possible to extract the s-boxes themselves, and the keys, using the techniques of Differential Fault Analysis.

In a later research announcement by the same authors, a modified fault model (with some more plausible assumptions) was introduced. This model makes it possible to find the secret key stored in a tamperproof cryptographic device even when nothing is known about the structure and operation of the cryptosystem. A prime example of such a scenario is the Skipjack cryptosystem [1].

---

6 E. Biham and A. Shamir, Research announcement: The next stage of differential fault cryptanalysis: How to break completely unknown cryptosystems. 25 September 1996
The main assumption behind the new fault model is that the cryptographic key is stored in an asymmetric type of memory, in which induced faults are much more likely to change a 1 bit into a 0 than to change a 0 bit into a 1 (or the other way around). The attack is guaranteed to succeed if the fault model is satisfied. A detailed description of the differential fault analysis can be found in [18].

Further Improved Differential Fault Analysis was introduced by Anderson and Kuhn\(^7\).

### 2.4 Advanced Encryption Standard (AES) Requirements


The draft minimum acceptability requirements and evaluation criteria are:

1. AES shall be publicly defined.
2. AES shall be a symmetric block cipher.
3. AES shall be designed so that the key length may be increased as needed.
4. AES shall be implementable in both hardware and software.
5. AES shall either be freely available or available under terms consistent with the American National Standards Institute (ANSI) patent policy.
6. Algorithms which meet the above requirements will be judged based on the following factors:
   1. Security (i.e., the effort required to cryptanalyze),
   2. Computational efficiency,
   3. Memory requirements,
   4. Hardware and software suitability,
   5. Simplicity,

\(^7\) A serious weakness of DES (posted to cypherpunks@toad.com, 2 November 1996)
(6) Flexibility, and

(7) Licensing requirements.

While there are many published block ciphers, none of them meets all the above requirements. Most of the previously published block ciphers have a 64-bit block length. However, the AES is most likely to be a 128-bit block cipher. Also, most of the previously published block ciphers have a fixed key length with no obvious way to increase the key length as needed. Designing an efficient block cipher that meets all the above requirement is a challenging project for the world cryptographic researchers. An issue that was overlooked by the standard requirements is the practical need to satisfy users who require a 64-bit block cipher for compatibility reasons. For more details about the AES development effort the reader is referred to the NIST home page [101].
2.5 Mathematical Background and Definitions

In this section we introduce some of the mathematical tools and definitions used throughout this thesis.

2.5.1 Cryptographic Criteria for Boolean Functions

Nonlinearity

A function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ is defined to be a linear function if $f(x) = a \cdot x$ for some constant $a \in \mathbb{Z}_2^n$.

A function $f$ is defined to be an affine function if $f(x) = a \cdot x + b$ for some constants $a \in \mathbb{Z}_2^n$, $b \in \mathbb{Z}_2$.

The nonlinearity of the function $f = (f_1 f_2 \cdots f_m) : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$, $f_i : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$, $i = 1, \cdots, m$, is defined as the minimum Hamming distance between the set of affine functions and every nonzero linear combination of the output coordinates of $f$, i.e.,

$$\mathcal{NL}_f = \min_{b,c,w} \# \{x \in \mathbb{Z}_2^n | c \cdot f(x) \neq w \cdot x \oplus b \},$$

(2.3)

where $w \in \mathbb{Z}_2^n$, $c \in \mathbb{Z}_2^m \setminus \{0\}$, $b \in \mathbb{Z}_2$, $w \cdot x$ denotes the dot product between $w$ and $x$ over $\mathbb{Z}_2$, and

$$c \cdot f(x) = \bigoplus_{i=1}^{m} c_i f_i(x)$$

(2.4)

where $c = \{c_1, \cdots, c_m\} \in \mathbb{Z}_2^m$.

Linear Structures

A linear structure of a boolean function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2$ can be identified as a vector $a \in \mathbb{Z}_2^n / \{0\}$ such that $f(x \oplus a) \oplus f(x)$ takes the same value (0 or 1) for all $x \in \mathbb{Z}_2^n$ [84].
Output Bit Independence Criterion (BIC)

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ satisfies output Bit Independence Criterion if whenever one bit of input is complemented, the correlation coefficient between every two output bit changes is zero [37].

Balance

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ satisfies the 0–1 balance criterion if it has equal numbers of 0’s and 1’s in its output sequence when the inputs are taken over all possible values.

In general, a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, $n \geq m$, is said to be balanced or regular if each output symbol $y = f(x) \in \mathbb{Z}_2^m$ appears an equal number of times (i.e., for $2^{n-m}$ times) as $x$ varies over all its possible $2^n$ values.

Completeness

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ satisfies the completeness criterion if every output bit of the function depends on all input arguments [66].

Strict Avalanche Criterion (SAC)

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ satisfies SAC if whenever a single input bit is complemented, the output bit changes with a probability of one half [37].

Higher Order SAC (Forré)

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ satisfies SAC of order $k$ if any function obtained from $f$ by keeping $k$ of its input bits constant satisfies SAC [47].

Higher Order SAC (Adams and Tavares)

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ satisfies high order SAC of degree $k$ if $f(x)$ changes with a probability of one half whenever $i$ ($1 \leq i \leq k$) bits of $x$ are complemented [4].
Propagation Criterion (PC)\(^8\)

A function \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) satisfies Propagation Criterion of degree \( k \) (denoted \( PC-k \)) if \( f(x) \) changes with a probability of one half whenever \( i \) \((1 \leq i \leq k)\) bits of \( x \) are complemented.

A function \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) satisfies the extended Propagation Criterion of degree \( k \) and order \( t \) (\( PC-k \) order \( t \)) if any function obtained from \( f \) by keeping \( t \) bits constant satisfies \( PC-k \) [105].

Correlation Immunity

A function \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) is \( k \)th order correlation immune if every \( k \)-tuple obtained by choosing \( k \) components from \( x \) is statistically independent of \( f(x) \) [126].

Binary Bent Functions

A function \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) is bent [116] if and only if

\[
F(w) = \pm 1, \quad w \in \mathbb{Z}_2^n.
\]  

(2.5)

where \( F(w) \) is the Walsh transform [6] of the function \((-1)^{f(x)} \) (See section 2.5.3).

Binary bent functions exist only for even \( n \).

If \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) is bent, then the distance between \( f \) and any affine function is \( 2^{n-1} \pm 2^{n/2-1} \). A bent function attains the maximum achievable nonlinearity (its nonlinearity is \( 2^{n-1} - 2^{n/2-1} \)) and it has the maximum distance to functions with linear structures \( (2^{n-2}) \). If \( f : \mathbb{Z}_2^n \to \mathbb{Z}_2 \) is bent, then we have

\[
\text{wt}(f) = 2^{n-1} \pm 2^{n/2-1}.
\]  

(2.6)

where

\[
\text{wt}(f) = \# \{ x | f(x) \neq 0 \}.
\]  

(2.7)

---

\(^8\) This is identical to Adams and Tavares higher order SAC and was defined independently by Preneel et al [105]
In [90] Nyberg extended the concept of bent functions to multi-output boolean functions. Nyberg presented what she called "perfect nonlinear s-boxes\(^9\). A function \(f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m\) is perfect nonlinear if for every fixed \(\Delta x \in \mathbb{Z}_2^n\) the difference \(f(x \oplus \Delta x) \oplus f(x)\) obtains each possible value \(y \in \mathbb{Z}_2^m\) for \(2^{n-m}\) values of \(x\). Every nonzero linear combination of the output coordinates of perfect non-linear functions is a bent function. The main result is that for perfect nonlinear s-boxes the number of input variables is at least twice the number of output variables.

**Linear Approximation Table**

For a given s-box constructed from a mapping \(f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m\), the linear approximation table entry \(LAT(a, b)\) is defined as [82]:

\[
LAT(a, b) = \#\{x \in \mathbb{Z}_2^n \mid a \cdot x = b \cdot f(x)\} - 2^{n-1}
\]

where \(a \in \mathbb{Z}_2^n\), \(b \in \{\mathbb{Z}_2^m\}/ \emptyset\), and \(a \cdot x\) denotes the inner product of the vectors \(a\) and \(x\) evaluated over \(\mathbb{Z}_2\).

The linear approximation entry with the maximum absolute values is denoted by \(LAT^*\).

**XOR Distribution Table**

For a given s-box constructed from a mapping \(f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m\), the XOR table entry \(N_{\Delta x \Delta y}\) is defined as [15]:

\[
N_{\Delta x \Delta y} = \#\{x \in \mathbb{Z}_2^n \mid f(x \oplus \Delta x) \oplus f(x) = \Delta y\}\quad (2.9)
\]

where \(\Delta x \in \mathbb{Z}_2^n\), \(\Delta y \in \mathbb{Z}_2^m\). The entry \(N_{00} = 2^n\), is not taken into consideration as it does not have any cryptographic significance.

For \(\Delta x \neq 0\), the maximum entry in the XOR table is denoted by \(XOR^*\).

---

\(^9\) Nyberg's definition is more general as it assumes that \(f : \mathbb{Z}_q^n \to \mathbb{Z}_q^m\).
2.5.2 The Inclusion-Exclusion Principle

A counting tool that is used very frequently throughout this thesis is the Inclusion-Exclusion Principle [113]. Different applications of the the Inclusion-Exclusion Principle to cryptography can be found in [98].

Definition

Consider a set of objects, each of which may or may not possess each property from a total set of \( \phi \) properties. Label the properties as \( \rho_i \), \( 1 \leq i \leq \phi \), and define the number of objects possessing the \( k \) properties

\[
\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_k}
\]

as \( N(\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_k}) \).

If \( N(\rho_1, \rho_2, \ldots, \rho_k) = N(\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_k}) \) for all values of \( k, 1 \leq k \leq \phi \), and all selections of \( i_1, i_2, \ldots, i_k \), then the properties are said to be symmetric.

Inclusion-Exclusion Principle:

Consider a set of objects, each of which may or may not possess each property from a total of \( \phi \) properties. If the properties are symmetric, the number of objects having one or more properties, \( \Psi \), is given by [113]

\[
\Psi = \sum_{i=1}^{\phi} (-1)^{i-1} \binom{\phi}{i} \Psi^*(i)
\]  

(2.11)

where \( \Psi^*(i) \) is the number of objects having \( i \) particular properties.
Generalization of Inclusion-Exclusion Principle:
Consider a set of objects, each of which may or may not possess each property from a total of \( \phi \) properties. If the properties are symmetric, the number of objects which possess exactly \( t, 1 \leq t \leq \phi \), properties, \( \Gamma(t) \), is given by [113]

\[
\Gamma(t) = \sum_{i=t}^{\phi} (-1)^{i-t} \binom{i}{t} \Gamma^*(i)
\]

(2.12)

where \( \Gamma^*(i) \) represents the number of objects which have \( i \) particular properties.

2.5.3 The Walsh Transform of Boolean Functions
In this section, we examine some properties of the Walsh transform [6,42] of boolean functions. For a boolean function \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \), the Walsh transform of the function \((-1)^{f(x)}\) is given by\(^{10}\)

\[
F(w) = \frac{1}{2^{n/2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x)}(-1)^{w \cdot x}.
\]

(2.13)

The function \((-1)^{f(x)}\) is given by the inverse Walsh transform as follows

\[
(-1)^{f(x)} = \frac{1}{2^{n/2}} \sum_{w \in \mathbb{Z}_2^n} F(w)(-1)^{w \cdot x}.
\]

(2.14)

Autocorrelation Function
The autocorrelation function \( \Pi_f \) of the function \( f \) is defined as

\[
\Pi_f(\Delta x) = \frac{1}{2^n} \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x \oplus \Delta x) \oplus f(x)}.
\]

(2.15)

The autocorrelation function as defined above can be determined from the Walsh coefficients by

\(^{10}\) From now on, we will refer to \( F(w) \) as the Walsh transform of \( f \).
\[ \Pi_f(\Delta x) = \frac{1}{2^n} \sum_{w \in \mathbb{Z}_2^n} F^2(w)(-1)^{\Delta x \cdot w} . \] (2.16)

**Summation Property**

The Walsh transform of a boolean function \( f \) satisfies

\[ \sum_{w \in \mathbb{Z}_2^n} F(w) = 2^{n/2}(-1)^{f(0)} . \] (2.17)

**Parseval's Theorem**

\[ \sum_{w \in \mathbb{Z}_2^n} F^2(w) = 2^n . \] (2.18)

**Dyadic Shift**

Let \( F_1(w) = F(w \oplus a) \) then \( f_1(x) \) is given by

\[ f_1(x) = f(x) \oplus a \cdot x . \] (2.19)

Similarly, if \( f_2(x) = f(x \oplus a) \) then

\[ F_2(w) = F(w)(-1)^{a \cdot w} . \] (2.20)

**Dyadic Convolution**

Let \( f(x) = f_1(x) \oplus f_2(x) \) then

\[ F(w) = \frac{1}{2^{n/2}} \sum_{u \in \mathbb{Z}_2^n} F_1(u) F_2(w \oplus u) . \] (2.21)

From the above result it can be shown that the Hamming distance

\[ d(f_1, f_2) \overset{\text{def}}{=} \# \{ x | f_1(x) \neq f_2(x) \} \] (2.22)

can be expressed as

\[ d(f_1, f_2) = 2^{n-1} - \frac{1}{2} \sum_{w \in \mathbb{Z}_2^n} F_1(w) F_2(w) . \] (2.23)

The Walsh transform is a powerful analytical tool for studying the properties of boolean functions for different reasons:
Some cryptographic properties, such as bentness, were originally defined in terms of the Walsh spectrum of the boolean functions.

All the cryptographic properties of a boolean function can be extracted from its Walsh spectrum. For example:

- The largest spectral component gives the nonlinearity of the function \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \)

\[
\mathcal{NL}_f = 2^{n-1} - 2^{n/2-1} \max_w |F(w)|
\]

(2.24)

Similarly, for a multi-output boolean function \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m \) we have

\[
\mathcal{NL}_f = 2^{n-1} - 2^{n/2-1} \max_{c \neq 0, w} |F_c(w)|,
\]

where \( F_c(w) \) is the Walsh transform of the function \((-1)^{cf} \).

- \( K \)th order correlation immune functions must have

\[
F(w) = 0, \quad 1 \leq wt(w) \leq k
\]

(2.26)

- SAC fulfilling functions must have

\[
\sum_{w \in \mathbb{Z}_2^n} F^2(w)(-1)^{w_i} = 0, \quad i \in \{1, 2, \ldots n\},
\]

(2.27)

where \( w_i \) is the i-th bit in \( w \).

- Functions satisfying PC-K must have

\[
\sum_{w} F^2(w)(-1)^{a \cdot w} = 0, \quad 1 \leq wt(a) \leq k.
\]

(2.28)

The field of digital signal processing is well established and it can provide us with a good theoretical foundation for studying different cryptographic characteristics of multi-output boolean functions.

Possible extension of the Walsh transform to the case of multi-output boolean functions: multi-output boolean functions can be described by the Walsh transform of every nonzero linear combination of its output coordinates.

This motivates us to express different forms of information leakage in terms of the Walsh transform coefficients (see section 5.6).
Chapter 3 Linear Approximation Table and XOR Distribution Table of Balanced S-boxes

Differential cryptanalysis [15], and linear cryptanalysis [82] are currently the most powerful cryptanalytic attacks on private-key block ciphers. The complexity of differential cryptanalysis depends on the size of the largest entry in the XOR table, the total number of zeroes in the XOR table, and the number of nonzero entries in the first column in that table [15], [122]. The complexity of linear cryptanalysis depends on the size of the largest entry in the linear approximation table (LAT) [81], [82].

Thus one way to reduce the risk of differential and linear cryptanalysis is to choose the s-boxes with small maximum LAT entries and small maximum XOR table entries.

O’Connor [97] studied the distribution of the XOR table of bijective mappings. In particular, O’Connor showed that for a randomly selected $n$-bit bijective mapping, the expected size of the largest entry in the XOR distribution table is bounded by $2^m$ while the fraction of the XOR table that is zero approaches $e^{-0.5} = 0.60653$. O’Connor also showed that the above two quantities are tending to optimal values in composite permutations [96].

In some block cipher designs, it is required to have balanced s-boxes (also known as a regular s-box). In this chapter, we derive an upper bound on the fraction of balanced functions (s-boxes) having a specified lower bound on the maximum entry in the XOR distribution table or the LAT. For reasonably small values of these maximum entries we show that this fraction decreases dramatically with the number of input variables.

We also calculate the probability that any nonzero linear combination of the output coordinates of a regular s-box is affine.

From [88], the total number of balanced s-boxes with $n$ input bits, and $m$ output bits
is given by

\[ B(n, m) = \frac{2^n!}{(2^{n-m})^m} \quad n \geq m. \quad (3.1) \]

### 3.2 Linear Approximation Table of Balanced S-boxes

Recall that for a given s-box constructed from a mapping \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m \), the linear approximation table entry \( LAT(a, b) \) is defined as [82]:

\[ LAT(a, b) = \#\{x \in \mathbb{Z}_2^n \mid a \cdot x = b \cdot f(x)\} - 2^{n-1} \]

where \( a \in \mathbb{Z}_2^n \), \( b \in \{\mathbb{Z}_2^m\}/0 \), and \( a \cdot x \) denotes the inner product of the vectors \( a \) and \( x \) evaluated over \( \mathbb{Z}_2 \).

It is straightforward to notice that \( LAT(a, b) = 2^{n-1} - d(a \cdot x, b \cdot f) \), where

\[ d(a \cdot x, b \cdot f) = \#\{x \in \mathbb{Z}_2^n \mid a \cdot x \oplus b \cdot f(x) = 1\}. \quad (3.3) \]

Let \( wt(a) \) be the Hamming weight of the binary vector \( a \); which is the number of 1’s in the vector \( a \). For \( wt(a) = 0 \), we have \( LAT(a, b) = 0 \) as \( b \cdot f \) is always a balanced function for a balanced s-box.

For \( a \neq 0, b \neq 0 \) we have

**Lemma 3.1**

\[ P\{d(a \cdot x, b \cdot f) = 2l\} = \frac{\left( \frac{2^{n-1}}{1} \right)^2 (2^{n-1})^2}{2^n!}. \quad (3.4) \]

**Proof:** For \( wt(b) = k > 0 \), we have \( \left( \frac{2^{n-1}}{(2^{n-1})^k} \right)^2 \) ways of arranging the bits of \( f(x) \) corresponding to nonzero bits of \( b \) such that \( d(a \cdot x, b \cdot f) = 0 \). This result follows directly by noting that the number of arrangements of \( r \) different objects, for each of which there are \( c \) copies, is \( (rc)!/(c!)^r \). In our case, we have \( 2^{k-1} \) \( k \)-bit symbols with the corresponding XOR equal 0, and \( 2^{k-1} \) \( k \)-bit symbols with the corresponding XOR.
equal 1. Each of these symbols occurs $2^{n-k}$ times. Thus we have \( \left( \frac{2^{n-k} \cdot 1}{(2^{n-k})^2} \right) \) possible arrangements for the symbols corresponding to the 0's of \( a \cdot x \) and a similar number of possible arrangements for the symbols corresponding to the 1's of \( a \cdot x \).

Since we can partition the input \( x \) into \( 2^k \) distinct sets, all \( x \)'s in a given set are assigned the same common value of the \( k \) bits of \( f(x) \) corresponding to the nonzero bits of \( b \).

We still need to assign the remaining \( m - k \) output bits for each \( x \). Each \( x \) within a given set must be assigned a distinct \( m - k \) tuple of the remaining output bits for \( 2^{n-m} \) times, each set can be assigned in \( \frac{2^{n-k}}{(2^{n-m})^2} \) ways, so the remaining bits can be assigned in \( \left( \frac{2^{n-k}}{(2^{n-m})^2} \right)^{2^k} \) ways.

Thus we have \( \left( \frac{2^{n-1}}{(2^{n-k})^{2^k}} \right)^2 \left( \frac{2^{n-k}}{(2^{n-m})^{2^m}} \right)^{2^k} = \frac{(2^{n-1})^2}{(2^{n-m})^{2^m}} \) distinct balanced functions with \( d(a \cdot x, b \cdot f) = 0 \). Using the same argument, and by noting that we have \( \left( \frac{2^{n-1}}{l} \right)^2 \) ways to generate balanced functions that have \( d(a \cdot x, b \cdot f) = 2l \) from \( a \cdot x \), we have \( \left( \frac{2^{n-1}}{l} \right)^2 \left( \frac{2^{n-1}}{l} \right)^2 \) ways of generating distinct balanced s-boxes with \( d(a \cdot x, b \cdot f) = 2l \).

By dividing by the total number of balanced s-boxes, \( B(n, m) \), we get the lemma above.

**Theorem 3.1**

For any integer value \( M_{LAT} \), \( 0 \leq M_{LAT} \leq 2^{n-2} \), let \( N_{LAT}^* \) denote the number of LATs with any entry having absolute value \( \geq 2M_{LAT} \), then \( N_{LAT}^* \) is upper bounded by

\[
\frac{N_{LAT}^*}{B(n, m)} < \frac{2(2^{n-1})^2(2^n - 1)(2^m - 1)}{2^{n!}} \sum_{l=M_{LAT}}^{2^{n-2}} \left( \frac{2^{n-1}}{2^{n-2} + l} \right)^2.
\]  

(3.5)

**Proof:** For \( l \neq 0 \) we have

\[
P\{LAT(a, b) = \pm 2l\} = 2P\{d(a \cdot x, b \cdot f) = 2^{n-1} + 2l\}.
\]  

(3.6)
Using lemma 3.1, and by noting that

\[
\frac{N_{LAT}^*}{B(n,m)} = P \left\{ \left( \max_{a,b \neq 0} |LAT(a,b)| \right) \geq 2M_{LAT} \right\} \\
\leq \sum_{a,b \neq 0} P \{ |LAT(a,b)| \geq 2M_{LAT} \}.
\]

(3.7)

we get the theorem above.

Numerical substitution in the formula above shows that the fraction of balanced s-boxes with undesirable LATs decreases dramatically as the number of inputs increases. To give a numerical example, consider the case where \( M_{LAT} = 2^{n-5} \), for \( n = 12, m = 6 \) we have \( \frac{N_{LAT}^*}{B(12,6)} < 3.86 \times 10^{-10} \).

### 3.3 XOR Distribution Table of Balanced S-boxes

Recall that for a given s-box constructed from a mapping \( f : \mathbb{Z}_2^2 \rightarrow \mathbb{Z}_2^n \), the XOR table entry \( N_{\Delta x \Delta y} \) is defined as [15]:

\[
N_{\Delta x \Delta y} = \# \{ x \in \mathbb{Z}_2^n \mid f(x \oplus \Delta x) \oplus f(x) = \Delta y \}
\]  

(3.8)

where \( \Delta x \in \mathbb{Z}_2^n, \Delta y \in \mathbb{Z}_2^m \).

**Lemma 3.2**

For \( \Delta x \neq 0, \Delta y \neq 0 \), the number of balanced functions with \( N_{\Delta x \Delta y} \geq 2k \) is upper bounded by \( \Psi_{n,m}(k) \), where

\[
\Psi_{n,m}(k) = \binom{2^n-1}{k} 2^k \sum_{\sum k_i = k} G(k_1, k_2, ..., k_{2m-1})
\]

(3.9)

\[
G(k_1, k_2, ..., k_{2m-1}) = C(k; k_1, k_2, ..., k_{2m-1}) \frac{2^n - 2k; l_1, l_2, ..., l_{2m-1}, l_{2m-1}}{k_1! ... k_{2m-1}!}, \text{ and } l_i = 2^n - m - k_i, l_i \geq 0, k_i \geq 0.
\]

Equation (3.7) simply means that the number of LATs with entries having absolute value \( \geq 2M_{LAT} \) is upper bounded by the number of these entries.
**Proof:** For any balanced function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$, $n \geq m$, we have $2^m$ distinct output symbols, $y_0, y_1, \ldots, y_{2^m-1}$, and each of them is repeated $2^{n-m}$ times. For a given $\Delta y \neq 0, \Delta x \neq 0$, let $S$ be the set of $2^{n-1}$ output symbols pairs $(y_i, y_j)$ with $y_i = y_j = \Delta y$. Thus $S$ has $2^{m-1}$ distinct unordered pairs, each of them is repeated $2^{n-m}$ times. Divide $S$ into $2^{m-1}$ subsets such that each of these subsets contains only identical unordered pairs. Let $S_1, S_2, \ldots, S_{2^m-1}$ denote these subsets.

Now we count the number of ways by which we can construct a balanced function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ for which $N_{\Delta x \Delta y} \geq 2k$. Select $k_i$ pairs from each subset $S_i$, $i = 1, 2, \ldots, 2^{m-1}$ such that $\sum_{i=1}^{2^{m-1}} k_i = k$. There is only one way to choose $k_i$ pairs from the subset $S_i$ (as the pairs within $S_i$ are indistinguishable). These $k$ pairs can be ordered in $C(k; k_1, k_2, \ldots, k_{2^m-1})$ ways. Note that we have 2 possible orders for each pair, giving $2^k$ total possible orders.

Now we select $k$ different input values $x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}$, for which $x_{i_0} \oplus x_{i_j} \neq \Delta x$ for $0 \leq i, j < k$. There are $\binom{2^{n-1}}{k}$ possible choices for $x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}$. For each such choice we assign to each pair $(f(x_{i_0}), f(x_{i_1} \oplus \Delta x))$ an element from the $k$ pairs chosen above.

The remaining $2^n - 2k$ output symbols of $f$ can be assigned in $C(2^n - 2k; l_1, l_1, l_2, l_2, \ldots, l_{2^m-1}, l_{2^m-1})$ ways, where $l_i = 2^{n-m} - k_i$.

The construction approach described above does not guarantee the construction of distinct balanced functions, and so $\Psi_{n,m}(k)$ is an upper bound. \qed
Lemma 3.3

For \( \Delta x \neq 0, \Delta y = 0 \), the number of balanced functions with \( N_{\Delta x_0} \geq 2k \) is upper bounded by \( \Phi_{n,m}(k) \), where

\[
\Phi_{n,m}(k) = \binom{2^{n-1}}{k} \sum_{k_i=k} D(k_1, k_2, \ldots, k_{2m}) .
\]  

(3.11)

\[
D(k_1, k_2, \ldots, k_{2m-1}) = C(k; k_1, k_2, \ldots, k_{2m}) \cdot C(2^n - 2k; l_1, l_2, \ldots, l_{2m}) .
\]  

(3.12)

and \( l_i = 2^{n-m} - 2k_i, \ l_i \geq 0, \ k_i \geq 0 \).

Proof: For any balanced function \( f : Z_2^n \rightarrow Z_2^m, n \geq m \), we have \( 2^m \) distinct output symbols, \( y_0, y_1, \ldots, y_{2^m-1} \), and each of them is repeated \( 2^{n-m} \) times. For a given \( \Delta x \neq 0, \Delta y = 0 \), let \( S \) be the set of \( 2^{n-1} \) output symbols pairs \((y_i, y_j)\) with \( y_i = y_j \). Thus \( S \) has \( 2^m \) distinct pairs, each of them is repeated \( 2^{n-m-1} \) times. Divide \( S \) into \( 2^m \) subsets such that each of these subsets contains only identical pairs. Let \( S_1, S_2, \ldots, S_{2m} \) denote these subsets.

Now we count the number of ways by which we can construct a balanced function \( f : Z_2^n \rightarrow Z_2^m \) for which \( N_{\Delta x_0} \geq 2k \). Select \( k_i \) pairs from each subset \( S_i, i = 1, 2, \ldots, 2^m \) such that \( \sum_{i=1}^{2^m} k_i = k \). There is only one way to choose \( k_i \) pairs from \( S_i \) (as the symbols within \( S_i \) are indistinguishable). These \( k \) pairs can be ordered in \( C(k; k_1, k_2, \ldots, k_{2m}) \) ways. In this case, we do not have the \( 2^k \) factor (as in Lemma 3.2) because the two elements within each pair are identical. Now we select \( k \) different input values \( x_{l_0}, x_{l_1}, \ldots, x_{l_{k-1}} \), for which \( x_{l_i} \oplus x_{l_j} \neq \Delta x \) for \( 0 \leq i, j < k \). There are \( \binom{2^{n-1}}{k} \) possible choices for \( x_{l_0}, x_{l_1}, \ldots, x_{l_{k-1}} \). For each such choice we assign to each pair \((f(x_{l_i}), f(x_{l_j} \oplus \Delta x))\) an element from the \( k \) \( y \) pairs chosen above. The remaining \( 2^n - 2k \) output symbols of \( f \) can be assigned in \( C(2^n - 2k; l_1, l_2, \ldots, l_{2m}) \) ways where \( l_i = 2^{n-m} - 2k_i \).

A similar statement about the upper bounding applies to \( \Phi_{n,m}(k) \). \( \square \)
Lemma 3.4
The exact number of balanced functions with $N_{\Delta x \Delta y} = 2k, \Delta x \neq 0, \Delta y \neq 0$ (denoted by $\Lambda_{n,m,\Delta y}(k)$) is given by

$$\Lambda_{n,m,\Delta y}(k) = \sum_{i=k}^{2^{n-1}} (-1)^{i-k} \binom{i}{k} \Psi_{n,m}(i). \quad (3.13)$$

Proof: Let $\rho_{ik}$ (cf. equation (2.10)) denote the number of balanced functions with the property that $N_{\Delta x \Delta y} \geq 2k$. The Lemma follows by direct application of the Inclusion-Exclusion Principle.\[\square\]

Similarly, if we denote the exact number of balanced functions with $N_{\Delta x \Delta y} = 2k, \Delta x \neq 0, \Delta y = 0$ by $\Lambda_{n,m,0}(k)$ then we have

$$\Lambda_{n,m,0}(k) = \sum_{i=k}^{2^{n-1}} (-1)^{i-k} \binom{i}{k} \Phi_{n,m}(i). \quad (3.14)$$

Using the above results, and by noting that

$$\frac{N_{XOR}^*}{B(n,m)} = P\left\{ \left( \max_{\Delta x \neq 0, \Delta y} N_{\Delta x \Delta y} \right) \geq 2M_{XOR} \right\}$$

$$\leq \sum_{\Delta x \neq 0, \Delta y} P\{N_{\Delta x \Delta y} \geq 2M_{XOR}\}, \quad (3.15)$$

we get the following theorem.

---

2. Equation (3.15) simply means that the number of the XOR distribution tables with entries having value $\geq 2M_{XOR}$ is upper bounded by the number of these entries.
The fraction of balanced functions with maximum XOR table entry $\geq 2^{M_{\text{XOR}}}$, $0 \leq M_{\text{XOR}} \leq 2^{n-1}$, is upper bounded by

$$\frac{N^*_{\text{XOR}}}{B(n, m)} \leq \frac{(2^n - 1)(2^m - 1)}{B(n, m)} \sum_{k = M_{\text{XOR}}}^{2^{n-1}} \Lambda_{n, m, \Delta y}(k)$$

$$+ \frac{(2^n - 1)}{B(n, m)} \sum_{k = M_{\text{XOR}}}^{2^{n-1}} \Lambda_{n, m, 0}(k). \quad (3.16)$$

Numerical substitution in the formula above shows that the fraction of balanced s-boxes with undesirable XOR distribution tables decreases dramatically as the number of inputs increases.

One special case of interest is for bijective mappings, i.e., when $n = m$. For this case (see [97] for more details on this special case) we have

$$\frac{N^*_{\text{LAT}}}{B(n, n)} \leq \frac{2(2^n - 1)^2}{2^{n!}} \sum_{l = M_{\text{LAT}}}^{2^{n-2}} \left( \frac{2^n - 1}{2^{n-2} + l} \right)^2. \quad (3.17)$$

It is also straightforward to see that

$$\psi_{n,n}(k) = \left( \frac{2^n - 1}{k!} \right)^2 2^k (2^n - 2k)! \cdot k!, \quad (3.18)$$

and

$$\phi_{n,n}(k) = \begin{cases} \frac{2^n}{(2^n - m)2^m}, & k = 0, \\ 0, & k > 0, \end{cases} \quad (3.19)$$

and hence for $0 < M_{\text{XOR}} \leq 2^{n-1}$, we have

$$\frac{N^*_{\text{XOR}}}{B(n, n)} \leq \frac{(2^n - 1)^2}{2^{n!}} \sum_{k = M_{\text{XOR}}}^{2^{n-1}} \Lambda_{n, n, \Delta y}(k). \quad (3.20)$$
To give a numerical example, consider the case where $M_{XOR} = n$. For $n = 12$ we have $\frac{N_{XOR}}{B(n,n)} < 8.9 \times 10^{-6}$.

Table 3.1 shows the results of our simulation$^3$ for $10^4$ randomly chosen 8-bits bijective mappings. T_Bound refers to our theoretical bound (equations (3.17) and (3.20) ) for the fraction of bijective mappings with $XOR^* = 2Max_{XOR}$, or $LAT^* = 2Max_{LAT}$. S_Bound refers to the bounds obtained from the simulation results. Only the nontrivial bounds are shown in the table.

Table 3.2 shows the number of bijective mappings with $XOR^* = 2Max_{XOR}$, and $LAT^* = 2Max_{LAT}$ obtained from our simulations.

<table>
<thead>
<tr>
<th>$2Max_{XOR}$</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>4066</td>
<td>5307</td>
<td>591</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>T_Bound</td>
<td>—</td>
<td>—</td>
<td>0.1106</td>
<td>0.0069</td>
<td>0.0004</td>
</tr>
<tr>
<td>S_Bound</td>
<td>1.0</td>
<td>0.5934</td>
<td>0.0627</td>
<td>0.0036</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$2Max_{LAT}$</th>
<th>30</th>
<th>32</th>
<th>34</th>
<th>36</th>
<th>38</th>
<th>40</th>
<th>42</th>
<th>44</th>
<th>46</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>19</td>
<td>1145</td>
<td>3842</td>
<td>3083</td>
<td>1333</td>
<td>442</td>
<td>110</td>
<td>23</td>
<td>3</td>
</tr>
<tr>
<td>T_Bound</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>0.6990</td>
<td>0.2091</td>
<td>0.0585</td>
<td>0.0153</td>
<td>0.0038</td>
<td>0.0009</td>
</tr>
<tr>
<td>S_Bound</td>
<td>1.0</td>
<td>0.9981</td>
<td>0.8836</td>
<td>0.4994</td>
<td>0.1911</td>
<td>0.0578</td>
<td>0.0136</td>
<td>0.0026</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

Table 3.1 Experimental Results versus Theoretical Bounds for $10^4$ Randomly Chosen 8-bit Bijective Mappings

---

$^3$ Simulation results in this thesis make use of the UNIX C library function "rand"
<table>
<thead>
<tr>
<th>$2^{\text{MaxLAT}}$</th>
<th>30</th>
<th>32</th>
<th>34</th>
<th>36</th>
<th>38</th>
<th>40</th>
<th>42</th>
<th>44</th>
<th>46</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{\text{MaxXOR}}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>506</td>
<td>1606</td>
<td>1232</td>
<td>495</td>
<td>177</td>
<td>36</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>583</td>
<td>2007</td>
<td>1653</td>
<td>729</td>
<td>237</td>
<td>69</td>
<td>15</td>
<td>—</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>52</td>
<td>217</td>
<td>189</td>
<td>99</td>
<td>27</td>
<td>5</td>
<td>2</td>
<td>—</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>4</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 3.2 Distribution of the Maximum XOR Table Entry and the Maximum LAT Entry (Experimental Results for $10^4$ Randomly Chosen 8-bit Bijective Mappings)

### 3.4 Counting the Number of Nonlinear Balanced S-boxes

Gordon and Retkin [50] calculated the probability that any of the output coordinates of a random, reversible substitution box (i.e., a bijective mapping) is an affine function. After linear cryptanalysis was introduced, it was realized that the cryptographic strength of a multi-output function depends not only on the strength of its individual output coordinates but also on the strength of every nonzero linear combination of these coordinates [90].

In this section, we calculate the probability that any nonzero linear combination of the output coordinates of a regular s-box is an affine function.

**Lemma 3.5**

The number of regular $n \times m$ s-boxes (described by the multi-output boolean functions $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$, $n \geq m$) for which the first $k$ output functions are affine is given by

$$R(n, m, k) = 2^k \left( \frac{2^n - 1}{(2^{n-m})^{2^{m-k}}} \right)^2 \prod_{i=0}^{k-1} (2^n - 2^i).$$ \hspace{1cm} (3.21)

**Proof:** By noting that every nonzero linear combination of the output coordinates of regular s-boxes is a balanced function, the first function can be chosen in $(2^{n+1} - 2)$ ways, which is the total number of balanced affine functions. The second function can be chosen from the set of balanced affine functions not including the first one or its
complement, i.e., in \((2^{n+1} - 4)\) ways. The third one can be chosen from the set of balanced affine functions not including any linear combination of the first two functions. Proceeding as above, the first \(k\) output functions can be chosen in \(2^k \prod_{i=0}^{k-1} (2^n - 2^i)\) ways. Since we can partition the input \(x\) into \(2^k\) distinct sets, all \(x\)'s in a given set are assigned the same common value of these \(k\) bits. We still need to assign the remaining \(m - k\) output bits for each \(x\). Each \(x\) within a given set must be assigned a distinct \(m - k\) tuple of the remaining output bits for \(2^{n-m}\) times, each set can be assigned in \(\frac{2^{n-k+1}}{(2^{n-m})^{2^m-k}}\) ways, so the remaining bits can be assigned in \(\left(\frac{2^{n-k+1}}{(2^{n-m})^{2^m-k}}\right)^{2^k}\) ways. \(\square\)

Consider the function \(\Phi : Z_2^n \rightarrow Z_2^{2^m-1}\) constructed from every nonzero linear combination of the output coordinates of \(f\). Counting the number of regular s-boxes with \(\mathcal{N} \mathcal{L} \neq 0\) corresponds to counting the number of functions \(\Phi\) with no affine coordinates. The number of ways one can choose \(l\) coordinates of \(\Phi\) such that \(k\) of them are linearly independent is equivalent to the number of \(l \times m\) binary matrices (without taking the order of rows into account, i.e., two matrices with the same rows but in different orders are counted once) with nonzero distinct rows which have rank \(k\). This is given by \([132],[49]\)

\[
LI(m, l, k) = \left[ \begin{array}{c} m \\ k \end{array} \right]_2 \sum_{j=0}^{k} (-1)^j \binom{2^{l-j}}{l} \left(\frac{2^{(k-j)} - 1}{l} \right) \left[ \begin{array}{c} k \\ j \end{array} \right]_2.
\]  

(3.22)

where

\[
\left[ \begin{array}{c} m \\ k \end{array} \right]_2 = \left\{ \begin{array}{ll}
1, & k = 0, \\
\prod_{i=0}^{k-1} \frac{(2^m - 2^i)}{(2^k - 2^i)}, & n \geq k > 0.
\end{array} \right.
\]  

(3.23)

**Lemma 3.6**

The number of ways to construct certain \(k\) linearly independent coordinates of \(\Phi\) from affine functions is also given by \(R(n, m, k)\).
Proof: It is clear that for every $k$ linearly independent coordinates of $\Phi$, denoted by $\phi_{i_1}, \phi_{i_2}, \ldots, \phi_{i_k}$, one can find $(m - k)$ coordinates of $\Phi$, denoted by $\phi_{i_{k+1}}, \phi_{i_{k+2}}, \ldots, \phi_{i_m}$, such that

$$
(\phi_{i_1}, \phi_{i_2} \ldots \phi_{i_m})^t = A (f_1, f_2 \ldots f_m)^t
$$

(3.24)

where $A$ is an $m \times m$ invertible binary matrix, $(f_1, f_2 \ldots f_m)$ denotes the output coordinates of $f$. This means that as $f$ varies over all the set of distinct regular s-boxes, $(\phi_{i_1}, \phi_{i_2} \ldots \phi_{i_m})$ scans the whole set but in a different order.

From the lemma above it follows that the number of ways to construct $l$ coordinates of $\Phi$ from affine functions is given by

$$
\sum_{k=1}^{\min(l, m)} LI(m, l, k) R(n, m, k).
$$

(3.25)

Using the inclusion-exclusion principle, the number of regular s-boxes with the property that one or more of the nonzero linear combinations of its output coordinates are affine, $RL(n, m)$, is given by

Theorem 3.3

$$
RL(n, m) = \sum_{l=1}^{2^{m-1}} (-1)^{l-1} \sum_{k=1}^{\min(l, m)} LI(m, l, k) R(n, m, k).
$$

(3.26)

To express the above count as a fraction of the total number of regular s-boxes, denoted by $FRL(n, m)$, we divide by the total number of $n \times m$ regular s-boxes.

To give a numerical example, for $n = 6, m = 4$, which is the size of DES s-boxes, $FRL(6, 4) = 2.46 \times 10^{-16}$. One can easily get an upper bound for $FRL(n, m)$ by noting that we have

$$
RL(n, m) < (2^m - 1) R(n, m, 1)
$$

(3.27)
and hence we have

\[ FRL(n, m) < \frac{2(2^n - 1)(2^m - 1)(2^{n-1})^2}{2^n!}. \]  

(3.28)

Since we have \( n \geq m \), then

\[ FRL(n, m) < \frac{2(2^n - 1)^2(2^{n-1})^2}{2^n!}. \]  

(3.29)

By noting that

\[ \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e\left(\frac{1}{12n+\frac{1}{2}}\right) < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e\left(\frac{1}{12n}\right), \]  

(3.30)

then we have

\[ FRL(n, m) < \sqrt{2\pi n}2^{3n/2+m}\frac{1}{2^n}e^{\left(\frac{1}{12n} - \frac{1}{12(2^{n+1})}\right)} \]

\[ = O\left(\frac{2^{3n/2}}{2^{2n}}\right), \]  

(3.31)

where \( O(\cdot) \) is the asymptotic upper bound order notation\(^4\).

### 3.5 Conclusion

In this chapter we derived an upper bound on the fraction of balanced functions (s-boxes) having a specified lower bound on the maximum entry in the XOR distribution table or the LAT. For reasonably small values of these maximum entries we showed that this fraction decreases dramatically with the number of input variables.

We also calculated the probability that any nonzero linear combination of the output coordinates of a regular s-box is affine. Our results shows that the number of balanced functions with nonzero nonlinearity goes down exponentially (as \( 2^{2^n-5n/2} \)) with the number of input variables \( n \).

While these results suggest that large cryptographically strong balanced s-boxes may be obtained by selecting them at random, these statistical results might have some practical

\[^4\ h(n) = O(g(n)) \text{ if there exists a positive constant } c \text{ and a positive integer } n_0 \text{ such that } 0 \leq h(n) \leq cg(n) \text{ for all } n \geq n_0.\]
limitations. Describing a large randomly chosen s-box requires a large amount of memory which might be impractical in some applications. This suggests that other methods such as algebraic constructions, which trade-off memory requirements and computational speed (see [89] for an example of such methods) might offer alternative approaches.
Chapter 4 Linear Approximation of Injective S-boxes

One way to reduce the size of the largest entry in the XOR table, and hence reduce the risk of differential cryptanalysis, is to use injective substitution boxes (s-boxes) such that the number of output bits of the s-box is sufficiently larger than the number of input bits. In this way, it is very likely that the entries in the XOR distribution table of a randomly chosen injective s-box will have only small values, making the block cipher resistant to differential cryptanalysis. Some proposed block ciphers, such as CAST [5] and Blowfish [119], take advantage of this property.

On the other hand, Biham [14] proved that if for an \( n \times m \) s-box described by \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m \) we have \( m \geq 2^n - n \), then at least one linear combination of the output bits must be an affine combination of the input bits and the block cipher can be trivially broken by linear cryptanalysis.

In this chapter, we estimate the size of the largest entry in the LAT randomly selected injective s-box. We also present two methods for constructing highly nonlinear s-boxes. Finally we show how the resistance of a CAST-like encryption algorithm to the basic linear cryptanalysis was underestimated in [75].

4.1 Definitions and Notation

The function \( g : X \rightarrow Y \) is injective (or one-to-one) if for \( x, \tilde{x} \) in \( X \) the equality \( g(x) = g(\tilde{x}) \) implies \( x = \tilde{x} \). That is, distinct elements of \( X \) can not have the same image in \( Y \) [62].

From the definitions of nonlinearity and LAT, it is clear that the nonlinearity of the function \( f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m \) is given by

\[
\mathcal{N}\mathcal{L}_f = 2^{n-1} - \max_{a,b} |LAT(a,b)|,
\]

where \( a \in \mathbb{Z}_2^n \), \( b \in \{\mathbb{Z}_2^m\}/0 \).
Throughout the rest of this chapter, the $b = 0$ case is not taken into consideration as it does not have any cryptographic significance.

It is easy to see that the number of injective $n \times m$ s-boxes is given by

$$\text{In}(n,m) = \prod_{i=0}^{(2^n-1)} (2^m - i).$$

(4.2)

Throughout this chapter, let

$f : Z^n_2 \to Z^m_2$ describe a random injective mapping,

$\text{wt}(a)$ denote the Hamming weight of the binary vector $a$,

$$\binom{i}{j} = \begin{cases} \frac{i!}{j!(i-j)!}, & i, j \text{ integers, } i \geq j \\ 0, & \text{otherwise,} \end{cases}$$

$d(a \cdot x, b \cdot f) = \# \{x \in Z^n_2 | a \cdot x \neq b \cdot f(x)\}$,

$$P_w(k) = P\{\text{wt}(b \cdot f) = k\},$$

$$P_d(a,b)(l) = P\{d(a \cdot x, b \cdot f) = l\}, \quad b \neq 0,$$

$$P_{d|w}(l, k) = P\{d(a \cdot x, b \cdot f) = l \mid \text{wt}(b \cdot f) = k\}.$$ 

### 4.2 Linear Approximation Table of Injective Mappings

We first determine the probability distribution of the weight of $b \cdot f$:

**Lemma 4.1**

$$P_w(k) = \binom{2^n-1}{k} \binom{2^m-1}{2^m-k}. \tag{4.4}$$

**Proof:** For $b \neq 0$, $\text{wt}(b \cdot f)$ follows the hypergeometric distribution [20]. This follows by noting that the weight of $b \cdot f$ has the same distribution as that of the function constructed by randomly choosing $2^n$ bits from the function $b \cdot \pi$ where $\pi : Z^m_2 \to Z^n_2$ is an arbitrary bijective mapping. 

44
Given the weight of $b \cdot f$, the distribution of the distance between $a \cdot x$ and $b \cdot f$ can be determined:

**Lemma 4.2**

For $a \neq 0$, we have

$$ P_{d\mid w}(l, k) = \frac{1}{\binom{2n}{k}} \left( \frac{2^{n-1}}{\frac{2n-1+k-l}{2}} \right) \left( \frac{2^{n-1}}{\frac{k-2n-1+l}{2}} \right). \quad (4.5) $$

**Proof:** Let $M_{ij} = \# \{ x \in \mathbb{Z}_2^n \mid a \cdot x = i, b \cdot f(x) = j \}, \quad i, j \in \mathbb{Z}_2$. Then we have $d(a \cdot x, b \cdot f) = M_{10} + M_{01}$. Since $wt(b \cdot f) = k$, we have $M_{01} + M_{11} = k$. We also have $M_{10} + M_{11} = 2^n - 1$ as $a \cdot x$ is a balanced function. Using these equations,

$$ P_{d\mid w}(l, k) = P_{M_{11}\mid w} \left( \frac{2^{n-1} + k - l}{2}, k \right). \quad (4.6) $$

where

$$ P_{M_{11}\mid w}(j, k) = \binom{2^{n-1}}{j} \binom{2^{n-1}}{k-j}, \quad (4.7) $$

we get the lemma. \qed

Combining the two results above, the following theorem gives the probability distribution of the distance between $a \cdot x$ and $b \cdot f$ for fixed $a$ and $b$. This can be used to find the probability that a given entry in the LAT has a particular value, $2^n - 1$.

**Theorem 4.1**

$$ P_d^{(a, b)}(l) = \begin{cases} P_w(l), & a = 0, \\ \sum_{k=0}^{2^n} \frac{P_w(k)}{\binom{2^n}{k}} \left( \frac{2^{n-1}}{\frac{2n-1+k-l}{2}} \right) \left( \frac{2^{n-1}}{\frac{k-2n-1+l}{2}} \right), & a \neq 0. \end{cases} \quad (4.8) $$

**Proof:** The $a = 0$ case follows because the distance between any function $f$ and the zero function is the weight of $f$. For $a \neq 0$, the result holds because

$$ P_d^{(a, b)}(l) = \sum_{k=0}^{2^n} P_{d\mid w}(l, k) P_w(k). \quad \Box $$

45
For any integer value $M_{LAT}$, $0 < M_{LAT} \leq 2^{n-1}$, and by denoting the number of LATs with any entry having absolute value $\geq M_{LAT}$ by $N_{LAT}^*$, we have the following upper bound:

**Corollary 4.1**

$$\frac{N_{LAT}^*}{\ln(n,m)} \leq 2 \left\{ (2^m - 1) \sum_{l=M_{LAT}}^{2^{n-1}} P_{d}^{(0,b)} (2^{n-1} - l) + (2^m - 1)(2^n - 1) \sum_{l=M_{LAT}}^{2^{n-1}} P_{d}^{(a,b)} (2^{n-1} - l) \right\}.$$  \hspace{1cm} (4.9)

**Proof:** We have

$$LAT(a, b) = 2^{n-1} - d(a \cdot x, b \cdot f).$$ \hspace{1cm} (4.10)

From Theorem 4.1, and by noting that

$$\frac{N_{LAT}^*}{\ln(n,m)} = P\left\{ \left( \max_{a,b\neq 0} |LAT(a,b)| \right) \geq M_{LAT} \right\} \leq \sum_{a,b\neq 0} P\{|LAT(a,b)| \geq M_{LAT}\},$$ \hspace{1cm} (4.11)

we get the Corollary above. \hfill \Box

Take the minimum value of $M_{LAT}$ for which $\frac{M_{LAT}}{\ln(n,m)} \leq 0.5$ as an estimate for the expected value of the maximum LAT entry, $\overline{LAT}$. Table 4.1 shows the simulation results for $\overline{LAT}$, the average value of the maximum entry in the LAT of a randomly selected $8 \times m$ injective s-box ($m = 10, 12, ..., 32$) together with our theoretical estimate, denoted by $\overline{LAT}^*$. Table 4.1 also shows the maximum and the minimum values for $\overline{LAT}$ found throughout the experiments. For all the results given in table 4.1, the sample variance is upper bounded by 2. The Fast Walsh transform [6] and other speed-up techniques were used in the calculation of the maximum LAT entries throughout our simulation. It is worth noting that for $m = 32$, four of the s-boxes out of the ten tested achieved a nonlinearity of 73, while the nonlinearity of the remaining six s-boxes was 72.
4.3 Construction of Highly Nonlinear Injective S-boxes

In this section we present two methods for constructing highly nonlinear injective s-boxes. Both of these methods, which are motivated by bounds on exponential sums, outperform the previously proposed methods. In particular, we are able to obtain injective $8 \times 32$ s-boxes with nonlinearity equal to 80 and maximum XOR table entry of 2. We also re-evaluate the resistance of the CAST encryption algorithm to the basic linear cryptanalysis.

4.3.1 Definitions

**Trace:** The sum

$$Tr(\alpha) = \sum_{j=0}^{m-1} \alpha^{2^j}$$

is called the trace of $\alpha \in GF(2^m)$.

**Basis and dual basis:** A set $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ of $m$ elements over $GF(2^m)$ which are linearly independent over $GF(2)$ is called a basis for $GF(2^m)$ over $GF(2)$.

The corresponding dual basis is defined to be the unique set of elements $\{\gamma_1, \gamma_2, \ldots, \gamma_m\} \subseteq GF(2^m)$ such that

$$Tr(\alpha_i \gamma_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

<table>
<thead>
<tr>
<th>$m$</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
<th>18</th>
<th>20</th>
<th>22</th>
<th>24</th>
<th>26</th>
<th>28</th>
<th>30</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Size</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$LAT^+$</td>
<td>39</td>
<td>41</td>
<td>43</td>
<td>45</td>
<td>47</td>
<td>48</td>
<td>50</td>
<td>52</td>
<td>53</td>
<td>55</td>
<td>56</td>
<td>58</td>
</tr>
<tr>
<td>$LAT^-$</td>
<td>38</td>
<td>40</td>
<td>42</td>
<td>44</td>
<td>46</td>
<td>47</td>
<td>49</td>
<td>50</td>
<td>52</td>
<td>53</td>
<td>55</td>
<td>56</td>
</tr>
<tr>
<td>Min $LAT^+$</td>
<td>34</td>
<td>36</td>
<td>39</td>
<td>41</td>
<td>43</td>
<td>44</td>
<td>46</td>
<td>48</td>
<td>50</td>
<td>52</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td>Max $LAT^+$</td>
<td>44</td>
<td>47</td>
<td>48</td>
<td>54</td>
<td>52</td>
<td>52</td>
<td>56</td>
<td>56</td>
<td>58</td>
<td>55</td>
<td>59</td>
<td>57</td>
</tr>
</tbody>
</table>

Table 4.1 Experimental Results for $8 \times m$ injective S-boxes
Lemma 4.3 (Carlitz and Uchiyama bound [25])

If $F(x)$ is a polynomial over $GF(2^m)$ of degree $r$ such that $F(x) \neq G(x)^2 + G(x) + b$ for all polynomials $G(x)$ over $GF(2^m)$ and constants $b \in GF(2^n)$, then

$$\left| \sum_x (-1)^{Tr(F(x))} \right| \leq (r - 1)2^{m/2}. \quad (4.14)$$

Lemma 4.4 (Kloosterman sum [137], [25])

$$\left| \sum_{x \in GF(2^m) \setminus \{0\}} (-1)^{Tr(x + \frac{x}{2})} \right| \leq 2^{m/2+1}. \quad (4.15)$$

Note that a function, $F$, over $GF(2^n)$ can also be expressed as a function over $GF(2)^n$, i.e., as $n$ functions over $GF(2)$. Let

$$f(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)) \quad (4.16)$$

be a function over $GF(2)^n$, let $B = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be any basis of $GF(2^n)$ over $GF(2)$, then

$$F(x) = \sum_{i=1}^{n} f_i(x_1, \ldots, x_n) \alpha_i \quad (4.17)$$

where $x = \sum_{i=1}^{n} x_i \alpha_i \in GF(2^n)$. This means that there is a one-to-one correspondence between the functions of $GF(2^n)$ and those of $GF(2)^n$ under a chosen basis of $GF(2^n)$ over $GF(2)$. If we let $\{\alpha_1^*, \ldots, \alpha_n^*\}$ be the dual basis of $B$, then each component of $f(x_1, \ldots, x_n)$ can be expressed as

$$f_i(x_1, \ldots, x_n) = Tr(F(x)\alpha_i^*), \quad (4.18)$$

where $x = \sum_{i=1}^{n} x_i \alpha_i^*$.  

48
The nonlinearity of the function $f(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$ is defined as the minimum Hamming distance between the set of affine functions and every nonzero linear combination of the output coordinates of $f$.

From the above, one can easily prove that the nonlinearity of the function $f$ that corresponds to the function $F$ is given by

$$\mathcal{NL}_f = \min_{c \neq 0, b, w} d(Tr(cF(X)), Tr(wX) \oplus b)$$

$$= 2^{n-1} - \max_{c \neq 0, w} \left| \sum_{x} (-1)^{Tr(cF(x)) \oplus wx} \right|.$$ (4.19)

where $c, w \in GF(2^n)$, $b \in GF(2)$.

4.3.2 Construction Method I

This construction method is based on the observation that highly nonlinear injective s-boxes may be obtained by adding the coordinate functions of highly nonlinear bijective s-boxes.

Given the distinct bijective functions $F_i$ over $GF(2^n)$, $1 \leq i \leq M$, an injective function $G$ over $GF(2^M)$ can be obtained by setting

$$G = (F_1 \| F_2 \| \cdots \| F_M).$$ (4.20)

In this method we use the inversion mapping proposed by Nyberg [91]

$$F_i(x) = \begin{cases} (x + a_i)^{-1}, & x \neq a_i, \\ 0, & x = a_i, \end{cases}$$ (4.21)

where $x \in GF(2^n)$.

Using lemma 4.4, the nonlinearity of the function $F_i$ is lower bounded by $2^{n-1} - 2^{n/2}$. Experimental results show that injective $8 \times 16$ s-boxes constructed by this method always have nonlinearity of 96. For $8 \times 24$ and 100 random choices of $a_i$ pairs, we found 57.40

---

1 The experimental results in this part were obtained by the second author in [141]
and 3 s-boxes with $NCL = 86.84$ and 80 respectively. The only $8 \times 32$ s-box tested to date has nonlinearity of 76.

**Conjecture**

The nonlinearity of the function $g$ that corresponds to the function

$$G = \left( x^{-1} \|(x + a)^{-1} \right), a \neq 0, x \in GF(2^n)$$  \hspace{1cm} (4.22)

obtained using construction method I, is bounded by

$$NCL_g \geq 2^{n-1} - 2^{n/2+1}.$$  \hspace{1cm} (4.23)

We verified this conjecture experimentally for $n \leq 10$ where we found that this bound is tight for even $n$.

**Remark:** We may prove that $NCL_g \geq 2^{n-1} - \left(2^{n/2+1} + 1 \right)$ by proving that for $a \neq 0$ we have

$$\left| \sum_{x \in GF(2^m) \setminus \{0,a\}} (-1)^{Tr(x + \frac{x^a}{x^a})} \right| \leq 2^{m/2+2}.$$  \hspace{1cm} (4.24)

but this seems to be a hard problem.

**4.3.3 Construction Method II**

This method is also based on the observation that highly nonlinear injective s-boxes may be obtained by adding the coordinate functions of highly nonlinear s-boxes (not necessary bijective).

If the concatenated functions are distinct polynomials over $GF(2^n)$ such that

$$wx + \sum_{i=1}^{M} a_i F_i(x) \neq U(x)^2 + U(x) + b$$  \hspace{1cm} (4.25)

for all polynomials $U(x)$ over $GF(2^n)$ and constants $a_i \in GF(2^n) \setminus \{0\}, b, w \in GF(2^n)$ then the Carlitz and Uchiyama bound can be used to provide a lower bound for the nonlinearity of the resulting s-box as follows.
If \( r = \max_i (\text{degree}(F_i)) \) then the nonlinearity of the resulting function is lower bounded by

\[
\mathcal{NL}_g \geq 2^{n-1} - 2^{n/2-1}(r - 1).
\] (4.26)

Using the Carlitz and Uchiyama bound one can check that the nonlinearity of the function \( F(x) = x^3, x \in GF(2^8) \) is lower bounded by 112. Also the nonlinearity of the function \( G(x) = (x^3||x^5), x \in GF(2^8) \) is lower bounded by 96.

Our basic result is based on the experimental observation that the Carlitz and Uchiyama bound is not tight for higher values of \( r \). In this section we consider the following five constructions

\[
\begin{align*}
G_1 &= (x^5||x^7||x^{11}||x^{13}), \\
G_2 &= (x^3||x^7||x^{11}||x^{13}), \\
G_3 &= (x^3||x^5||x^{11}||x^{13}), \\
G_4 &= (x^3||x^5||x^7||x^{13}), \\
G_5 &= (x^3||x^5||x^7||x^{11}).
\end{align*}
\] (4.27)

Using Carlitz and Uchiyama bound we have

\[
\mathcal{NL}_g \geq \begin{cases} 
32, & i = 1, 2, 3, 4, \\
48, & i = 5.
\end{cases}
\] (4.28)

Experimental results shows that

\[
\mathcal{NL}_g = \begin{cases} 
72, & i = 1, 2, \\
80, & i = 3, 4, 5.
\end{cases}
\] (4.29)

Let \( F_i = x^3, x^5, x^7, x^{11}, x^{13} \) for \( i = 1, 2, 3, 4, 5 \) respectively. By noting that \( F_i \) is bijective for \( i = 3, 4, 5 \) then any construction that includes any of \( F_i, i = 3, 4 \) or 5 will be injective. In fact, it is not hard to see that all the s-boxes below are injective.

Table 4.2 below shows the nonlinearity and the maximum XOR table entry, \( XOR^* \), for the injective \( 8 \times 16 \) s-boxes constructed by this method by concatenating \( F_i \) with \( F_j \). Table 4.3 shows similar results for the \( 8 \times 24 \) s-boxes.
Table 4.2 \(NL, XOR^*\) for 8 x 16 S-boxes Obtained Using Construction Method II

<table>
<thead>
<tr>
<th>i, j, k</th>
<th>1, 2, 3</th>
<th>1, 2, 4</th>
<th>1, 2, 5</th>
<th>1, 3, 4</th>
<th>1, 3, 5</th>
<th>1, 4, 5</th>
<th>2, 3, 4</th>
<th>2, 3, 5</th>
<th>2, 4, 5</th>
<th>3, 4, 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NL)</td>
<td>96</td>
<td>96</td>
<td>80</td>
<td>80</td>
<td>96</td>
<td>96</td>
<td>88</td>
<td>96</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>XOR^*</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4.3 \(NL, XOR^*\) for 8 x 24 S-boxes Obtained Using Construction Method II

<table>
<thead>
<tr>
<th>i, j, k</th>
<th>1, 2, 3</th>
<th>1, 2, 4</th>
<th>1, 2, 5</th>
<th>1, 3, 4</th>
<th>1, 3, 5</th>
<th>1, 4, 5</th>
<th>2, 3, 4</th>
<th>2, 3, 5</th>
<th>2, 4, 5</th>
<th>3, 4, 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NL)</td>
<td>96</td>
<td>96</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>88</td>
<td>80</td>
<td>88</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>XOR^*</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

For the 8 x 32 s-boxes, \(XOR^* = 2\) for all the constructions above except for \(G_1\) where it is equal to 4 which may limit the usefulness of \(G_1\).

Table 4.4 shows the best s-box nonlinearity obtained by different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>(8 \times 16)</th>
<th>(8 \times 24)</th>
<th>(8 \times 32)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>87</td>
<td>80</td>
<td>73</td>
</tr>
<tr>
<td>Mister and Adams [9]</td>
<td>—</td>
<td>—</td>
<td>74</td>
</tr>
<tr>
<td>Method I</td>
<td>96</td>
<td>86</td>
<td>76</td>
</tr>
<tr>
<td>Method II</td>
<td>96</td>
<td>96</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 4.4 Best S-box Nonlinearity Obtained By Different Construction Methods

The construction methods proposed in this section can be extended to other highly nonlinear mappings such as that proposed in [26],[91].

In order to frustrate possible algebraic attacks, the four 8 x 32 s-boxes should be generated using different irreducible polynomials. When using s-boxes obtained from construction method II, we recommend that the bytes XORed together should correspond to different degrees. For example, if \(G_5\) is used, then the 8 x 32 s-boxes may be constructed as follows

\[
\begin{align*}
\text{s}_1 &= (x^3||x^5||x^7||x^{11}), \\
\text{s}_2 &= (x^5||x^7||x^{11}||x^3), \\
\text{s}_3 &= (x^7||x^{11}||x^3||x^5), \\
\text{s}_4 &= (x^{11}||x^3||x^5||x^7),
\end{align*}
\]

such that the exponents form a Latin square [113].
4.3.4 Comments on the Security of the CAST Encryption Algorithm

Figure 4.1 shows the CAST round function. In this paper we assume that operations $a, b, c$ and $d$ are XOR addition of 32-bit quantities.

The resistance of CAST-like encryption algorithms [2] constructed using randomly generated s-boxes against the basic linear cryptanalysis [82] was studied in [75]. The number of known plaintexts, $N_p$, in a basic linear attack (Algorithm 1 in [82]) required to give a 97.7% confidence of getting the right key bit is approximately given by

$$N_p \approx \left| p_l - \frac{1}{2} \right|^{-2},$$

(4.31)

$$\left| p_l - \frac{1}{2} \right| \leq 2^{7-1} \left| p_s - \frac{1}{2} \right|^{\gamma},$$

(4.32)

where $\gamma$ is the number of s-boxes involved in the $R$-round linear approximation expression, $|p_s - \frac{1}{2}| = \frac{2^{n-1} - N L_S}{2^n}$ and $n$ is the number of input bits to the s-box. The bounds for $N_p$ can be improved by re-evaluating the expressions in [75] using the new $8 \times 32$ s-boxes nonlinearity. However a better bound can be obtained by considering the nonlinearity of the resulting $32 \times 32$ s-boxes. The nonlinearity, $N L_S$
can be bounded using the nonlinearity, $\mathcal{NL}_s, 1 \leq i \leq 4$, of the four $8 \times 32$ s-boxes used in its construction as follows

$$\mathcal{NL}_S \geq 2^{32} - \frac{1}{2} \prod_{i=1}^{4} (2^8 - 2\mathcal{NL}_s).$$

(4.33)

The exact nonlinearity can be efficiently calculated using the Walsh transforms [6] of the four $8 \times 32$ s-boxes. Since an $R$-round linear approximation must involve at least as many s-boxes as $R/2$ iterations of the best 2-round approximation, the number of $32 \times 32$ s-boxes involved in an $R$-round linear approximation is at least $R/2$ and hence we have

$$N_p \approx \left| \frac{1}{2} - \frac{\mathcal{NL}_S}{2^{32}} \right|^{-R}$$

(4.34)

where $\mathcal{NL}_S$ is the nonlinearity of the $32 \times 32$ s-box.

An important observation, which was overlooked in [87] is that the nonlinearity of the $32 \times 32$ s-box depends not only on the nonlinearity of $8 \times 32$ s-boxes used in the construction, but it also depends on how the four $8 \times 32$ s-boxes interact together. This means that improving the nonlinearity of the individual $8 \times 32$ s-boxes does not always guarantee improving the resistance of the cipher to the basic linear cryptanalysis. For example, when combining the output of the four CAST-128 s-boxes [2] (each with nonlinearity 74) by XOR, the resulting $32 \times 32$ s-box has nonlinearity $2.132.774.912$ which is less than $2.133.721.088$, the nonlinearity we found by combining four randomly selected s-boxes with nonlinearity less than 74. Using such s-boxes, we have $N_p \approx 2^{72} > 2^{64}$ for $R = 8$ which is much higher than $N_p \approx 2^{34}$ estimated in [75].

Finally, one should note that the primary motive for this work is obtaining highly nonlinear injective s-boxes. We are not proposing the use of such s-boxes in CAST-like ciphers before examining their other cryptographic properties. In fact, we believe that randomly selected s-boxes are a good choice for CAST-like ciphers.

---

O'Connor and Klapper [100] showed that the expected degree of the Algebraic Normal Form of a randomly selected boolean function $f : Z_2^n \rightarrow Z_2$ is equal to $\frac{n}{2} + \Theta(\frac{1}{\sqrt{n}})$. This suggests that randomly selected $8 \times 32$ s-boxes are more resistant to higher order differential cryptanalysis than CAST-128 s-boxes.
Chapter 5 Information Leakage and Spectral Properties of Boolean Functions

5.1 Introduction

Several cryptographic criteria have been previously proposed as a measure of the strength of cryptographic functions. Among these criteria are balance, correlation immunity [126], resiliency [28], nonlinearity [84], Strict Avalanche Criterion (SAC) [136], higher order SAC [47], Propagation Criterion (PC), higher order PC [105], Bit Independence Criterion [135], and Completeness [66].

The above set of cryptographic criteria are not independent of each other and a cryptographic function that satisfies all these criteria would be a golden one. Unfortunately, it can be proven that no function can satisfy all the above set of criteria simultaneously. This can be considered as the main motive for proposing a new set of criteria based on information theory.

Several design criteria, based on information theory, have been proposed in [37],[48], [130], and [155].

Information Leakage can be classified into two classes: Static information leakage and dynamic information leakage. It is argued in [155] that a boolean function is resistant to statistical analysis (e.g., differential cryptanalysis [15], linear cryptanalysis [82], and Siegenthaler’s correlation attack [125]) if there is no significant static and dynamic information leakage between its inputs and outputs.

It is worth noting that Brynielsson [24] gives an approximate expression for the expected value of the mutual information between the output and input subvectors for multi-output boolean functions. Here we follow the definition of information leakage given in [155] and give an exact expression for the expected value of different forms of these information leakages for a randomly selected multi-output boolean function and for
some other combinatorial structures of interest such as regular mappings, and injective mappings.

Gordon and Retkin [50] conjectured that good substitution boxes (s-boxes) may be built by choosing a random reversible mapping of sufficient size. Their argument is based on the observation that the probability of accidental linearity occurring in such s-boxes decreases dramatically as the size of the s-box increases. In this chapter, we provide further evidence that bigger s-boxes (by bigger we mean s-boxes with a larger number of inputs) are better by showing that the expected value of information leakage of a randomly selected boolean function decreases rapidly with the number of input variables. We also give extended definitions of many cryptographic criteria such as such as balance, correlation immunity, SAC, higher order SAC, and Propagation Criterion to multi-output boolean functions.

Finally we study the relationship between the Walsh-Hadamard transform and various types of information leakage. Conditions on the Walsh transform are given, which imply that the function satisfies certain cryptographic properties of interest.

5.2 Definitions

Entropy and Mutual Information

*Entropy*

Let $X$ be a discrete random variable with alphabet $\mathcal{X}$ and probability distribution function $p(x) = Pr\{X = x\}, x \in \mathcal{X}$ then the entropy of a random variable $X$ is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$

(5.1)

If $X$ is a binary random variable with $Pr(x = 0) = p$, then

$$H(X) = -p \log_2 p - (1 - p) \log_2 (1 - p) = h(p),$$

(5.2)
where \( h(p) \) is called the binary entropy function of probability \( p \).

**Conditional Entropy**

Let \( Y \) and \( X \) be discrete random variables with alphabet \( \mathcal{Y}, \mathcal{X} \) respectively, and with a conditional distribution \( p(y|x) \) then the conditional entropy \( H(Y|X) \) is defined by

\[
H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X = x),
\]

where

\[
H(Y|X = x) = -\sum_{y \in \mathcal{Y}} p(y|x) \log_2(p(y|x)).
\]

**Mutual Information**

The mutual information of discrete random variables \( Y \) and \( X \) is defined by

\[
I(Y; X) = H(Y) - H(Y|X).
\]

For a good general reference for information theory, see [30].

Throughout this chapter, let \( f : Z_2^n \rightarrow Z_2^m \) be a randomly selected boolean function and let \( Y \) denote the output of \( f \).

**Static Information Leakage**

The static information leakage of \( Y \), given input subvector \( X_k \in Z_2^k \) (i.e., given that we know \( k \) bits of the \( n \)-bit input vector), is defined by

\[
SL(Y|X_k) = m - H(Y|X_k),
\]

where \( H(Y \mid X_k) \) is the conditional entropy of \( Y \) given \( X_k \).
Remark: It is easy to show that

\[ SL(Y | X_k) = m - H(Y) + I(Y; X_k). \]  \hspace{1cm} (5.7)

where \( I(Y; X_k) \) is the mutual information between \( Y \) and \( X_k \). Note that if the mutual information \( I(Y; X_k) \) is used to define the static information leakage, then the minimum of \( I(Y; X_k) \) can be achieved while \( H(Y) = 0 \) which contradicts our objective.

The self static information leakage of \( Y \) is defined as:

\[ SSL(Y) = m - H(Y). \]  \hspace{1cm} (5.8)

By noting that \( H(Y | X_k) \leq n - k \), then from (5.6) we have

\[ SL(Y | X_k) \geq m - n + k. \]  \hspace{1cm} (5.9)

One should note that the value of all the forms of information leakage defined above is greater than or equal to zero, i.e., we always have \( SSL(Y) \geq 0 \), and \( SL(Y | X_k) \geq 0 \).

**Dynamic Information Leakage**

The dynamic information leakage of \( \Delta Y \), given the input change vector \( \Delta X \) is defined by:

\[ DL(\Delta Y | \Delta X) = m - H(\Delta Y | \Delta X), \]  \hspace{1cm} (5.10)

where \( \Delta Y = f(X) \oplus f(X \oplus \Delta X) \).

We also note that the static information leakage \( SL(Y | X_k) = 0 \) is achieved by \( k^{th} \) order resilient functions (see [28], [126] for the definition and properties of resilient functions), while zero dynamic information leakage for all values of \( \Delta X \neq 0 \) is achieved only by perfect nonlinear functions (see [90], [116] for the definition and properties of both bent and perfect nonlinear functions).
By noting that we always have $H(\Delta \mathbf{Y}|0) = 0$, then from (5.10) we have

$$DL(\Delta \mathbf{Y}|\Delta \mathbf{X}) = m - \frac{2^n - 1}{2^n} \sum_{\Delta \mathbf{x} \neq \mathbf{0}} H(\Delta \mathbf{Y}|\Delta \mathbf{x})$$

$$\geq m - \frac{2^n - 1}{2^n} m = \frac{m}{2^n}. \quad (5.11)$$

The equality in the above bound can be achieved only by perfect nonlinear functions for which $n \geq 2m$.

Let

$$N_y = \#\{x \in \mathbb{Z}_2^n | f(x) = y\},$$

$$N_{xy} = \#\{x \in \mathbb{Z}_2^n | X_k = \mathbf{x}, Y = y\},$$

$$N_{\Delta x \Delta y} = \#\{x \in \mathbb{Z}_2^n | f(x \oplus \Delta x) \oplus f(x) = \Delta y\}. \quad (5.12)$$

where $\mathbf{x} \in \mathbb{Z}_2^k$, $\Delta \mathbf{x} \in \mathbb{Z}_2^n$, $y \in \mathbb{Z}_2^m$, $\Delta \mathbf{y} \in \mathbb{Z}_2^m$.

Assuming that all input vectors are equally probable, we have:

$$SSL(\mathbf{Y}) = m - \sum_{y \in \mathbb{Z}_2^m} \frac{N_y}{2^n} \log_2 \left(\frac{2^n}{N_y}\right),$$

$$SL(\mathbf{Y} | \mathbf{X}_k) = m - 2^{-k} \sum_{y \in \mathbb{Z}_2^m \atop \mathbf{x} \in \mathbb{Z}_2^n} \frac{N_{xy}}{2^{n-k}} \log_2 \left(\frac{2^{n-k}}{N_{xy}}\right), \quad (5.13)$$

$$DL(\Delta \mathbf{Y} | \Delta \mathbf{X}) = m - 2^{-n} \sum_{\Delta \mathbf{x} \in \mathbb{Z}_2^n \atop \Delta \mathbf{y} \in \mathbb{Z}_2^m} \left(\frac{N_{\Delta x \Delta y}}{2^n}\right) \log_2 \left(\frac{2^n}{N_{\Delta x \Delta y}}\right).$$

The problem of finding the expected values of the above forms of information leakage is now reduced to finding the marginal probability distribution of the random variables $N_y, N_{xy}$ and $N_{\Delta x \Delta y}$. 
5.3 Information Leakage of a Randomly Selected Boolean Function

Lemma 5.1

Let $Y$ be the output of a randomly selected boolean function $f : Z^m_2 \rightarrow Z^m_2$ then we have the following probabilities:

\[
P(N_y = i) = \binom{2n}{i} \left( \frac{1}{2m} \right)^i \left( 1 - \frac{1}{2m} \right)^{2n-i},
\]

\[
P(N_{\Delta Y} = i) = \binom{2n-k}{i} \left( \frac{1}{2m} \right)^i \left( 1 - \frac{1}{2m} \right)^{2n-k-i},
\]

\[
P(N_{\Delta X \Delta Y} = 2i) = \binom{2n-1}{i} \left( \frac{1}{2m} \right)^i \left( 1 - \frac{1}{2m} \right)^{2n-1-i}, \quad \Delta X \neq 0.
\]

Proof: The proof of the above lemma follows by noting that $N_y$, $N_{\Delta Y}$, and $N_{\Delta X \Delta Y}/2$ follow the multi-nomial distribution.

Theorem 5.1

The expected values of the static and dynamic information leakage of a randomly selected boolean function $f : Z^m_2 \rightarrow Z^m_2$ are given respectively by

\[
SL(Y | X_k) = m - 2^m \sum_{i=0}^{2n-k} P(N_{\Delta Y} = i) \left( \frac{i}{2n-k} \right) \log_2 \left( \frac{2n-k}{i} \right), \quad 0 \leq k \leq n.
\]

\[
DL(\Delta Y | \Delta X) = m - \frac{2^m(2^n - 1)}{2n} \sum_{i=0}^{2n-1} P(N_{\Delta X \Delta Y} = 2i) \left( \frac{i}{2n-1} \right) \log_2 \left( \frac{2n-1}{i} \right).
\]

Proof: Theorem 5.1 follows directly from the definition of the expected value, and (for part 2) by noting that $\Delta Y = 0$ for $\Delta X = 0$, and hence $H(\Delta Y | 0) = 0$.

Figures 5.2 and 5.3 show the expected value of the self static information leakage and the expected value of the static leakage given that half the input bits are known. From
these graphs, it is clear that the relative dimensions of the boolean functions (i.e., the ratio between \( n, m \)) greatly affect different forms of information leakage.

Based on the results above, one cannot conclude that s-boxes with \( n > m \) are better than s-boxes with \( n < m \) because of the method we used in the normalization step (dividing by the number of output bits to get information leakage per output bit). Moreover, s-boxes with \( n < m \) provide better diffusion characteristics, and may be used in SPNs with no permutation layers [3] which leads to faster software implementation. The conclusion that we can make at this time is that all forms of information leakage seem to decrease with the number of input variables.

Using theorem 5.1, one can derive an upper bound for the information leakage of a randomly selected single output boolean function. Single output functions are of practical interest especially for the combining functions in stream ciphers.

![Figure 5.1 Lower Bound for the Binary Entropy Function](image)
**Corollary 5.1**

Let $Y$ be the output of a randomly selected single output boolean function, then the expected values of both the static leakage and dynamic leakage are bounded by

\[
\overline{SL(Y | X_k)} \leq \frac{1}{2^{n-k}}, \quad 0 \leq k \leq n. \tag{5.16}
\]

\[
\overline{DL(\Delta Y | \Delta X)} \leq \frac{3}{2^n}.
\]

**Proof:** The above corollary follows by direct substitution into theorem 5.1 and by noting that for the binary entropy function

\[
h(t) = -t \log_2(t) - (1 - t)\log_2(1 - t). \tag{5.17}
\]

we have (see Figure 5.1)

\[
h(t) \geq h_1(t) = 4t - 4t^2, \quad 1 \geq t \geq 0. \tag{5.18}
\]
Figure 5.2 Expected Value of $SSL(Y)$ for a Randomly Selected Boolean Function

Figure 5.3 Expected Value of $SSL(Y)$ and Expected Value of $SL(Y | X_{n/2})$ for a Randomly Selected Boolean Function
5.4 Information Leakage of a Randomly Selected Balanced Boolean Function

In this section, we calculate the expected values of both the dynamic leakage and the static leakage for regular (balanced) functions.

**Lemma 5.2**

Let $Y$ be a randomly selected balanced function $f : Z_2^n \rightarrow Z_2^m$, $n \geq m$, then we have

$$P(N_{xy} = i) = \frac{1}{B(n, m)} \binom{2^{n-k}}{i} \binom{2^n - 2^{n-k}}{2^{n-m} - i} \frac{(2^n - 2^{n-m})!}{(2^{n-m})!(2^{m-1})}.$$  

(5.19)

where $B(n, m) = \frac{2^n!}{(2^{n-m})!m!}$ is the number of $n \times m$ balanced boolean functions.

**Proof**: For $n \times m$ balanced boolean functions, we have $N_y = 2^{n-m}$. If we fix $k$ input variables, then there are $\binom{2^{n-k}}{i} \binom{2^n - 2^{n-k}}{2^{n-m} - i}$ ways of arranging the output such that $Y = y$ when $X_k = \bar{x}$ for $i$ times. The remaining $(2^n - 2^{n-m})$ outputs, of which there are only $(2^{n-m} - 1)$ distinct ones, can be permuted in $\frac{(2^n - 2^{n-m})!}{(2^{n-m})!(2^{m-1})}$ ways. \qed

**Corollary 5.2**

Let $Y$ be a randomly selected bijective mapping $\pi : Z_2^n \rightarrow Z_2^n$ then the expected value of the static information leakage of $Y$ given the input subvector $X_k$, $0 \leq k \leq n$, is given by

$$\overline{SL}(Y \mid X_k) = k.$$  

(5.20)

**Proof**: The proof follows directly by substituting (5.19), with $n = m$, into the first part of (5.15). A simpler proof (independent of lemma 5.2 ) follows by noting that for any arbitrary bijective function, $\pi : Z_2^n \rightarrow Z_2^n$, if we fix $k$ input bits, we will have $2^{n-k}$ different output symbols with $H(Y \mid X_k) = n - k$. \qed

64
From the proof of the lemma above, it follows that we have

$$S_L(Y \mid X_k) = k$$  \hspace{1cm} (5.21)

for any arbitrary bijective function.

In chapter 3, we derived expressions for the number of balanced functions with

$$N_{\Delta x \Delta y} = 2i, \quad 0 \leq i \leq 2^{n-1}.$$  \hspace{1cm} \text{Using these results we have for } \Delta x \neq 0

$$P(N_{\Delta x \Delta y} = 2i) = \begin{cases} \frac{\Lambda_{n,m,\Delta y}(i)}{B(n,m)}, & \Delta y \neq 0, \\ \frac{\Lambda_{n,m,0}(i)}{B(n,m)}, & \Delta y = 0. \end{cases}$$  \hspace{1cm} (5.22)

where \(\Lambda_{n,m,\Delta y}(i), \Lambda_{n,m,0}(i)\) are given by equations (3.13) and (3.14).

Thus the expected value of the dynamic leakage of a randomly selected balanced function is given by

**Theorem 5.3**

$$DL(\Delta Y \mid \Delta X) = m - \frac{(2^n - 1)(2^m - 1)}{2^n} \sum_{i=0}^{2^{n-1}} \frac{\Lambda_{n,m,\Delta y}(i)}{B(n,m)} \left( \frac{i}{2^{n-1}} \right) \log_2 \left( \frac{\frac{i}{2^{n-1}}}{i} \right)$$

$$- \frac{(2^n - 1)}{2^n} \sum_{i=0}^{2^{n-1}} \frac{\Lambda_{n,m,0}(i)}{B(n,m)} \left( \frac{i}{2^{n-1}} \right) \log_2 \left( \frac{\frac{2^{n-1}}{i}}{i} \right).$$  \hspace{1cm} (5.23)

**Corollary 5.2**

Let \(Y\) be the output of a randomly selected bijective mapping \(\pi : Z_2^n \rightarrow Z_2^n\) then the expected value of the dynamic information leakage given the input change vector \(\Delta X\), is given by

$$DL(\Delta Y \mid \Delta X) = n - \frac{(2^n - 1)^2}{2^n} \sum_{i=0}^{2^{n-1}} \frac{\Lambda_{n,n,\Delta y}(i)}{n!} \left( \frac{i}{2^{n-1}} \right) \log_2 \left( \frac{\frac{2^{n-1}}{i}}{i} \right).$$  \hspace{1cm} (5.24)

**Proof:** The corollary above is a special case (with \(n = m\)) of theorem (5.3). \hspace{1cm} \Box
Remark: Note that $\Lambda_{n,n,0} = 0$ as each output symbol occurs once.

For $m = n$, the minimum dynamic information leakage is achieved by functions with differentially 2-uniform XOR table. Hence, using equation (5.13), we have

$$DL(\Delta Y | \Delta X) = n - 2^{-n} \sum_{\Delta x \in \mathbb{Z}_2^n \setminus \Delta x} \left( \frac{N_{\Delta x \Delta y}}{2^n} \right) \log_2 \left( \frac{2^n}{N_{\Delta x \Delta y}} \right)$$

$$\geq n - 2^{-n} (2^n - 1) (2^{n-1}) \frac{2}{2^n} \log_2 \left( \frac{2^n}{2} \right)$$

$$= n - \frac{(2^n - 1)(n - 1)}{2^n}$$

$$= 1 + \frac{n - 1}{2^n}.$$  \hfill (5.25)

Figure 5.4 shows a comparison between the expected value of dynamic information leakage of a randomly chosen $n \times n$ bijective mapping and that of a randomly chosen function of the same dimensions.

![Figure 5.4 Expected Value of $DL(\Delta Y | \Delta X)$ for an $n \times n$ Random Mapping and an $n \times n$ Random Bijective Mapping](image)
5.5 Information Leakage of a Randomly Selected Injective Boolean Function

In this section, we calculate the expected values of both the dynamic leakage and the static leakage for injective functions.

**Theorem 5.3**

Let $Y$ be the output of a randomly selected injective function $f : Z_2^n \rightarrow Z_2^m$. $n \leq m$, then

$$\mathbb{E}[L(Y \mid X_k)] = (m - n) + k. \quad (5.26)$$

**Proof:** The theorem follows by noting that for any arbitrary injective function, $f : Z_2^n \rightarrow Z_2^m$, if we fix $k$ input bits, we have $2^{n-k}$ different output symbols with $H(Y \mid X_k) = n - k$.

From the proof of the lemma above, it follows that we have

$$\mathbb{E}[L(Y \mid X_k)] = (m - n) + k \quad (5.27)$$

for any arbitrary injective function.

**Lemma 5.3**

The number of injective functions with $N_{\Delta x \Delta y} \geq 2k, \Delta x \neq 0, \Delta y \neq 0$, is upper bounded by

$$\Psi_{n,m}(k) = \binom{2^{n-1}}{k} \binom{2^{m-1}}{k} 2^k In(2^m - 2k, 2^n - 2k), \quad (5.28)$$

where

$$In(u, v) = \prod_{i=0}^{v-1} (u - i). \quad (5.29)$$

**Proof:** We will count the number of ways to construct an injective function such that $N_{\Delta x \Delta y} \geq 2k, \Delta x \neq 0, \Delta y \neq 0$. There are $\binom{2^{n-1}}{k}$ possible choices for the $x$ positions
of these \( k \) pairs and there are \( \binom{2^{m-1}}{k} \) choices for such pairs. We also have 2 possible orders for each pair, giving \( 2^k \) total possible orders. The rest of the output symbols can be picked up in any order from the remaining \( 2^m - 2k \) unused symbols, i.e., they can be picked in \( \ln(2^m - 2k, 2^n - 2k) \) ways.

Using the Inclusion-Exclusion principle, we get

**Lemma 5.4**

The exact number of injective functions with \( N_{\Delta x \Delta y} = 2k, \Delta x \neq 0, \Delta y \neq 0 \), is given by

\[
\Lambda_{n,m,\Delta y}(k) = \sum_{i=k}^{2^{n-1}} (-1)^{i-k} \binom{i}{k} \psi_{n,m}(i).
\]  

(5.30)

Remark: Note that \( N_{\Delta x 0} = 0 \) for \( \Delta x \neq 0 \), and hence \( \Lambda_{n,m,0} = 0 \).

**Theorem 5.4**

\[
\overline{DL(\Delta Y|\Delta X)} = m - \frac{(2^n - 1)(2^m - 1)}{2^n} \sum_{i=0}^{2^{n-1}} \frac{\Lambda_{n,m,\Delta y}(i)}{\ln(2^m, 2^n)} \left( \frac{i}{2^{n-1}} \right) \log_2 \left( \frac{2^{n-1}}{i} \right).
\]  

(5.31)

where \( \ln(2^m, 2^n) \), the number of \( n \times m \) injective boolean functions, is given by equation (5.29).

Numerical substitution into the theorem (5.4) shows that the dynamic information leakage of a randomly selected injective function decreases with the number of input variables.

This rate of decrease is very similar to that of a randomly selected boolean function with the same number of inputs and outputs, especially for \( n \ll m \). This can be explained by noting that for \( n \ll m \), a randomly selected function is most likely to be injective.
5.6.1 Relation Between the Walsh Transform and Different Forms of Information Leakage

Let $y$ be the output of a boolean function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$. Let $N_y$, $N_{xy}$ and $N_{\Delta x \Delta y}$ be as defined in equation (5.12).

Recall that

$$F_c(w) = \frac{1}{2^{n/2}} \sum_x (-1)^{c \cdot f(x)} (-1)^{w \cdot x}.$$  \hspace{1cm} (5.32)

Thus for $w = 0$ we have

$$F_c(0) = \frac{1}{2^{n/2}} \sum_x (-1)^{c \cdot f(x)}.$$  \hspace{1cm} (5.33)

Multiplying both sides by $(-1)^{c \cdot y}$ and summing over $c$ we get

$$\sum_c F_c(0)(-1)^{c \cdot y} = \frac{1}{2^{n/2}} \sum_c \sum_x (-1)^{c \cdot f(x)} (-1)^{c \cdot y}$$

$$= \frac{1}{2^{n/2}} \sum_x \sum_c (-1)^{c \cdot (f(x) \oplus y)}.$$  \hspace{1cm} (5.34)

By noting that

$$\sum_c (-1)^{c \cdot (f(x) \oplus y)} = \begin{cases} 2^m, & f(x) = y, \\ 0, & \text{otherwise}, \end{cases}$$  \hspace{1cm} (5.35)

we get

$$\frac{1}{2^{n/2}} \sum_x \sum_c (-1)^{c \cdot (f(x) \oplus y)}$$

$$= 2^{m-n/2} \times \# \{ x | f(x) = y \} = 2^{m-n/2} N_y.$$  \hspace{1cm} (5.36)

and hence

$$N_y = 2^{n/2-m} \sum_c F_c(0)(-1)^{c \cdot y}.$$  \hspace{1cm} (5.37)

If $w_i$ denotes the vector with 1 in position $i$ and zeros otherwise, then

$$2^{n/2}F_c(w_i) = \sum_y (N_{0y} - N_{1y})(-1)^{c \cdot y}.$$  \hspace{1cm} (5.38)

where $N_{by}$ denotes the number of times the symbol $y$ appears, and $x_i = b$. Thus we have
Adding and subtracting equations (5.39) and (5.40) we get

\[ N_0y = N_0y + N_1y = 2^{n/2-m} \sum_c F_c(w_i)(-1)^{c \cdot y}. \]

Adding and subtracting equations (5.39) and (5.40) we get

\[ N_y = N_0y + N_1y = 2^{n/2-m} \sum_c F_c(0)(-1)^{c \cdot y}. \] (5.40)

Similarly we can prove that

\[ N_{\hat{x}y} = 2^{n/2-m-k} \sum_{c,\hat{w}} F_c(w^o)(-1)^{\hat{w} \cdot \hat{x} \oplus c \cdot y}, \] (5.42)

where \( w^o \) denotes the \( n \)-dimensional vector obtained by completing the \( k \)-dimensional subvector \( \hat{w} \) with zeros. For example if \( n = 6 \), \( \hat{x} = \{ x_0, x_2, x_5 \} \) then \( w^o = \{ \hat{w}_0, 0, \hat{w}_2, 0, 0, \hat{w}_5 \} \).

Recall that the autocorrelation function is related to the Walsh transform as follows:

\[ \Pi_{c,f}(\Delta x) = \frac{1}{2^n} \sum_w F_c^2(w)(-1)^{\Delta x \cdot w}. \] (5.43)

Multiplying both sides by \( (-1)^{c \cdot \Delta y} \) and summing over \( c \) we get

\[ \sum_c \Pi_{c,f}(\Delta x)(-1)^{c \cdot \Delta y} = \frac{1}{2^n} \sum_c \sum_w F_c^2(w)(-1)^{\Delta x \cdot w}(-1)^{c \cdot \Delta y}. \] (5.44)
The left hand side of the equation above can be expressed as

\[ \sum_c \Pi_c f(\Delta x)(-1)^c \Delta y \]

\[ = \frac{1}{2^n} \sum_c \sum_x (-1)^c (f(x) \oplus f(x \oplus \Delta x) \oplus \Delta y) \]

\[ = \frac{1}{2^n} \sum_x \sum_c (-1)^c (f(x) \oplus f(x \oplus \Delta x) \oplus \Delta y) . \tag{5.45} \]

By noting that

\[ \sum_c (-1)^c (f(x) \oplus f(x \oplus \Delta x) \oplus \Delta y) \]

\[ = \begin{cases} 2^m, & f(x) \oplus f(x \oplus \Delta x) = \Delta y, \\ 0, & \text{otherwise}, \end{cases} \tag{5.46} \]

we get

\[ \sum_c \Pi_c f(\Delta x)(-1)^c \Delta y \]

\[ = 2^{m-n} \# \{ x | f(x) \oplus f(x \oplus \Delta x) = \Delta y \} \]

\[ = 2^{m-n} N_{\Delta x \Delta y} . \tag{5.47} \]

and hence

\[ N_{\Delta x \Delta y} = \frac{1}{2^m} \sum_{c,w} F_c^2(w)(-1)^{\Delta x \cdot w \oplus \Delta y \cdot c} . \tag{5.48} \]

Using the above results we get the following theorem:
**Theorem 5.5**

Let $y$ be the output of a boolean function $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ then the different forms of information leakage of $y$ can be expressed as:

$$SSL(Y) = m - \sum_{y \in \mathbb{Z}_2^m} \frac{N_y}{2^n} \log_2 \left( \frac{2^n}{N_y} \right),$$

$$SL(Y|X_k) = m - 2^{-k} \sum_{y \in \mathbb{Z}_2^m} \frac{N_{xy}}{2^{n-k}} \log_2 \left( \frac{2^{n-k}}{N_{xy}} \right),$$

$$DL(\Delta Y|\Delta X) = m - 2^{-n} \sum_{\Delta x \in \mathbb{Z}_2^n} \left( \frac{N_{\Delta x \Delta y}}{2^n} \right) \log_2 \left( \frac{2^n}{N_{\Delta x \Delta y}} \right).$$

where $N_y, N_{xy}, N_{\Delta x \Delta y}$ are given by

$$N_y = 2^{n/2-m} \sum_c F_c(0)(-1)^{c \cdot y},$$

$$N_{xy} = 2^{n/2-m-k} \sum_{c, \bar{w}} F_c(w^0)(-1)^{\bar{w} \cdot \bar{x} \bar{c} \cdot y}.$$  

(5.50)

$$N_{\Delta x \Delta y} = \frac{1}{2^m} \sum_{c, \bar{w}} F_c^2(w)(-1)^{\Delta x \cdot \bar{w} \otimes \Delta y \cdot \bar{c}}.$$

Theorem (5.5) expresses the static and dynamic information leakage of a multi-output boolean function in terms of the Walsh transform of the nonzero linear combinations of its output coordinates. It is worth noting that static leakage of order $k$, $SL(y|x_k)$ depends on every Walsh coefficient $F_c(w)$ for $c \in \mathbb{Z}_m\cdot wt(w) \leq k$, and by noting that $SL(y|x_{k+1}) \geq SL(y|x_k)$ this implies that the Walsh coefficients for $w$ with smaller Hamming weight have more impact on the static information leakage. It is also clear that minimizing the dynamic information leakage can be achieved by having the energy spectrum $F_c^2(w)$ as flat as possible. Similar results for single output boolean functions were reported by Forré [48].
5.6.1.1 Extended Definitions

In this section we consider some extended definitions for some other cryptographic criteria to the multi-output boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$.

As an illustration for how one can extend the definitions of cryptographic criteria, consider the Strict Avalanche Criterion (SAC) defined for single-output boolean function as:

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ satisfies the SAC if whenever a single input bit is complemented, the output bit changes with a probability of one half [136].

The natural extended definition of SAC for multi-output boolean functions would be:

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ satisfies SAC if whenever a single input bit is complemented, the resulting output change vector occurs with a probability $p = \frac{1}{2^m}$.

The same concept can be applied to other cryptographic criteria such as balance, Propagation Criterion [105], correlation immunity [126,140], higher order SAC, and higher order Propagation Criterion [105].

The following is a summary of these extended definitions.

**Definition:**

- A balanced function is defined as a function for which $N_y = 2^{n-m}$.
- A $k$ resilient function is a function for which $N_{x,y} = 2^{n-k-m}$ for all $x \in \mathbb{Z}_2^k$.
- A SAC-fulfilling function is a function for which we have $N_{\Delta x,\Delta y} = 2^{n-m}$ for all $\Delta x$ with $wt(\Delta x) = 1$.
- A function satisfies propagation criterion of degree $k$ (PC-$k$) if $N_{\Delta x,\Delta y} = 2^{n-m}$ for all $\Delta x$ with $wt(\Delta x) \leq k$.

Using the above extended definitions, let criterion "C" be any of the following: balance, correlation immunity, Strict Avalanche Criterion (SAC), higher order SAC, Propagation Criterion (PC), higher order PC, or perfect nonlinearity. Then we have
Theorem 5.6

If \( y \) is the output of a multi-output boolean function then \( y \) satisfies criterion \( C \) if and only if every nonzero linear combination of its output coordinates satisfies criterion \( C \).

Proof: For balanced functions we have \( N_y = 2^{n-m} \). By noting that \( F_0(0) = 2^{n/2} \), then we have

\[
N_y = 2^{n/2 - m} \sum_c F_c(0)(-1)^{c \cdot y}
= 2^{n-m} + 2^{n/2 - m} \sum_{c \neq 0} F_c(-1)^{c \cdot y},
\]

and hence

\[
\sum_{c \neq 0} F_c(0) (-1)^{c \cdot y} = 0. \tag{5.52}
\]

As the equation above should be satisfied for all \( y \in \mathbb{Z}_2^m \) then we must have \( F_c(0) = 0 \) for all \( c \neq 0 \) which implies that the function \( c \cdot f, c \neq 0 \) is a balanced function.

Similarly for \( k \)-resilient functions we have

\[
N_{\hat{x}_y} = 2^{n-k-m} = 2^{n/2-m-k} \sum_{\hat{w}} F_c(\hat{w}^o)(-1)^{\hat{w} \cdot \hat{x} + c \cdot y}. \tag{5.53}
\]

By noting that

\[
F_0(\hat{w}^o) = 2^{n/2} \delta(0), \tag{5.54}
\]

then the following equation must be satisfied for all \( y \in \mathbb{Z}_2^m \)

\[
\sum_{\hat{w}, c \neq 0} F_c(\hat{w}^o)(-1)^{\hat{w} \cdot \hat{x} + (-1)^{c \cdot y} = 0. \tag{5.55}
\]

This can be achieved if and only if

\[
F_c(\hat{w}^o) = 0, \hat{w} \in \mathbb{Z}_2^k. \tag{5.56}
\]
In other words, we must have $F_c(w) = 0$ for $k \geq wt(w) \geq 1$.

For SAC fulfilling functions we have for $wt(\Delta x) = 1$

$$N_{\Delta x \Delta y} = 2^{-m} \sum_{c,w} F^2_c(w)(-1)^{\Delta x \cdot w + c \cdot \Delta y}$$

$$= 2^{-m} \sum_{c,w} F^2_c(w)(-1)^{w_i}(-1)^{c \cdot \Delta y} = 2^n - m. \quad (5.57)$$

Substituting with the value of $F_0(w) = 2^{n/2} \delta(0)$ we get

$$\sum_{c \neq 0} \left( \sum_w F^2_c(w)(-1)^{w_i} \right)(-1)^{c \cdot \Delta y} = 0. \quad (5.58)$$

As the above equation should hold for all $\Delta y \in \mathbb{Z}_2^m$ then we must have

$$\sum_w F^2_c(w)(-1)^{w_i} = 0, \quad i \in \{1, 2, ..., n\}, c \neq 0. \quad (5.59)$$

This means that a multi-output function fulfills SAC if and only if every non-zero linear combination of its output coordinates fulfills SAC.

The rest of the theorem follows using a similar argument.

\[ \square \]

5.7 Conclusion

Many of the previously known cryptographic criteria are related to information leakage. Most of these criteria require zero information leakage in some domain. However, they often constrain the function to such an extent that large information leakage of other types become likely. These leakages provide useful information for the cryptanalyst to develop attacks on the cipher. This motivates the minimization of information leakage as a general criterion for cryptographic functions.

We have derived expressions for the expected values of the static and dynamic information leakage of randomly selected boolean functions and for randomly selected balanced, and injective boolean functions. Based on this we showed that the expected values of the
information leakages decrease dramatically with the number of input variables. In some cases, we showed that this decrease is exponential. With the same approach developed in this chapter, one can show that the variance of different forms of information leakage also decreases dramatically with the number of input variables. In fact, one can also show that the expected maximum value of different forms of information leakage decrease with the number of input variables. This indicates that cryptographically strong boolean functions may be obtained by choosing random mappings of sufficiently large dimensions.

The generalized result in theorem (5.6) confirms that the cryptographic strength of a multi-output boolean function depends on the strength of every nonzero linear combination of its output coordinates.
Chapter 6 A New Class of Substitution-Permutation Networks

6.1 Introduction

Feistel [44] was the first to suggest that a basic substitution-permutation network (SPN) consisting of iterative rounds of nonlinear substitutions (s-boxes) connected by bit permutations was a simple, effective implementation of a private-key block cipher. The SPN structure is directly based on Shannon's principle of a mixing transformation using the concepts of "confusion" and "diffusion" [123]. Letting $N$ represent the block size of a basic SPN consisting of $R$ rounds of $n \times n$ s-boxes, a simple example of an SPN with $N = 16$, $n = 4$, and $R = 3$ is illustrated in Figure 6.1. Keying the network can be accomplished by XORing the key bits with the data bits before each round of substitution and after the last round. The key bits associated with each round are derived from the master key according to the key scheduling algorithm.

![Figure 6.1 SPN with $N = 16$, $n = 4$, and $R = 3$.](image)

One advantage of the basic SPN model is that it is a simple, yet elegant, structure for which it is generally possible to prove security properties. Indeed, it has been shown that a basic SPN can be constructed to possess good cryptographic properties such as
completeness or nondegeneracy [66], adherence to the avalanche criterion [61], and resistance to differential and linear cryptanalysis [60].

The basic SPN architecture differs from a DES-like architecture in which the substitutions and permutations, used as a mixing transformation, operate on only half of the block at a time. Since SPNs do not have this last property, in general, SPNs need two different modules for the encryption and the decryption operations. In an SPN, decryption is performed by running the data backwards through the inverse network (i.e., applying the key scheduling algorithm in reverse and using the inverse s-boxes and the inverse permutation layer). In a DES-like cipher, the inverse s-boxes and inverse permutation are not required. Hence, a practical disadvantage of the basic SPN architecture compared with the DES-like architecture is that both the s-boxes and their inverses must be located in the same encryption hardware or software. The resulting extra memory or power consumption requirements may render this solution less attractive in some situations especially for hardware implementations.

One proposal to overcome this problem is to use a single s-box and its inverse for both the encryption and the decryption. This idea was employed in SAFER [79]. Unfortunately, in SAFER, the encryption and the decryption are different and one still needs two different hardware modules.

In this chapter, we introduce a special class of substitution-permutation networks. This class has the advantage that the same network can be used to perform both the encryption and the decryption operations. The basic idea is to use involution substitution layers and involution permutation layers or linear transformations. We investigate the resistance of these networks to both differential and linear cryptanalysis: it is shown that using an appropriate linear transformation between rounds is effective in improving the security of the SPNs in relation to these two attacks. Further results suggest that the cyclic properties of the overall network are not negatively influenced by the cyclic properties of the involution s-boxes. As well, a key scheduling algorithm is proposed that has
the advantages of preventing weak keys and ensuring that, given that key bits in a particular round are compromised, it is hard to get any information about the key bits of other rounds.

6.2 S-boxes

2.1 Semi-Involution Functions

It is possible to construct SPNs which do not require inverse s-boxes if the s-boxes in the network belong to the class of functions that we refer to as semi-involution functions. Such functions have the property that their inverses can be easily obtained by a simple XOR operation on the function input and output. Hence, differences between the s-boxes in the encryption network and the decryption network can be accommodated by incorporating the XOR into the application of the round key bits.

Definition: A bijective function \( \pi : Z^n_2 \rightarrow Z^n_2 \) is called a semi-involution function if

\[
\pi^{-1}(x) = \pi(x \oplus a) \oplus b
\] (6.1)

for some constants \( a, b \in Z^n_2 \).

Involution functions are the sub-class of semi-involution functions for which \( a = b = 0 \).

Lemma 6.1

A semi-involution function as defined above has \( a \oplus b \) as a linear structure.

Proof: Let \( y = \pi^{-1}(x) \) and, from (6.1), we have \( y \oplus b = \pi(x \oplus a) \). Therefore, \( x = \pi^{-1}(y \oplus b) \oplus a \). Hence, \( \pi(y) = \pi^{-1}(y \oplus b) \oplus a \). Now replacing \( y \oplus b \) with \( x \) gives \( \pi(x \oplus b) = \pi^{-1}(x) \oplus a \). From (6.1), \( \pi(x \oplus a) \oplus b = \pi(x \oplus b) \oplus a \). Replacing \( x \) with \( x \oplus b \) gives

\[
\pi(x \oplus a \oplus b) = \pi(x) \oplus a \oplus b,
\] (6.2)

which is the definition of a linear structure [43], [84].
Thus a semi-involution function has $N_{\Delta x \Delta y} = 2^n$ where $N_{\Delta x \Delta y}$ is the XOR difference distribution table entry[15] for input $\Delta x = a \oplus b$ and $\Delta y = a \oplus b$. For $a \oplus b \neq 0$ this renders the SPN trivially broken by differential cryptanalysis. This means that, if we want to use the same SPN for both the encryption and decryption, then only semi-involution s-boxes with $a = b$ can be used.

The following lemma shows how the useful class of semi-involution functions can be obtained from involution functions.

**Lemma 6.2**

Let $\phi : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ be an involution function, then the function $\pi(x) = \phi(x) \oplus a$ is a semi-involution function such that $a = b$, i.e., $\pi^{-1}(x) = \pi(x \oplus a) \oplus a$.

**Proof:** From the definition of involution functions, $\phi^2(x) = x$. Hence, $\pi(\pi(x) \oplus a) \oplus a = x$. Replacing $x$ with $x \oplus a$ gives $\pi(x \oplus a) \oplus a = \pi^{-1}(x)$.

Lemma (6.2) is important, not only because it provides an easy way to generate the useful class of semi-involution functions from involution functions, but also because it implies that the functions $\phi(x)$ and $\pi(x)$ belong to the same cryptographic class and hence they have the same linear approximation table [82], and the same XOR difference distribution table [15].

The only cryptographic difference between involution s-boxes and semi-involution s-boxes with $a = b$, $a \neq 0$, is their cyclic properties. All cycles of involution functions have length one or two. In SPNs where the key bits are XORed with the data bits at the s-box input, if we assume that all the key bits are equi-probable, then both the SPNs built using semi-involution s-boxes with $a = b \neq 0$ and the SPNs built using involution s-boxes will have the same cryptographic properties. In the rest of the chapter we will focus on the class of SPNs that use involution s-boxes.
An interesting class of involution mappings is the inversion mapping in $GF(2^n)$ defined as [91]:

$$
\pi(x) = \begin{cases} 
  x^{-1}, & x \neq 0 \\
  0, & x = 0.
\end{cases}
$$

(6.3)

Different cryptographic properties of this mapping were studied in [91]. This inversion mapping is differentially 2-uniform if $n$ is odd and it is differentially 4-uniform if $n$ is even. The nonlinearity of this mapping is given by $\mathcal{NL}(\pi) \geq 2^{n-1} - 2^{n/2}$.

The above class of s-boxes can be generated using different irreducible polynomials. The number of monic polynomials of degree $n$ which are irreducible over $GF(q)$, where $q$ is any prime power, is given by [11], [78]:

$$
\frac{1}{n} \sum_{d|m} \mu(d)q^{m/d}
$$

(6.4)

where $\mu(d)$ is the Möbius function given by

$$
\mu(d) = \begin{cases} 
  1, & d = 1 \\
  (-1)^r, & d \text{ is a product of } r \text{ distinct primes} \\
  0, & \text{otherwise.}
\end{cases}
$$

(6.5)

For $n = 8$, we have 30 irreducible polynomials of degree 8 and hence we can generate 30 such s-boxes. All these 30 s-boxes have nonlinearity equal to 112 and maximum XOR table entry equal to 4. In order to frustrate possible algebraic attacks, the SPN should use s-boxes generated using different irreducible polynomials. Another approach is to use randomly generated s-boxes so that the overall cipher would not have any easy algebraic description. In section 2.3 we study some of the cryptographic properties of such randomly generated involution s-boxes.
Lemma 6.3

The number of involution functions $\pi : Z_2^n \rightarrow Z_2^n$ is given by

$$
\sum_{i=0}^{2^{(n-1)}} \frac{2^n!}{(2^{n-1} - i)! (2i)! 2^{2^{n-1}-i}}.
$$

Proof: An involution function can only have an even number of fixed points. There are $\binom{2^n}{2i}$ ways to specify any of these $2i$ fixed points. Note also that an involution function with $2i$ fixed points must have $2^{n-1} - i$ cycles of length 2. An involution function is completely defined by specifying its fixed points and a single point on each of its $2^{n-1} - i$ cycles. Now, we will count the number of ways of assigning these $2^n - 2i$ points along the $2^{n-1} - i$ cycles. To choose the first point, pick any arbitrary point $x_0 \in Z_2^n$ such that $x_0$ is not equal to any of the assigned fixed points. Choose a random value $r_0 \in Z_2^n$ for $\pi(x_0)$. $r_0$ should not be equal to any of the fixed points. It also should not be equal to $x_0$. Thus there are $(2^n - 2i - 1)$ ways to choose $r_0$. To choose a second point, pick another arbitrary point $x_1$ such that $\pi(x_1)$ has not been assigned yet (this also ensures that it belongs to a new distinct cycle) and pick a random $r_1 \in Z_2^n$ for $\pi(x_1)$. Again, $r_1$ should satisfy the following conditions: $r_1 \neq x_1$ and it should not be equal to any of the previously assigned values for $\pi$. Proceeding as above, we have

$$
\prod_{j=0}^{2^{n-1}-i-1} (2^n - 2i - 1 - 2j) = \frac{(2^n - 2i)!}{(2^{n-1} - i)! 2^{2^{n-1}-i}}.
$$

---

1. The number of fixed points of $\pi : Z_2^n \rightarrow Z_2^n$ is equal to $\# \{x \in Z_2^n | \pi(x) = x\}$.

2. A cycle of $\pi : Z_2^n \rightarrow Z_2^n$ is a sequence of elements $x_1, x_2, \ldots, x_L$ such that $x_{i+1} = \pi(x_i)$ for $1 \leq i \leq (L-1)$, and $x_1 = \pi(x_L)$, but $x_i \neq \pi(x_l)$ for $2 \leq l \leq (L-1)$. The length of the cycle is $L$. Let $\phi(x)$ be the length of the cycle to which $x$ belongs, and let $C_1, C_2, \ldots, C_\phi$ be the distinct cycles of $\pi$. Then the average cycle length is given by

$$
\frac{1}{2^n} \sum_{x \in Z_2^n} \phi(x) = \frac{1}{2^n} \sum_{i=1}^{\phi} |C_i|^2,
$$

where $|C_i|$ denotes the length of $C_i$. 

82
ways of assigning these points. Hence, the number of involution functions is given by

\[
\binom{2^n}{2^n} + \sum_{i=0}^{2^n-1} \binom{2^n}{2i} \prod_{j=0}^{2^n-i-1} (2^n - 2i - 1 - 2j) \nonumber
\]

\[
= \sum_{i=0}^{2^n-1} \binom{2^n}{2i} \frac{(2^n - 2i)!}{(2^n-1-i)! \cdot 2^{2^n-1-i}} \tag{6.8}
\]

\[
= \sum_{i=0}^{2^n-1} \frac{2^n!}{(2^n-1-i)! \cdot (2i)! \cdot 2^{2^n-1-i}}
\]

which proves the lemma.

\[\square\]

### 2.2 Equivalence Classes

Two s-boxes \(\pi_1, \pi_2\) are said to belong to the same cryptographic class [130] if

\[
\pi_2(x) = \pi_1(x \oplus a) \oplus b
\]

for arbitrary constants \(a, b \in \mathbb{Z}_2^n\).

The use of s-boxes within the same cryptographic classes was suggested as a means to design SPNs that are resistant to differential cryptanalysis [130]. Unfortunately, involution s-boxes cannot be used in such SPNs because, as shown in the following lemma, if two involution s-boxes belong to the same cryptographic class then they possess a linear structure.

\textbf{Lemma 6.4}

If \(\pi_1\) and \(\pi_2\) are both involution mappings and

\[
\pi_2(x) = \pi_1(x \oplus a) \oplus b
\]

then \(\pi_1, \pi_2\) have \(a \oplus b\) as a linear structure.

---

3 The first term in the equation above stands for the unity bijection mapping with \(2^n\) fixed points.
Proof: By noting that \( \pi_2(x) = \pi_1(x \oplus a) \oplus b \) then we have \( \pi_2^2(x) = \pi_1(\pi_1(x \oplus a) \oplus a \oplus b) \oplus b \). But we also have \( \pi_2^2(x) = x \) and, hence, \( \pi_1(\pi_1(x \oplus a) \oplus a \oplus b) \oplus b = x \). Thus, we have \( \pi_1(x \oplus a) \oplus a \oplus b = \pi_1^{-1}(x \oplus b) \).
Replacing \( x \oplus b \) by \( x \) and noting that \( \pi_1^{-1}(x) = \pi_1(x) \) gives

\[
\pi_1(x \oplus a \oplus b) = \pi_1(x) \oplus a \oplus b
\]

which is the definition of a linear structure. By a similar argument, one can show that \( \pi_2 \) also has \( a \oplus b \) as a linear structure.

\[ \square \]

2.3 Number of Fixed Points

Involution s-boxes have the characteristic that all cycles are of length one or two and, as will be shown, have a larger expected number of fixed points than a randomly chosen s-box. Although there is no known effective cryptanalytic attack directly based on the existence of fixed points in the s-boxes, it is of interest to determine if a large number of fixed points affects other cryptographic properties, such as the nonlinearity and the maximum XOR table entry, that lead to other cryptographic attacks.

Figure 6.2 shows the experimental results for the average nonlinearity and the average maximum XOR table entry as a function of the number of fixed points for 8-bit random bijections and 8-bit random involutions. One thousand random bijective s-boxes and one thousand random involution s-boxes were tested for each point. The graphs were derived by incrementing the number of fixed points by 2. The graphs clearly indicate a strong correlation between the cryptographic properties and the number of fixed points and suggest that the s-boxes should be chosen to contain few fixed points.
Figure 6.2 Average Nonlinearity and Average Maximum XOR Table Entry Versus the Number of Fixed Points (n = 8)

We now calculate the expected number of fixed points for a random bijection and for a random involution.

**Lemma 6.5**

The expected value of the number of fixed points for a random bijective mapping is 1.

**Proof:** The number of bijective mappings with exactly $t$ fixed points is given by (this result follows by using the inclusion-exclusion principle)

$$
\sum_{i=t}^{n} (-1)^{i-t} \binom{i}{t} \binom{2^n}{i} (2^n - i)!.
$$

(6.12)

The probability of having exactly $t$ fixed points is given by the above formula divided by $2^n!$.

Hence, the expected number of fixed points is given by
The last step in the equation above follows by noting that
\[
\sum_{i=0}^{i} (-1)^t \cdot t \left( \begin{array}{c} \text{i} \\ \text{t} \end{array} \right) = \begin{cases} -1 & \text{if } \text{i} = 1, \\ 0 & \text{otherwise.} \end{cases}
\] (6.14)

which completes the proof of the lemma. \(\square\)

Similarly, one can show that the variance of the number of fixed points is also 1.

**Lemma 6.6**

The expected number of fixed points for a random involution mapping is given by
\[
E(N_{fp}) = \sum_{i=0}^{2n-1} 2i \Phi(n, i) \left/ \sum_{i=0}^{2n-1} \Phi(n, i) \right.
\] (6.15)

where
\[
\Phi(n, i) = \frac{2^i}{(2n-1-i)! (2i)!}.
\] (6.16)

**Proof:** From the proof of lemma (6.3), the number of involution functions with \(2i\) fixed points is given by
\[
\frac{2^n!}{(2n-1-i)! (2i)! 2^{2n-1-i}}, \quad 0 \leq i \leq 2^{n-1}.
\] (6.17)
The probability of randomly selecting an involution function with $2i$ fixed points is obtained by dividing (6.17) by the total number of involution functions. Thus, the expected number of fixed points for a random involution function is given by

$$
\frac{2^n!}{\sum_{i=0}^{2^n-1} \frac{2^n!}{(2^{n-1}-i)! (2i)!} 2^{n-1-i}} = \frac{2^n!}{\sum_{i=0}^{2^n-1} \frac{2^n}{(2^{n-1}-i)! (2i)!}}.
$$

(6.18)

which completes the proof of the lemma.

Numerical substitution in the formula above shows that the expected number of fixed points of a random involution exceeds that of a random injective mapping by a large factor. For example, an 8-bit involution mapping is expected to have about 16 fixed points. Fortunately, the construction proof of lemma (6.3) can be used to generate involution functions with a predetermined number of fixed points. A special case of interest is involution functions with zero fixed points since this seems to optimize their cryptographic properties (see Figure. 6.2). The number of such functions follows from the proof of lemma (6.3) and can be approximated using Stirling’s formula as follows

$$
\frac{2^n!}{2^{n-1}! 2^{2n-1}} \approx \sqrt{2\left(\frac{2^n}{e}\right)^{2^n-1}}.
$$

(6.19)

### 6.3 S-box Interconnection Layer

In order to use the same SPN to perform both the encryption and the decryption operations, the s-box inter-connection layer should also be an involution mapping. One permutation layer, applicable to networks for which $N = n^2$, with nice cryptographic properties [60] and which satisfies the involution requirement is described by: output bit $i$ of s-box $j$ at round $r$ is connected to input bit $j$ of s-box $i$ at round $r + 1$. 87
In [60] it was shown that with such a permutation layer we can develop upper bounds on the differential characteristic probability [15] and on the probability of a linear approximation [82] as a function of the number of rounds of substitution. Unfortunately, to achieve good bounds, with a relatively small number of rounds, it is suggested to have s-boxes with a large diffusion order [60]. Letting $\Delta x$ and $\Delta y$ denote the input change vector and the output change vector, respectively, an s-box satisfies diffusion order of $\lambda$. $\lambda \geq 0$, if for $wt(\Delta x) > 0$,

$$wt(\Delta y) = \begin{cases} 
\lambda & wt(\Delta x) < \lambda + 1, \\
0 & otherwise.
\end{cases}$$

(6.20)

where $wt(\cdot)$ denotes the Hamming weight of the enclosed argument.

Using the depth-first search algorithm proposed in [60] we could not find any $8 \times 8$ involution s-boxes with diffusion order greater than 1 (without the involution constraint, some $8 \times 8$ s-boxes with $\lambda = 2$ were found in [60]). As an alternative to this, the authors in [60] proposed the use of an invertible linear transformation between rounds. The SPN resistance to linear and differential cryptanalysis was very encouraging. Unfortunately, their proposed linear transformation is not very attractive in practice as it requires a bit XORing operation of all the output bits of the round.

### 6.3.1 Proposed Linear Transformation

We propose a more efficient linear transformation that runs much faster. Moreover it has improved bounds for the linear approximation and the differential characteristic. The linear transformation between rounds of s-boxes is described by

$$z_i = \bigoplus_{l=1,l \neq i}^{M} w_l, \ 1 \leq i \leq M$$

(6.21)

where $z_i$ represents the $i^{th}$ $n$-bit output word of the transformation, $w_i$ is the $i^{th}$ input word, $M = \frac{N}{n}$ denotes the number of s-boxes, and $\oplus$ denotes a bit-wise XOR operation. It is assumed that $M$ is even. For $8 \times 8$ s-boxes this is a byte oriented operation. One can easily check that this linear transformation operation is an involution.
The linear transformation described above may be efficiently implemented by noting that each \( z_i \) could be simply determined by XORing \( w_i \) with the XOR sum of all \( z_j \), \( 1 \leq j \leq M \), i.e.,

\[
z_i = q \oplus w_i, \tag{6.22}
\]

where

\[
q = \bigoplus_{l=1}^{M} w_l. \tag{6.23}
\]

Equation (6.23) above requires \((M - 1)\) word-oriented XORs (which can be done in parallel in \( \log_2 M \) steps) and equation (6.22) requires \( M \) word-oriented XORs (which can be done in one step). Hence for a 64-bit SPN using \( 8 \times 8 \) s-boxes, the above linear transformation requires \( 7 + 8 = 15 \) byte-oriented XORs compared to \( 63 + 64 = 127 \) bit-oriented XORs required for the linear transformation of [60].

6.3.2 Involution Linear Transformations based on MDS codes

An interesting class of linear transformations is the one based on Maximum Distance Separable (MDS) codes [78]. The use of such linear transformations was first proposed in [133] and then utilized in the cipher SHARK [108] and later in the cipher SQUARE [34]. This class of linear transformations has the advantage that the number of s-boxes involved in any 2 rounds of a linear approximation or in any 2 rounds of a differential characteristic is equal to \( M + 1 \), which is the maximum theoretically possible number.

In this section we provide two construction methods for involution linear transformations based on Maximum Distance Separable Codes.

Rijmen et al [108] noted that the framework of linear codes over \( GF(2^n) \) provides an elegant way to construct the linear transformation layer. More details about the theory of error correcting codes can be found in [78].

Let \( C \) be a \((2M, M, d)\) code over \( GF(2^n) \). Let \( G = [I|A] \) be the generator matrix in echelon form where \( A \) is a nonsingular \( M \times M \) matrix and \( I \) is the \( M \times M \) identity
matrix. Then $C$ defines an invertible linear mapping

$$GF(2^n)^M \rightarrow GF(2^n)^M : X \rightarrow Y = AX. \quad (6.24)$$

If the matrix $A$ is used in the implementation of the linear transformation of the SPN, then it is easy to see that the number of s-boxes involved in any 2 rounds of a differential characteristic or a linear approximation expression is lower bounded by $d$, the minimum distance of the code [108]. The minimum distance of the code is equal to the minimum number of linearly dependent columns in its null matrix (also known as the parity-check matrix).

**Lemma 6.7 [78]:**
An $(n, k, d)$ code with generator matrix $G = [I|A]$, where $A$ is a $k \times (n - k)$ matrix, is MDS if and only if every square submatrix (formed from any $i$ rows and any $i$ columns, for any $i = 1, 2, \ldots, \min\{k, n - k\}$) of $A$ is nonsingular.

**Random Construction**

One way to obtain an involution matrix $A$ which satisfies the above constraint is to pick a random involution matrix and test it for the condition in lemma (6.7).

Let

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (6.25)$$

be an $M \times M$ random matrix where $A_{11}, A_{12}, A_{21}$ and $A_{22}$ are $\frac{M}{2} \times \frac{M}{2}$ matrices. An involution matrix is one which satisfies $A^2 = I$, and thus $A$ is an involution iff

$$A_{11}A_{12} \oplus A_{12}A_{22} = 0, \quad (6.26)$$

$$A_{11}^2 \oplus A_{12}A_{21} = I, \quad (6.27)$$

$$A_{21}A_{11} \oplus A_{22}A_{21} = 0, \quad (6.28)$$

90
If we let $A_{22} = A_{11}$ then equation (6.26) is satisfied iff $A_{11}$ and $A_{12}$ commute with each other. To achieve this we let $A_{12} = A_{11}^{-1}$. For these choices of $A_{12}$ and $A_{22}$, equations (6.27), (6.28) and (6.29) are linearly dependent with the solution $A_{21} = A_{11}^3 \oplus A_{11}$.

Thus the $M \times M$ matrix

$$A = \begin{bmatrix} A_{11} & A_{11}^{-1} \\ A_{11}^3 \oplus A_{11} & A_{11} \end{bmatrix},$$

(6.30)

where $A_{11}$ is a random nonsingular $M/2 \times M/2$ matrix, is an involution over $GF(2^n)$.

For $n = 8$, random search for a matrix $A$, with the structure in equation (6.30), that satisfies the condition in lemma (6.7), terminates within a few seconds for even values of $M$, $M \leq 6$. For $M = 8$ we were unable to obtain any matrix that satisfies the conditions in lemma (6.7) by random search. Table 6.1 [143] shows the experimental results for the minimum distance distribution for $10^5$ randomly selected invertible linear transformations and $10^5$ randomly selected invertible involution linear transformation in the form of equation (6.30).

<table>
<thead>
<tr>
<th>$d$</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental (Random)</td>
<td>46</td>
<td>15539</td>
<td>84415</td>
</tr>
<tr>
<td>Experimental (Involution)</td>
<td>118</td>
<td>7714</td>
<td>92168</td>
</tr>
</tbody>
</table>

Table 6.1 Minimum Distance for $10^5$ Randomly Selected Involution Linear Transformations ($n = M = 8$)

**Algebraic Construction**

In this section we show how to obtain an involution matrix satisfying lemma (6.7) by a simple algebraic construction.
Lemma 6.8 [78]:

Given \( x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1} \) the matrix \( A = [a_{ij}], 0 \leq i, j \leq n - 1 \) where \( a_{ij} = \frac{1}{x_i + y_j} \) is called a Cauchy matrix. It is known that

\[
\det(A) = \prod_{0 \leq i < j \leq n-1} \frac{(x_j - x_i)(y_j - y_i)}{(x_i + y_j)}.
\]  

(6.31)

Hence, provided the \( x_i \) are distinct, the \( y_i \) are distinct, and \( x_i + y_j \neq 0 \) for all \( i, j \), it follows that any square submatrix of a Cauchy matrix is nonsingular over any field.

Let

\[
x_i = i, \quad y_i = i \oplus r.
\]  

(6.32)

where

\[
i = (00 \cdots 0 i_{\log_2(M) - 1} \cdots i_{1} i_{0}) \in GF(2^n), \quad \sum_{l=0}^{\log_2(M) - 1} 2^l i_l = i.
\]  

(6.33)

and the least significant \( [\log_2(M)] \) bits of \( r \neq 0 \) are zeros.

For \( A^2 = H = [h_{ij}] \) we have

\[
h_{ij} = \bigoplus_{k=0}^{M-1} \frac{1}{(i \oplus k \oplus r)(j \oplus k \oplus r)} = \begin{cases} 
\frac{1}{(k \oplus r)^2}, & i = j \\
0, & i \neq j
\end{cases},
\]  

(6.34)

where \( i, j \) and \( k \) are evaluated as in equation (6.33). Thus the matrix \( A \) will satisfy

\[
A^2 = c^2 I, \quad c = \bigoplus_{i=1}^{n} a_{ii} \text{ over } GF(2^n).
\]

Dividing (division over \( GF(2^n) \)) each element of \( A \) by

\[
\sqrt{c} = \bigoplus_{k=0}^{M-1} \frac{1}{(k \oplus r)} = \bigoplus_{i=1}^{n} a_{ii},
\]  

(6.35)
we obtain an involution matrix for which every square submatrix is nonsingular over $GF(2^n)$. Figure 6.3 shows an example for $M = n = 8$, using the irreducible polynomial 11d$^4$.

\[
\begin{array}{cccccccc}
93 & 13 & 57 & da & 58 & 47 & c & 1f \\
13 & 93 & da & 57 & 47 & 58 & 1f & c \\
57 & da & 93 & 13 & c & 1f & f8 & 47 \\
da & 57 & 13 & 93 & 1f & c & 47 & 58 \\
58 & 47 & c & 1f & 93 & 13 & 57 & da \\
47 & 58 & 1f & c & 13 & 93 & da & 57 \\
c & 1f & 58 & 47 & 57 & da & 93 & 13 \\
1f & c & 47 & 58 & da & 57 & 13 & 93 \\
\end{array}
\]

*Figure 6.3* Involution Linear Transformation Based on MDS Codes ($M = n = 8$, Irreducible Polynomial = 11d)

The resistance of SPNs using this MDS linear transformation is studied in [108] and [143]. Throughout the rest of this chapter we will concentrate on the class of linear transformations described by equation (6.21).

### 6.4 Resistance to Differential and Linear Cryptanalysis

Using an approach similar to the analysis in [60], it is possible to establish upper bounds on the most likely differential characteristic and linear approximation expression using the linear transformation of (6.21). The results of this section are obtained by assuming that all the round keys are independent.

#### 4.1 Differential Cryptanalysis

The following lemma gives a lower bound on the number of s-boxes involved in any 2 rounds of a differential characteristic.

---

$^4$ All numbers are in hexadecimal format
**Lemma 6.9**

Consider an SPN with $M$ s-boxes, $M \geq 4$. If the SPN employs the linear transformation described in (6.21), then the number of s-boxes involved in any 2 rounds of a differential characteristic is greater than or equal to 4.

**Proof:** From the linear transformation expression one can check that if only one s-box is involved in round $r$ this implies that $M - 1$ s-boxes are involved in round $r + 1$. If 2 s-boxes are involved in round $r$, (6.21) ensures that at least 2 s-boxes will be involved in round $r + 1$. The rest of the proof follows by noting that the minimum number of s-boxes involved per round is 1.

The number of chosen plaintext/ciphertext pairs required for differential cryptanalysis of an $R$ round SPN (based on the best characteristic and not the best differential [94],[73]) may be approximated by [15], [60]

$$N_D = \frac{1}{P_{\Omega_{R-1}}} \quad (6.36)$$

where $P_{\Omega_{R-1}}$ is the probability of the best $R - 1$ round characteristic. This probability can be bounded by

$$P_{\Omega_{R-1}} \leq (P_b)^\alpha \quad (6.37)$$

where the maximum s-box XOR pair probability is given by $P_b = \frac{XOR^*}{2^n}$ with $XOR^*$ denoting the maximum entry in the XOR distribution tables of the s-boxes used in the SPN and $\alpha$ is the total number of s-boxes involved in the characteristic. For even $R$, from lemma 6.9 and assuming that only one s-box will be involved in round $R - 1$ then we have

$$\alpha \geq 4 \left( \frac{R}{2} - 1 \right) + 1 = 2R - 3, \quad (6.38)$$

and, hence,

$$N_D \geq \frac{1}{(P_b)^{2R-3}}. \quad (6.39)$$
Using $8 \times 8$ involution s-boxes with maximum XOR table entry of 10 (easily found by randomly selecting involution s-boxes), an 8 round 64-bit SPN that utilizes the proposed linear transformation will have $N_D \geq 2^{60.8}$ chosen plaintext/ciphertext pairs required for differential cryptanalysis. If we use the inversion s-boxes given by (6.3), then we will have $N_D \geq 2^{78}$.

4.2 Linear Cryptanalysis

The following lemma gives a lower bound on the number of s-boxes involved in any 2 round linear approximation and is based on the assumption of independence between linear approximation of different rounds.

**Lemma 6.10**

Consider an SPN with $M$ s-boxes, $M \geq 4$. If the SPN employs the linear transformation described in (6.21) then the number of s-boxes involved in any 2 rounds of a linear approximation is greater than or equal to 4.

**Proof:** If the number of s-boxes involved in round $r + 1$, $l$, is odd, then the number of s-boxes involved in round $r$ is $M - l$. If $l$ is even, then the number of s-boxes involved in round $r$ is $l$. The lemma above follows by considering different values for $l$. □

For an SPN based on $n \times n$ s-boxes, the number of known plaintexts required for the basic linear cryptanalysis (algorithm 1 in [82]) may be approximated by [60]

$$N_L = \frac{1}{|P_L - \frac{1}{2}|^2}$$

(6.40)

where

$$|P_L - \frac{1}{2}| \leq 2^{\alpha - 1}(P_t)\alpha$$

(6.41)
and
\[ P_e = \left( \frac{2^{n-1} - N_L}{2^n} \right). \]  

(6.42)

with \( N_L \) denoting the minimum nonlinearity [90] of the s-boxes used in the SPN and \( \alpha \) is the total number of s-boxes involved in the linear approximation. From the above argument we have

\[ \alpha \geq 4 \left( \frac{R}{2} \right) = 2R. \]  

(6.43)

and, hence,

\[ N_L \geq \frac{1}{2^{4R-2} P_e^{4R}}. \]  

(6.44)

Using 8 \( \times \) 8 involution s-boxes with nonlinearity of 98 (easily found by randomly selecting involution s-boxes), an 8 round 64-bit SPN that utilizes the proposed linear transformation will have \( N_L \geq 2^{68.98} \) known plaintext/ciphertext pairs required for the basic linear attack. Since this number is greater than the size of the plaintext set, we interpret this to mean that the basic linear attack is not effective against this class of SPNs, even if we use all possible plaintexts. If we use the inversion s-boxes given by (6.3), then we will have \( N_L \geq 2^{98} \).

### 6.5 Cyclic Properties of the Proposed SPN

A significant difference between an involution s-box and a non-involutioin s-box is likely to be their cyclic properties. For a randomly chosen \( n \)-bit bijective mapping, the expected value and the variance of the number of cycles are both approximately equal to \( \log_e(2^n) \approx 0.69n \) [46]. The expected value of the cycle length is equal to \( 2^{n-1} + 1/2 \) [35]. For an involution mapping with \( N_{fp} \) fixed points, the expected cycle length is given by

\[ \frac{N_{fp} \times 1 + (2^n - N_{fp}) \times 2}{2^n} = 2 - \frac{N_{fp}}{2^n}. \]  

(6.45)

and the number of cycles is given by \( 2^{n-1} + N_{fp}/2 \).

96
In order to investigate whether the cyclic properties of involution s-boxes affect the cyclic properties of the SPN, we measured the cycle distribution for 100,000 16-bit SPNs with 4-rounds. Each SPN uses four $4 \times 4$ random involution s-boxes with zero fixed points, nonlinearity greater than or equal to 4 and maximum XOR table entry equal to 4. The cycle length distribution is shown in Figure 6.4 (a) (the dark line shows the average distribution over 100 adjacent points). The normalized frequency of occurrence is equal to the actual number divided by 100,000. In this case, the average cycle length over all SPNs is equal to 32779. Figure 6.5 shows a typical expanded segment of the cycle length distribution in Figure 6.4 (a).

We performed the same experiment on 100,000 SPNs using random bijective mappings with the same constraints on the nonlinearity and the XOR table. The simulation results are shown in Figure 6.4 (b). The average cycle length over all SPNs is equal to 32766. It is clear that the two distributions are almost indistinguishable. This suggests that the involution s-boxes do not have a negative impact on the cyclic properties of the SPN.

### 6.6 Key Scheduling Algorithm

In our discussion, we assume that the SPN is keyed by a simple key mixing operations of the key bits before each substitution and after the last substitution (details will be...
explained below). A weak key, $k_w$, is any key for which $E_{k_w}(E_{k_w}(p)) = p$ for every plaintext vector $p$ where $E_{k_w}(-)$ denotes the encryption operation using the key $k_w$. In this section we propose a simple key scheduling algorithm for the SPN. Three design principles were employed:

(i) Prevent weak keys.

(ii) Given that some or all of the key bits at round $r$ are compromised, it is hard to get any information about the other round keys.

Although the above key scheduling can be controlled to be relatively slow in order to make brute force attack harder [106], it is far easier and more effective to use a larger key. Using a larger key has the advantage that it does not penalize implementations which must change the key often.

In the following algorithm $key$ denotes the user supplied key which is assumed to be of the same length as the block length of the SPN, $E_k^r(p)$ denotes the output of the SPN when it has $p$ as an input, and the round keys are all set to $k$. Consider the key scheduling algorithm shown in Figure 6.6.
One can assign any other arbitrary value to $x_0$. $Op(\cdot)$ denotes any simple operation that guarantees that all $x_i$'s are different. By noting that $E^*_k$ is a bijective mapping for any fixed key then all $k_i$'s will be different which guarantees that we do not have any weak keys. An example of operation $Op(\cdot)$ is the complementing of different bits in $x_0$ for each $i$. Note that we control the key scheduling speed by controlling the number of rounds used in the encryption operation $E^*_k$. Also, this scheme is similar to the scheme proposed in [69].

\[
x_0 = 0;
\]

\[
for \ i = 1 \ to \ (R + 1)
\]

\[
\{
\]

\[
k_i = E^*_{key}(x_{i-1});
\]

\[
x_i = Op(x_{i-1});
\]

\[
}\]

**Figure 6.6 Key Scheduling Algorithm**

It is also worth noting that the above keying scheme does not have the complementation property; this property makes DES susceptible to exhaustive key search of $2^{55}$ rather than $2^{56}$. This scheme also ensures that there are no simply related keys which leads to Biham's related keys attack [13].

The key scheduling described above can be extended to accommodate the case where the user supplied key size is a multiple of the SPN block length (Keys which are not multiples can be padded to be so).
6.7 Performance

While the usefulness of a cryptographic algorithm is based on assumptions about its security, the complexity of the cryptographic function is another feature that should not be overlooked. Table 6.2 shows the relative speed (the larger the number, the faster the cipher) of Q-CAST\(^5\) and SPNs on three platforms: an 8–bit microcontroller (Motorola 6811), a SUN SPARC workstation and a SUN ULTRA workstation. All algorithms operate on a 64–bit blocks and implemented 16 rounds. The speed of the SPN in Table 6.2 is normalized independently for each processor.

In considering these numbers, one should take into account that the proposed SPN is a hardware oriented cipher (while Q-CAST is a software oriented cipher), and the round function of the proposed SPN provides a better degree of security than the round function of Q-CAST.

<table>
<thead>
<tr>
<th>Platform</th>
<th>SPN</th>
<th>Q-CAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motorola 6811</td>
<td>1</td>
<td>0.46</td>
</tr>
<tr>
<td>SUN SPARC-20</td>
<td>1</td>
<td>7.5</td>
</tr>
<tr>
<td>SUN ULTRA-1</td>
<td>1</td>
<td>1.56</td>
</tr>
</tbody>
</table>

Table 6.2 Relative Speed of the Proposed SPN and Q-CAST

6.8 Conclusion

We have presented a special class of SPNs that have the advantage that the same network can be used to perform both the encryption and the decryption operation. The s-boxes used are involution mappings and the permutation layer is replaced by an efficient involution linear transformation layer. In a few seconds on a SPARC-20 workstation, we were able to obtain tens of \(8 \times 8\) involution s-boxes with nonlinearity of 98 and maximum XOR table entry of 10. Using these s-boxes, an 8 round 64–bit SPN that utilizes the

\(^5\) Queen's University version of the CAST encryption algorithm
proposed linear transformation will be resistant to both the basic linear cryptanalysis and to the differential cryptanalysis based on the best \((R - 1)\)-round characteristic. We also confirmed that the cyclic properties of this special class of SPNs reveal no apparent weakness. A key scheduling algorithm which satisfies certain design principles was also proposed.
Chapter 7 Modelling Avalanche Characteristics of Substitution-Permutation Networks Using Markov Chains

7.1 Introduction

An SPN is considered to display good avalanche characteristics if a one bit change in the plaintext input is expected to cause close to half the ciphertext bits to change. Good avalanche characteristics are important to ensure that a cipher is not susceptible to statistical attacks such as clustering attacks [58]. More formally, the avalanche is defined as follows:

A cipher is said to satisfy the avalanche criterion if, for each key, on average half the ciphertext bits change when one plaintext bit is changed. That is, \( E(\text{wt}(\Delta C) \mid \text{wt}(\Delta P) = 1) = N/2 \), where \( \text{wt}(\cdot) \) denotes the Hamming weight of the enclosed argument, \( \Delta C \) and \( \Delta P \) denote the ciphertext and plaintext change vectors, respectively.

![Diagram of an SPN with \( N = 16 \), \( n = 4 \), and \( R = 3 \).]

Figure 7.1 SPN with \( N = 16 \), \( n = 4 \), and \( R = 3 \).
In [61], Heys and Tavares analyzed the avalanche characteristics of SPNs based on two general network models, distinguished by the nature of their permutation layer. In the first model, they considered a network where the permutation between two rounds is modelled as a random variable whose values are equally likely. In the second model, they considered a network which has a specified fixed permutation between rounds. In [56], Heys extended this work and developed a model of the avalanche characteristics of DES-like ciphers.

In this chapter we develop analytical models for the avalanche characteristics of other classes of SPNs. In particular, we consider the following four general network models, distinguished by their linear transformation layers (see Figure 7.1):

**Model A** — In this model we consider SPNs, with $M = n$, in which the linear transformation layer is a permutation layer $\pi \in \Omega$, where $\Omega$ is defined to be the set of permutations for which no two outputs of an s-box are connected to one s-box in the next round.

**Model B** — In this model we consider SPNs in which the linear transformation layer is given by the invertible bitwise linear transformation defined by $v = \pi(L(u))$ where $v = [v_1 v_2 \cdots v_N]$ is the vector of input bits to a round of s-boxes, $u = [u_1 u_2 \cdots u_N]$ is the vector of bits from the previous round output, $L(u) = [l_1(u) \cdots l_N(u)]$, $\pi \in \Omega$, where $\Omega$ is the set of permutations defined in model A, and

$$l_i(u) = \bigoplus_{j \neq i} u_j.$$  

(7.1)

$N$ is assumed to be even so that the linear transformation is invertible.

Note that $l_i(u)$ could be simply determined by noting that

$$l_i(u) = u_i \oplus q,$$  

(7.2)

---

1 This model was also considered by Heys and Tavares but we include it here for completeness.
where

\[ q = \bigoplus_{j=1}^{N} u_j. \quad (7.3) \]

The resistance of SPNs with this class of linear transformation against linear and differential cryptanalysis is studied in [60].

**Model C**— In this model we consider SPNs in which the linear transformation layer is given by the invertible wordwise linear transformation defined by

\[ z_i = \bigoplus_{l=1, l \neq i}^{M} w_l, \quad 1 \leq i \leq M, \quad (7.4) \]

where \( z_i \) represents the \( i^{th} \) \( n \)-bit output word of the transformation, \( w_i \) is the \( i^{th} \) input word, and \( M = \frac{N}{n} \) denotes the number of s-boxes. It is assumed that \( M \) is even so that the linear transformation is invertible. For \( 8 \times 8 \) s-boxes this is a byte oriented operation.

The linear transformation described above may be efficiently implemented by noting that each \( z_i \) could be simply determined by XORing \( w_i \) with the XOR sum of all \( z_j \), \( 1 \leq j \leq M \), i.e.,

\[ z_i = q \oplus w_i, \quad (7.5) \]

where

\[ q = \bigoplus_{j=1}^{M} w_j. \quad (7.6) \]

The resistance of SPNs with this class of linear transformation against linear and differential cryptanalysis is studied in [153].

**Model D**— In this model we consider SPNs in which the linear transformation layer is given by the invertible linear transformation

\[ GF(2^n)^M \rightarrow GF(2^n)^M : X \rightarrow Y = AX, \quad (7.7) \]

where \( A \) is a nonsingular \( M \times M \) matrix for which every square submatrix (formed from any \( i \) rows and any \( i \) columns, for any \( i = 1, 2, \ldots, M \)) is nonsingular. Note that
the matrix $G = [I|A]$, where $I$ is the $M \times M$ identity matrix, is the generator matrix of an $(2M, M, M + 1)$ MDS code. Throughout the rest of this thesis, we will refer to this type of linear transformations as the MDS linear transformation. The resistance of SPNs with this class of linear transformation against linear and differential cryptanalysis is studied in [108].

### 7.2 Convergence in Markov Chains

A Markov chain is ergodic if it is finite, aperiodic and irreducible. A sufficient condition for a Markov chain with $n$ states and a state transition matrix

\[
P = \begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix}
\]

\[\text{def} \ [p_{ji}], 1 \leq i, j \leq n, \quad (7.8)\]

to be aperiodic is that $p_{ii} > 0$ for some $i, 1 \leq i \leq n$, and it is irreducible if for all $i, j$ there exists $r$ such that $P_{ji}^{(r)} > 0, 1 \leq i, j \leq n$, where $P^r = [p_{ji}^{(r)}]$.

If $P$ is ergodic, then there exists a unique distribution $\Pi = (\pi_1, \pi_2, \ldots, \pi_n)$ such that

\[
\pi_i = \lim_{r \to \infty} P_{ji}^{(r)}.
\]

(7.9)

The distribution $\Pi$ is said to be the limiting distribution of $P$. The classical method to determine the rate of convergence towards this limiting distribution is to consider the eigenvalues of $P$ [46].

Suppose that we have a matrix $P$ with distinct and nonzero eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Let the column vector $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})^T$ satisfy

\[
P x^{(i)} = \lambda_i x^{(i)},
\]

(7.10)

and the row vector $y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})$ satisfy

\[
y^{(i)} P = \lambda_i y^{(i)},
\]

(7.11)

---

$p_{ji}$ is the element in row $j$ and column $i$ in the matrix $P$. 

105
for \( i = 1, 2, \ldots, n \). Then

\[
P^r = \sum_{i=1}^{n} \lambda_i^r B_{(i)}, \quad B_{(i) j,k} = \frac{x_j^{(i)} y_k^{(i)}}{x^{(i)} \cdot y^{(i)}},
\]

(7.12)

It follows that a matrix \( P \) with distinct eigenvalues has a limiting distribution if and only if the largest eigenvalue is 1, and the remaining eigenvalues are less than one in modulus. We order the eigenvalues with \( 1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \).

Powers of a transition matrix can be written as

\[
P^r = \sum_{i=1}^{n} \lambda_i^r B_{(i)} = P^* + \sum_{i=2}^{n} \lambda_i^r B_{(i)},
\]

(7.13)

where \( P^* = [\Pi, \Pi, \cdots, \Pi]^T \), \( \Pi \) is the limiting distribution of \( P \). Thus all entries of \( P^r \) converge to their limit exponentially fast as a function of \( \lambda_2 \).

### 7.3 Modelling Avalanche in S-boxes

We will use the same s-box model proposed in [61]. Let the s-boxes in the network be defined by a random bijective mapping \( S : X \rightarrow Y \). Assume that any set of one or more input bit changes to an s-box results in a number of output bit changes represented by the random variable \( D \), i.e., \( D = wt(\Delta Y) \) where \( \Delta Y \) is the output change vector of the s-box. We assume that the likelihood of a particular nonzero value for \( D \) is given by assuming that all possible values of \( \Delta Y \) belonging to the set of \( 2^n - 1 \) nonzero changes are equally likely. Hence the probability distribution of \( D \) is given by

\[
P_D(D = 0) = \begin{cases} 1, & wt(\Delta X) = 0, \\ 0, & wt(\Delta X) \geq 1, \end{cases}
\]

(7.14)

and

\[
P_D(D = d) = \begin{cases} 0, & wt(\Delta X) = 0, \\ \frac{d}{2^n-1}, & wt(\Delta X) \geq 1, \end{cases}
\]

(7.15)

for \( 1 \leq d \leq n \). Note that the above s-box model essentially represents an average over all randomly selected s-boxes and is not intended to characterize the behavior of an actual physically realizable s-box. However, as experimental evidence [61] suggests,
modelling the number of output changes of each s-box as a random variable is a suitable approximation when considering an SPN constructed using randomly selected bijective s-boxes.

Let $W_r$ represent the random variable corresponding to the number of bit changes after round $r$ given a one bit plaintext change, i.e.,

$$W_r = \sum_{s=1}^{M} \text{wt}(\Delta Y_{rs}),$$

(7.16)

where $\Delta Y_{rs}$ denotes the output change vector of the $s^{th}$ s-box in round $r$. Hence, the expected value of $W_r$ is given by

$$E(W_r) = \sum_{i=1}^{M} E(D)iP_{A_r}(A_r = i)$$

$$= \frac{2^n-1}{(2^n - 1)} \sum_{i=1}^{M} iP_{A_r}(A_r = i),$$

(7.17)

where $P_{A_r}(A_r = i)$ denotes the probability of having $i$ active s-boxes in round $r$, i.e., $i$ s-boxes with nonzero input change vectors.

Thus we have

$$P_{A_r}(A_r = i) = \sum_{j=0}^{M} P_{A_r}(A_r = i|A_{r-1} = j)P_{A_{r-1}}(A_{r-1} = j),$$

(7.18)

with the initial condition

$$p_{A_1}(A_1 = j) = \delta(j = 1),$$

(7.19)

where $\delta(a = b) = 1$ if $a = b$ and $\delta(a = b) = 0$ if $a \neq b$.

It is clear that $P_{A_r}(A_r = i|A_{r-1} = j)$ does not depend on $r$ and hence the number of active s-boxes can be modelled by a Markov chain. Now our problem is reduced to calculating the state transition matrix $P = [p_{ji}]$, where $p_{ji} = P_{A_r}(A_r = i|A_{r-1} = j)$ is the probability of having $i$ active s-box in round $r$ given that we have $j$ active s-boxes in round $r - 1$. 

107
7.4 Modelling the Linear Transformation Layer

Throughout this section, we are interested in the case \( M = n \).

7.4.1 Model A — Fixed Permutation Layer

Lemma 7.1

Assume an SPN with a fixed permutation layer \( \pi \in \Omega \), then we have

\[
p_{ji} = \frac{1}{(2^n - 1)^j} \sum_{l=n-1}^{n} (-1)^{l-n+i} \binom{l}{n-i} \binom{n}{l} (2^{n-l} - 1)^j.
\]  

(7.20)

Proof: The number of arrangements of the output bit changes so that the first \( l \) s-boxes in round \( r \) are not active given that we have \( j \) active s-boxes in round \( r - 1 \) is obtained by setting the input bit changes to these non-active s-boxes to zero and varying the other bit changes over all its possible nonzero changes. Thus the number of such arrangements is \((2^{n-l} - 1)^j\).

Using the inclusion-exclusion principle, the number of arrangements such that we have exactly \( i \) non-active s-box in round \( r \) given that we have \( j \) active s-box in round \( r - 1 \) is given by

\[
NA(i, j) = \sum_{l=i}^{n} (-1)^{l-i} \binom{l}{i} \binom{n}{l} (2^{n-l} - 1)^j.
\]  

(7.21)

The lemma follows by noting that \( p_{ji} = \frac{NA(n-i,j)}{(2^n-1)^j} \).

7.4.2 Model B — Linear Transformation Type 1

The following lemma will be used throughout this section.

Lemma 7.2

Let \( S = \bigoplus_{i=1}^{l} b_i \), \( b_i \in \mathbb{Z}_2 \) with \( P(b_i = 1) = p, 0 < p < 1 \), then the probability that \( S = 0 \) is given by

\[
P(S = 0) = \sum_{i=0}^{[l/2]} \binom{l}{2i} p^{2i}(1-p)^{l-2i} \overset{def}{=} \Phi(l,p).
\]  

(7.22)
For $p = 0$ or $p = 1$, we have

$$
\Phi(l, 0) = 1,
$$

$$
\Phi(l, 1) = \begin{cases} 
1, & l \text{ even} \\
0, & l \text{ odd}.
\end{cases} \quad \text{(7.23)}
$$

**Lemma 7.3**

Assume an SPN with a linear transformation type 1, then for $\Delta Q = 0$ (see equation (7.3), the number of arrangements for the output bit changes such that we have $i$ active s-boxes in round $r$ given that we have $j$ active s-boxes in round $r - 1$ is given by

$$
A_0(i, j) = \sum_{l=0}^{M-1} (-1)^{l-M+i} \binom{l}{M - i} \binom{M}{l} (2^n - l - 1)^j \Phi\left(j, \frac{1/2}{1 - 2^{-l-n}}\right), \quad \text{(7.24)}
$$

where $\Phi(i, j)$ is given by equations (7.22) and (7.23).

**Proof:** For every active s-box in round $r - 1$, all the input bit changes to the $i$ non-active s-boxes should be set to zero. The other output bit changes can be assigned arbitrarily such that $\Delta Q = 0$. Note that, in this case, $\Delta Q$ can be seen as the XOR of $j$ binary i.i.d. random variables, $b_l \in \mathbb{Z}_2$, $1 \leq l \leq j$, with $P(b_l = 1) = \frac{2^{n-i} - 1}{2^n - 1} = \frac{1/2}{1 - 2^{-l-n}}$. Thus the rest of the output change bits can be assigned in $(2^n - l - 1)^j \Phi\left(j, \frac{1/2}{1 - 2^{-l-n}}\right)$. Using the inclusion-exclusion principle, the number of ways to have exactly $i$ non-active s-boxes in round $r$ given that we have $j$ active s-boxes in round $r - 1$ is given by

$$
NA_0(i, j) = \sum_{l=i}^{M-1} (-1)^{l-i} \binom{l}{i} \binom{M}{l} (2^n - l - 1)^j \Phi\left(j, \frac{1/2}{1 - 2^{-l-n}}\right). \quad \text{(7.25)}
$$

The lemma follows by noting that $A_0(i, j) = NA_0(M - i, j)$. $\square$
Lemma 7.4
Assume an SPN with a linear transformation type 1, then for $\Delta Q = 1$ (see equation (7.3)), the number of arrangements for the output bit changes such that we have $i$ active s-box in round $r$ given that we have $j$ active s-box in round $r-1$ is given by

$$A_1(i, j) = \begin{cases} NA_1((M - i), M), & j = M, \\ (2^n - 1)^j \left(1 - \Phi \left(j, \frac{1/2}{1-2^{-n}}\right)\right), & j \neq M, i = M, \\ 0, & \text{otherwise.} \end{cases}$$  \hspace{1cm} (7.26)

where

$$NA_1(i, M) = \sum_{l=i}^{M-1} (-1)^{l-i} \binom{l}{i} \binom{M}{l} \Psi(l),$$  \hspace{1cm} (7.27)

and

$$\Psi(l) = \begin{cases} \left(\frac{2^{n-l}}{2}\right)^M, & l \neq 0, \\ (2^n - 1)^M \left(1 - \Phi \left(M, \frac{1/2}{1-2^{-n}}\right)\right), & l = 0, \end{cases}$$  \hspace{1cm} (7.28)

and $\Phi(i, j)$ is given by equations (7.22) and (7.23).

Proof: For $\Delta Q = 1$, we have $j = M$ for every $i \neq M$. For $i = M$, for all the $M$ s-boxes in round $r - 1$, all the output bit changes connected to the $i$ non-active s-boxes in round $r$ should be set to one. The other output bit changes can be assigned arbitrarily such that $\Delta Q = 1$. Note that, in this case, $\Delta Q$ can be seen as the XOR of $j$ binary i.i.d. random variables, $b_l \in \mathbb{Z}_2$, $1 \leq l \leq j$, with $P(b_l = 1) = \frac{1}{2}$. Thus the rest of the output change bits can be assigned in $(2^n)^j \Phi \left(j, \frac{1}{2}\right) = \left(\frac{2^{n-l}}{2}\right)^M$ ways. The total number of bit changes arrangements with $\Delta Q = 1$ is given by $(2^n - 1)^M \left(1 - \Phi \left(M, \frac{1/2}{1-2^{-n}}\right)\right)$. Using the inclusion-exclusion principle, the number of ways to have exactly $i$ non-active s-box in round $r$ given that we have $M$ active s-box in round $r-1$ is given by

$$NA_1(i, M) = \sum_{l=i}^{M-1} (-1)^{l-i} \binom{l}{i} \binom{M}{l} \Psi(l),$$  \hspace{1cm} (7.29)

where

$$\Psi(l) = \begin{cases} \left(\frac{2^{n-l}}{2}\right)^M, & l \neq 0, \\ (2^n - 1)^M \left(1 - \Phi \left(M, \frac{1/2}{1-2^{-n}}\right)\right), & l = 0. \end{cases}$$  \hspace{1cm} (7.30)
Thus the number of ways to have exactly $i$ active s-box in round $r$ given that we have $j$ active s-box in round $r - 1$ is given by

$$A_1(i, j) = \begin{cases} 
N A_1((M - i), M), & j = M. \\
(2^n - 1)^j \left(1 - \Phi\left(j, \frac{1/2}{1-2^{-n}}\right)\right), & j \neq M, i = M. \\
0, & \text{otherwise.}
\end{cases}$$

(7.31)

which proves the lemma.

Combining the results above, we have

$$p_{ji} = \frac{A_0(i, j) + A_1(i, j)}{(2^n - 1)^j}.$$  

(7.32)

### 7.4.3 Model C — Linear Transformation Type 2

**Lemma 7.5**

Let $\Psi(n, k)$ be the number of choices of $k$ nonzero elements of $\mathbb{Z}_2^n$ which sum to zero, then

$$\Psi(n, k) = (-1)^k + \sum_{r=0}^{k-1} (-1)^r \binom{k}{r} 2^{n(k-r-1)}.$$  

(7.33)

**Proof:** Let $\Upsilon(n, k, r)$ denote the number of ways to choose $k$ elements of $\mathbb{Z}_2^n$ which sum to zero such that $r$ or more of them are zero. Thus

$$\Upsilon(n, k, r) = \begin{cases} 
\binom{k}{r} 2^{n(k-r-1)}, & 0 \leq r < k \\
1, & r = k.
\end{cases}$$

(7.34)

Using the inclusion-exclusion principle, we have

$$\Psi(n, k) = \sum_{r=0}^{k} (-1)^r \Upsilon(n, k, r),$$

(7.35)

which proves the lemma.
Lemma 7.6
Assuming SPN with a linear transformation type 2, then we have

\[ p_{ji} = (\Phi_1 + \Phi_2)/(2^n - 1)^j, \]

(7.36)

where

\[ \Phi_1 = \Psi(n, j) \delta(i = j), \]

(7.37)

and

\[
\Phi_2 = \Psi(n - 1, i + j - M) \binom{j}{M - i} (2^n - 1) \\
\times \left( 2^{i+j-M-1} + \frac{1}{2} (-1)^{j-i} \delta(i + j = M) \right).
\]

(7.38)

Proof: Since all choices for which \( j \) s-boxes in round \( r - 1 \) are active are equally likely, we may assume the first \( j \) s-boxes are active. Thus the last \((M - j)\) s-boxes in round \( r \) will have the same input change vector. \((\delta(i = j) - \delta(i = M))\Psi(n, j)\) counts the number of ways this could give rise to the last \((M - j)\) s-boxes in round \( r \) being non-active and makes sure that they are not counted towards the case where all the s-boxes in round \( r \) are active. If the last \((M - j)\) s-boxes in round \( r \) are active then our problem is basically reduced to considering the first \( j \) s-boxes in round \( r \), and we need to compute \( p(i + j - M|j) \) for \( M = j \). In this case we have \( \binom{M}{M-i} \) ways to choose which \((M - i)\) of the remaining \( j \) s-boxes in round \( r \) should be non-active. The input change vectors to these s-boxes must be the same and we have \((2^n - 1)\) ways to choose which value they have. Call this value \( w \). Since the solution does not depend on the specific value of \( w \neq 0 \), we assume that \( w = (100 \cdots 0) \). The remaining \( i + j - M \) input changes can not be \( 0 \) (by definition they are nonzero changes) or \( w \) (because this will contradict with the assumption that we have \( i \) active s-box). So these input changes must have the form \((ab)\), \( a \in Z_2, b \in Z_2^{n-1} \setminus \{0\} \). Moreover these nonzero \( b \)'s must add to \( 0 \) and the \( a \)'s must add to \( 1 \) over \( Z_2 \). If \( i + j \neq M \) then we have

\footnote{Two different proofs for this Lemma were provided by \([132]\) and \([51]\).}
\[ \Psi(n - 1, i + j - M) \text{ choices for the } b\text{'s and } 2^{i+j-M-1} \text{ choices for the } a\text{'s. If } i + j = M \text{ (this is only correct if } j \text{ is odd) then the sum of the } a\text{'s will always equal to 1. Thus we have } 2^{i+j-M-1} + \frac{1}{2}(-1)^{i-1}\delta(i + j = M) \text{ total choices for the } a\text{'s. Similarly, one can show that the number of arrangements such that all the s-boxes in round } r \text{ are active is given by } 2^{j-1}(2^n - 1)\Psi(n - 1, j) + \Psi(n, j)\delta(j = M). \] The conditional probability is obtained by dividing by the total number of nonzero input vector changes, \((2^n - 1)^j\). \(\square\)

7.4.4 Model D — Linear Transformation Type 3

Lemma 7.7 [78]:

The number of code words of weight \(w\) in an \([n, k, d = n - k + 1]\) MDS code over \(GF(q)\) is given by

\[ \Phi(n, d, w) = \binom{n}{w}(q-1)\sum_{l=0}^{w-d}(-1)^l\binom{w-1}{l}q^{w-d-l}. \quad (7.39) \]

Lemma 7.8

Assuming an SPN with a linear transformation type 3, then we have

\[ p_{ji} = \Psi(M - j, i + j)/(2^n - 1)^j\binom{M}{j}, \quad (7.40) \]

where

\[ \Psi(i, w) = \sum_{l=i}^{M}(-1)^{l-i}\binom{l}{i}\binom{M}{l}\Phi(2M - l, M + 1, w), \quad (7.41) \]

and \(\Phi(n, d, w)\) is given by equation (7.39).

Proof: If the number of non-active s-boxes in round \(r - 1\) is greater than or equal to \(i\), then our problem corresponds to \([2M - i, M - i, M + 1]\) MDS code. Let \(A_r\) denote the number of active s-boxes in round \(r\), then the number of arrangements for which we have \(A_{r-1} + A_r = w\) and the number of non-active s-boxes in round \(r - 1\) is greater
than or equal to \( i \) is given by
\[
\Phi(2M - i, M + 1, w) \binom{M}{i}.
\] (7.42)

Using the Inclusion-Exclusion Principle, the number of choices for which \( A_{r-1} + A_r = w \) and the number of non-active s-boxes in round \( r - 1 \) is exactly equal to \( i \) is given by
\[
\Psi(i, w) = \sum_{l=i}^{M} (-1)^{l-i} \binom{l}{i} \binom{M}{l} \Phi(2M - l, M + 1, w).
\] (7.43)

Thus we have
\[
p_{ji} = \Psi(M - j, i + j)/(2^n - 1)^l \binom{M}{j},
\] (7.44)
which proves the lemma.

### 7.5 Results and Discussion

For \( N = 64 \) and \( M = n = 8 \) (a practical size of SPN) the transition matrices for SPNs based on the four models are shown in Figure 7.3 and Figure 7.4. It is easy to check that these matrices are ergodic. The limiting distribution and the eigenvalues of these matrices is shown in Figure 7.5.

Let
\[
\epsilon = \lim_{r \to \infty} |E(W_r) - N/2|,
\] (7.45)
i.e., \( \epsilon \) denotes the deviation of the limiting avalanche characteristics from the ideal characteristics. By numerical substitution with \( n = M = 8 \), and the limiting distributions in Figure 7.5, we get
\[
\epsilon = \left| \frac{N}{2} - \frac{n/2}{1 - 2^{-n}} \sum_{i=1}^{M} i \Pi(i) \right| < 2^{-41}
\] (7.46)
for the four models above. If the SPN is constructed using one randomly selected \( 64 \times 64 \) s-box then, using equation (7.15), we have
\[
\epsilon = \left| 32 - \sum_{d=0}^{64} d \frac{\binom{64}{d}}{264^2 - 1} \right| = \left| 32 - \frac{264}{264^2 - 1} \sum_{d=0}^{64} d \frac{\binom{64}{d}}{264} \right|
\] (7.47)
\[
= \left| 32 - 32 \frac{264}{264^2 - 1} \right| \approx 2^{-59}.
\]
Table 7.1 shows the value of the second largest eigenvalue for the four models.

<table>
<thead>
<tr>
<th></th>
<th>Model A</th>
<th>Model B</th>
<th>Model C</th>
<th>Model D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\lambda_2</td>
<td>|$</td>
<td>$3.185 \times 10^{-2}$</td>
<td>$4.439 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 7.1 The Second Largest Eigenvalues for SPNs with $N = 64$, and $M = n = 8$.

From Table 7.1, it is clear that the transition matrices for SPNs with a permutation layer have the slowest convergence (largest second eigenvalue), and the transition matrices for SPNs with an MDS linear transformation layer have the fastest convergence (smallest second eigenvalue). It is also clear that the linear transformation is effective in improving the avalanche characteristics of SPNs. This is also illustrated by noting that the transition matrices of SPNs with linear transformations tend to have more zeroes in the places corresponding to transitions from small number of active s-boxes to small number of active s-boxes. This implies that for small number of rounds, on average, SPNs with linear transformations tend to have more active s-boxes than SPNs with permutation layers. Figure 7.2 shows how the experimental results agree with our theoretical model for SPNs with $N = 64$, $n = M = 8$.

Figure 7.2 Theoretical and Experimental Avalanche for SPN with $N = 64$, $n = M = 8$.  

115
One should note that the limiting distribution is the same for all the models above. This is a nice illustration of the product cipher design philosophy: a strong block cipher can be obtained by iterating a weaker one for a sufficient number of rounds.

While the s-box model used throughout this chapter is suitable for studying the avalanche characteristics of SPNs, it can not be used directly to determine the resistance of the network to differential cryptanalysis, where the analysis should be performed on a specified set of s-boxes. However, from the transition matrix, we can calculate the minimum number of s-boxes involved in any 2 rounds of a differential characteristic. This number will be greater than or equal to \( \min_{i,j,P_{ij}=0} (i + j) \).

This model also helps visualizing the average dynamic performance of SPNs with randomly selected s-boxes and different linear transformation layers.

In summary, we have presented analytical models for the avalanche characteristics of four general classes of substitution-permutation encryption networks. The results show that using an appropriate diffusive linear transformation between rounds can improve the avalanche characteristics of the network. This facilitates the construction of efficient ciphers with fewer rounds.
\[
P_A = \begin{bmatrix}
3.14 \times 10^{-2} & 1.09 \times 10^{-1} & 2.19 \times 10^{-1} & 2.75 \times 10^{-1} & 2.19 \times 10^{-1} & 1.09 \times 10^{-1} & 3.14 \times 10^{-2} & 3.92 \times 10^{-3} \\
1.23 \times 10^{-4} & 3.01 \times 10^{-3} & 2.15 \times 10^{-2} & 8.50 \times 10^{-2} & 2.08 \times 10^{-1} & 3.13 \times 10^{-1} & 2.69 \times 10^{-1} & 1.01 \times 10^{-1} \\
4.82 \times 10^{-7} & 4.22 \times 10^{-5} & 8.95 \times 10^{-4} & 9.12 \times 10^{-3} & 5.43 \times 10^{-2} & 1.95 \times 10^{-1} & 3.94 \times 10^{-1} & 3.46 \times 10^{-1} \\
1.89 \times 10^{-9} & 5.23 \times 10^{-7} & 2.86 \times 10^{-5} & 6.87 \times 10^{-4} & 9.19 \times 10^{-3} & 7.23 \times 10^{-2} & 3.17 \times 10^{-1} & 6.01 \times 10^{-1} \\
7.42 \times 10^{-12} & 6.26 \times 10^{-9} & 8.53 \times 10^{-7} & 4.50 \times 10^{-5} & 1.29 \times 10^{-3} & 2.16 \times 10^{-2} & 1.98 \times 10^{-1} & 7.79 \times 10^{-1} \\
2.91 \times 10^{-14} & 7.40 \times 10^{-11} & 2.35 \times 10^{-8} & 2.78 \times 10^{-6} & 1.69 \times 10^{-4} & 5.84 \times 10^{-3} & 1.09 \times 10^{-1} & 8.84 \times 10^{-1} \\
1.14 \times 10^{-16} & 8.73 \times 10^{-13} & 6.53 \times 10^{-10} & 1.67 \times 10^{-7} & 2.13 \times 10^{-5} & 1.51 \times 10^{-3} & 5.77 \times 10^{-2} & 9.41 \times 10^{-1} \\
4.47 \times 10^{-19} & 1.03 \times 10^{-14} & 1.79 \times 10^{-11} & 9.94 \times 10^{-9} & 2.63 \times 10^{-6} & 3.81 \times 10^{-4} & 2.95 \times 10^{-2} & 9.70 \times 10^{-1}
\end{bmatrix}
\]

\[
P_B = \begin{bmatrix}
0.0 & 1.09 \times 10^{-1} & 0.0 & 2.75 \times 10^{-1} & 0.0 & 1.09 \times 10^{-1} & 0.0 & 5.06 \times 10^{-1} \\
1.23 \times 10^{-4} & 1.29 \times 10^{-3} & 1.12 \times 10^{-2} & 4.19 \times 10^{-2} & 1.04 \times 10^{-1} & 1.56 \times 10^{-1} & 1.34 \times 10^{-1} & 5.50 \times 10^{-1} \\
0.0 & 2.19 \times 10^{-5} & 4.46 \times 10^{-4} & 4.56 \times 10^{-3} & 2.72 \times 10^{-2} & 9.75 \times 10^{-2} & 1.97 \times 10^{-1} & 6.73 \times 10^{-1} \\
1.89 \times 10^{-9} & 2.58 \times 10^{-7} & 1.43 \times 10^{-5} & 3.44 \times 10^{-4} & 4.59 \times 10^{-3} & 3.62 \times 10^{-2} & 1.59 \times 10^{-1} & 8.00 \times 10^{-1} \\
0.0 & 3.14 \times 10^{-9} & 4.18 \times 10^{-7} & 2.25 \times 10^{-5} & 6.49 \times 10^{-4} & 1.08 \times 10^{-2} & 9.89 \times 10^{-2} & 8.89 \times 10^{-1} \\
2.91 \times 10^{-14} & 3.69 \times 10^{-11} & 1.18 \times 10^{-8} & 1.39 \times 10^{-6} & 8.47 \times 10^{-5} & 2.92 \times 10^{-3} & 5.49 \times 10^{-2} & 9.42 \times 10^{-1} \\
0.0 & 4.37 \times 10^{-13} & 3.26 \times 10^{-10} & 8.37 \times 10^{-8} & 1.06 \times 10^{-5} & 7.54 \times 10^{-4} & 2.89 \times 10^{-2} & 9.70 \times 10^{-1} \\
5.77 \times 10^{-17} & 5.61 \times 10^{-14} & 3.49 \times 10^{-11} & 1.33 \times 10^{-8} & 3.00 \times 10^{-6} & 4.06 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1}
\end{bmatrix}
\]

Figure 7.3 The Transition Matrices for Model A and B ($M = n = 8$)
\[ P_C = \begin{bmatrix}
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\
0.0 & 3.92 \times 10^{-3} & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 9.96 \times 10^{-1} \\
0.0 & 0.0 & 3.91 \times 10^{-3} & 0.0 & 1.54 \times 10^{-5} & 0.0 & 1.17 \times 10^{-2} & 9.84 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 3.91 \times 10^{-3} & 0.0 & 9.19 \times 10^{-5} & 1.54 \times 10^{-2} & 9.81 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 2.37 \times 10^{-10} & 0.0 & 3.91 \times 10^{-3} & 1.51 \times 10^{-4} & 1.92 \times 10^{-2} \\
0.0 & 0.0 & 0.0 & 3.53 \times 10^{-9} & 1.19 \times 10^{-6} & 4.13 \times 10^{-3} & 2.29 \times 10^{-2} & 9.77 \times 10^{-1} \\
3.64 \times 10^{-15} & 0.0 & 1.94 \times 10^{-11} & 8.15 \times 10^{-9} & 2.07 \times 10^{-6} & 3.15 \times 10^{-4} & 3.06 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 1.01 \times 10^{-13} & 5.11 \times 10^{-11} & 1.62 \times 10^{-8} & 3.29 \times 10^{-6} & 4.19 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1}
\end{bmatrix} \]

\[ P_D = \begin{bmatrix}
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 3.14 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 4.31 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 3.38 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 0.0 & 1.66 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 0.0 & 0.0 & 5.19 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \\
0.0 & 1.02 \times 10^{-13} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \\
1.14 \times 10^{-16} & 9.86 \times 10^{-14} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1}
\end{bmatrix} \]

Figure 7.4 The Transition Matrices for Model C and D ($M = n = 8$)
Model A:

$\Pi_A = \begin{bmatrix} 1.11 \times 10^{-16} & 9.87 \times 10^{-14} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \end{bmatrix}$.

$\lambda_A = \{1.0, 3.18 \times 10^{-2}, 3.18 \times 10^{-2}, 3.19 \times 10^{-3}, 2.81 \times 10^{-3}, 9.19 \times 10^{-4}, 2.77 \times 10^{-4}, 3.94 \times 10^{-5}\}$.

Model B:

$\Pi_B = \begin{bmatrix} 1.11 \times 10^{-16} & 9.87 \times 10^{-14} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \end{bmatrix}$.

$\lambda_B = \{1.0, 4.44 \times 10^{-3}, 4.39 \times 10^{-3}, 3.12 \times 10^{-3}, -3.11 \times 10^{-3}, 7.39 \times 10^{-4}, 2.79 \times 10^{-4}, 3.76 \times 10^{-5}\}$.

Model C:

$\Pi_C = \begin{bmatrix} 1.11 \times 10^{-16} & 9.87 \times 10^{-14} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.1733 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \end{bmatrix}$.

$\lambda_C = \{1.0, 3.92 \times 10^{-3}, 3.92 \times 10^{-3}, 3.91 \times 10^{-3}, 3.91 \times 10^{-3}, 3.91 \times 10^{-3}, 5.77 \times 10^{-8}, 5.77 \times 10^{-8}\}$.

Model D:

$\Pi_D = \begin{bmatrix} 1.11 \times 10^{-16} & 9.87 \times 10^{-14} & 5.03 \times 10^{-11} & 1.60 \times 10^{-8} & 3.27 \times 10^{-6} & 4.17 \times 10^{-4} & 3.04 \times 10^{-2} & 9.69 \times 10^{-1} \end{bmatrix}$

$\lambda_D = \{1.0, -2.08 \times 10^{-8}, 1.39 \times 10^{-8}, -1.12 \times 10^{-9}, -6.39 \times 10^{-9}, -6.35 \times 10^{-9}, 1.85 \times 10^{-9}, -1.85 \times 10^{-9}\}$.

Figure 7.5 The Limiting Distribution and the Eigenvalues for Model A, B, C and D ($M = n = 8$)
Chapter 8 Conclusion

Although some researchers feel that the design of secure and efficient block ciphers has a solid theoretical foundation, we feel that the work is not yet complete. In this thesis, we have provided many results that enhance our understanding of how to achieve this objective.

8.1 Contributions of the Thesis

Our work has contributed to cryptography and cryptanalysis in several ways. In particular we have

- derived expressions for the expected size of the maximum XOR table entry and the maximum Linear Approximation Table entry for some combinatorial structures of interest such as regular mappings, and injective mappings.
- developed an expression to estimate the size of the largest entry in the LAT of a randomly selected injective s-box. In particular we showed that the nonlinearity of a randomly selected $8 \times 32$ s-box (the size of CAST s-boxes) is about 72.
- presented a new construction method for highly nonlinear injective s-boxes and showed how the resistance of CAST-like encryption algorithms (based on randomly selected substitution boxes) to the basic linear cryptanalysis was underestimated.
- extended the definitions of many cryptographic criteria to multi-output boolean functions.
- showed the relationship between the Walsh-Hadamard transform and various types of information leakage.
- introduced a new class of Substitution Permutation Networks (SPNs) with the advantage that the same network can be used to perform both the encryption and the decryption operations, and examined different cryptographic properties of this class such as resistance to both linear and differential cryptanalysis.
- showed how to construct MDS linear transformations with the involution property.
• developed an analytical model of the avalanche criterion for SPNs with different linear transformation layers.

• showed that the private key cryptosystem proposed at Crypto’ 90 by Koyama and Terada is affine over $GF(2)$ and hence it is completely insecure (see appendix A).

• proved a conjecture by Cusick regarding the number of functions satisfying the Strict Avalanche Criterion (see appendix C).

Most of the contributions above were published in [154], [145], [144], [146], [149], [153], [150], [142], [147], [148], [141], [152], [151] and [143].

8.2 Future Work

As we mentioned before, we feel that design and analysis of efficient and secure block ciphers is far from being a complete task. In this section we present several directions for future research

• Investigation of the non s-box based approach in the design of block ciphers. This approach (e.g., TEA and RC5) is attractive since it has the minimum memory requirements which makes it suitable for applications with limited memory such as smart cards and wireless applications. Although IDEA can be considered as belonging to this class of block ciphers, we feel that the data dependent rotation used in RC5, because of its simplicity compared to complex operations such as modular multiplication, is a more elegant operation. The security of this simple operation should be explored further and more operations might be investigated.

• Finding s-boxes with high diffusion order. The first step is to find a theoretical bound on the diffusion order. Algebraic constructions or clever search techniques, such as Genetic Algorithms or simulated annealing, can be used.

• Extending the results in this thesis to Higher order differentials and truncated differentials.
• Examining the pseudo-randomness properties of SPNs is still an untouched area. A complexity theoretic approach similar to Luby-Rackoff results using DES-like ciphers can be investigated.

• Examining possible modifications of the CAST s-boxes such that the resulting round function is bijective. This will make the CAST algorithm immune to the attack described by Rijmen et al. Although we are aware of many techniques to achieve this, all of them result in some degradation of the nonlinearity and the differential uniformity of the round function.

• Investigation of efficient algorithms to calculate the nonlinearity of the resulting CAST 32 × 32 s-box when the injective s-boxes are combined by modular addition or modular subtraction.

• Deriving upper bounds for the differential probabilities and linear correlations for the different SPNs studied in this thesis.
Appendix A Cryptanalysis of the "Nonlinear-Parity Circuits" Proposed at Crypto '90

At Crypto '90, K. Koyama and R. Terada from the Japanese NTT Research Laboratory proposed a family of cryptographic functions for application to symmetric block ciphers [71]. Here, we show that this family of circuits is affine over $GF(2)$. More explicitly, for any specific key (key), the ciphertext $y$ is related to the plaintext $x$ by the simple affine relation $y = M_{key}x \oplus d_{key}$ where $M_{key}$ is an $n \times n$ non singular binary matrix and $d_{key}$ is an $n \times 1$ binary vector where $n$ is the block length of the cipher. This renders this family of ciphers completely insecure as it can be broken with only $n + 1$ linearly independent plaintext blocks and their corresponding ciphertext blocks.

Definitions and Cipher Description:
Koyama and Terada described their proposed cipher as follows:

**Definition 1:** A parity circuit layer of length $n$, or simply an $L(n)$ circuit layer, is a boolean device with an $n$-bit input and an $n$-bit output, characterized by a key that is a sequence of $n$ symbols from $\{0, 1, -, +\}$.

**Definition 2:** Function $b = f(k, a)$, as computed by an $L(n)$ circuit layer with key $k = k_1k_2\cdots k_n \in \{0, 1, -, +\}^n$ is the relation from an $n$-bit input sequence $a = a_1a_2\cdots a_n \in \{0, 1\}^n$ to an $n$-bit output sequence $b = b_1b_2\cdots b_n \in \{0, 1\}^n$ defined below. An $L(n)$ circuit layer first computes a variable $T \in \{0, 1\}$ such that:

$$T = \bigoplus_{j=1}^{n} t_j,$$  \hspace{1cm} (A.1)

where

$$t_j = \begin{cases} 
1, & \text{if} (k_j = 0 \text{ and } a_j = 0) \text{ or } (k_j = 1 \text{ and } a_j = 1) \\
0, & \text{otherwise.} 
\end{cases} \hspace{1cm} (A.2)$$
Output \( b = b_1 b_2 \cdots b_n \) of the circuit layer is then

\[
b_j = \begin{cases} 
\overline{a_j} & \text{if } k_j = - \text{ and } T = 1 \\
a_j & \text{if } k_j = + \text{ and } T = 0 \\
\text{otherwise.} & k_j = 1
\end{cases}
\] (A.3)

The proposed cipher is obtained by composing the parity circuit layers as follows:

**Definition 3:** A parity circuit of width \( n \) and depth \( d \), or simply a \( C(n, d) \) circuit, is a matrix of \( d \) \( L(n) \) circuit layers with keys denoted by \( \text{key} = k_1 || k_2 || \cdots || k_d \) for which the \( n \) output bits of the \( (i-1) \)-th circuit layer are the \( n \) input bits for the \( i \)-th circuit layer for \( 2 \leq i \leq d \). The key for the \( C(n, d) \) circuit is a \( d \times n \) matrix whose \( d \) lines contain the circuit layer keys.

An example of \( C(n, d) \) with \( n = 10 \), and \( d = 3 \) is shown in Table A.1.

<table>
<thead>
<tr>
<th>Input</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>K2</td>
<td>+</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>K3</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Output</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.1. A parity circuit \( C(n, d) \) with \( n = 10 \), and \( d = 3 \)

124
Main Result:

The lemma below follows from the fact that the set of affine functions is closed under the XOR operation.

Lemma A.1

Any combinational boolean function that can be realized using only XOR gates is an affine function over $GF(2)$.

Koyama and Terada presented many cryptographic properties of these $C(n,d)$ circuits. They also claimed, without proof, that the order of the boolean canonical form of $C(n,d)$, defined to be the maximum of the order of its product terms, increases exponentially as $n$ or $d$ increases, and hence it is practically infeasible to cryptanalyze $C(n,d)$ if $n \geq 64$ and $d \geq 8$. The following theorem shows that this is not true.

Theorem A.1

The output ciphertext, $y$, of the parity check circuit $C(n,d)$ is related to the plaintext input, $x$, by the following affine relation:

$$y = M_{key}x \oplus d_{key}, \quad (A.4)$$

where $M_{key}$ is an $n \times n$ non-singular binary matrix and $d_{key}$ is an $n \times 1$ binary vector.

Proof: Equations (A.2) and (A.3) can be expressed as

$$t_j = \begin{cases} a_j \oplus k_j \oplus 1, & k_j = 0, \\ a_j \oplus 1, & k_j = 1, \\ 0, & k_j = +, \\ 0, & k_j = - \end{cases} \quad (A.5)$$

$$b_j = \begin{cases} a_j, & k_j = 0, \\ a_j \oplus T \oplus 1, & k_j = +, \\ a_j \oplus T, & k_j = - \end{cases} \quad (A.6)$$
Hence for any fixed key, \( k = k_1 || k_2 || \cdots || k_d \) equations (1), (2), and (3), and consequently the parity circuit, \( C(n, d) \), can be realized using only XOR gates. Hence every output component of the parity circuit is an affine function of the input variables over \( GF(2) \). Since \( C(n, d) \) is a bijective mapping, then the matrix \( M_{key} \) is non-singular.

**Example:** For the cipher described by Table A.1, the ciphertext, \( y \), in relation to the plaintext input, \( x \), is given by the following equation:

\[
y = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix} \oplus \begin{bmatrix}
1 \\
1 \\
0 \\
1 \\
0 \\
0 \\
1 \\
1 \\
0 \\
0 \\
1
\end{bmatrix}. \quad (A.7)
\]

By noting that affine functions cannot satisfy the Strict Avalanche Criterion (SAC) then the \( C(n, d) \) circuits can never satisfy the SAC as was falsely claimed.

**Conclusion**

The private key cryptosystem proposed at Crypto' 90 by Koyama and Terada is affine over \( GF(2) \) and hence it is completely insecure.
Appendix B Cryptanalysis of “key agreement scheme based on generalized inverses of matrices”

Dawson and Wu [36] proposed a new key agreement scheme based on generalized inverses of matrices [10]. They suggested that for a key length of $kn$, using binary matrices of size $k \times m$, $k < m$, and $m \times n$, the security parameter is at least $2^{(m-k)n}$. Here, we show that this scheme is insecure. In particular, we show that breaking this scheme is equivalent to solving a set of $m \times n$ consistent linear equations with $m^2$ binary variables.

**Definition:** Any matrix $A$ has a generalized inverse $A^-$ if and only if

$$AA^-A = A.$$  \hspace{1cm} (B.1)

$A^-$ is given by

$$A^- = C^{-1}\begin{bmatrix} I_r & U \\ V & W \end{bmatrix} R^{-1},$$  \hspace{1cm} (B.2)

where $r$ is the rank of $A$, $I_r$ is the identity matrix of order $r$, $R$ and $C$ are such that

$$A = R\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} C,$$  \hspace{1cm} (B.3)

$U, V$ and $W$ are arbitrary matrices of the proper order. A generalized inverse which satisfies $AA^-A = A$ and $A^-AA^- = A^-$ is called a reflexive generalized inverse of $A$. In this case, we must have $W = VU$. Note that a generalized inverse defined by equation (B.1) always exists. Our discussion will be on the binary field $F_2 = \{0, 1\}$.

The key agreement scheme proposed in [36] is described by the authors as follows: Assume that a user, Alice, wants to establish a common secret key with Bob via a public channel. They can follow the steps below:
(i) Alice randomly chooses a binary $k \times m$ matrix $A$ and an arbitrary generalized inverse $A^{-}$ of $A$.

(ii) Alice sends Bob $A^{-}A$. Alice keeps $A$ and $A^{-}$ secret.

(iii) Bob randomly chooses a binary $m \times n$ matrix $B$ and an arbitrary generalized inverse $B^{-}$ of $B$.

(iv) Bob sends Alice the matrices $A^{-}AB$ and $A^{-}ABB^{-}$. Bob keeps $B$ and $B^{-}$ secret.

(v) Alice sends Bob the matrix $AA^{-}ABB^{-} = ABB^{-}$.

(vi) Alice and Bob are then able to formulate the key $K = AB$ which is a $k \times n$ matrix by computing $AA^{-}AB = AB$ and $ABB^{-}B = AB$, respectively.

Pinch [103] proposed two attacks on the above scheme. The first attack reduces the security parameter to $2^{k^2}$ and the second attack, which applies to about 28% of the cases, reduces the security parameter to $2^{(m-k)m}m^3$. Here we show that breaking this scheme (i.e., obtaining $AB$ from the public information) is equivalent to solving a set of consistent $m \times n$ linear equations with $m^2$ binary variables.

The following Lemma will be used in the attack.

**Lemma B.1**

There exists at least one matrix $Y$ that satisfies the equation

$$XBB^{-}YXB = XB. \quad (B.4)$$

**Proof:** Take $Y = B(XB)^{-}$ then

$$XBB^{-}YXB = X(BB^{-}B)(XB)^{-}XB$$

$$= (XB)(XB)^{-}(XB) = XB. \quad (B.5)$$

The security argument given in [36] was based on the difficulty of solving for $A$ and $B$ given the public information $A^{-}A, A^{-}AB, A^{-}ABB^{-}$ and $ABB^{-}$. Here we show that we can determine the product $AB$ without solving separately for $A$ or $B$. 
The Proposed Attack:

Let \( X = A^\top A \) then solve the linear equation

\[
(XBB^-)Y(XB) = (XB)
\]  
(B.6)

for the \( m \times m \) matrix \( Y \). From the above lemma, this equation has at least one valid solution. Note that we need to solve this equation ( i.e., we can not directly use the solution given in the lemma above because \( B \) is not known.)

Multiplying both sides of equation (B.6) by \( A \) (and by noting that \( AX = AA^\top A = A \) we get

\[
(ABB^-)Y(XB) = AB.
\]  
(B.7)

Both \( ABB^- \) and \( XB = A^\top AB \) are sent over the public channel. Hence we can determine the secret key.

For \((k, m, n) = (7, 12, 15)\), the security parameter suggested in [36] is \( 2^{75} \). The attack described in this letter gets the key by solving a set of consistent 180 linear equations with 144 binary variables.

Example:

Take \((k, m, n) = (3, 4, 5)\). Let

\[
A = \begin{pmatrix}
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}, \quad
A^- = \begin{pmatrix}
1 & 0 & 1 \\
1 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{pmatrix}.
\]  
(B.8)

and

\[
B = \begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}, \quad
B^- = \begin{pmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}.
\]  
(B.9)
Then we have

$$XB = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad XBB^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad ABB^{-1} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (B.10)$$

Solving equation (B.6) for $Y$ we get 4096 distinct valid solutions. Here we give just two of them

$$Y = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{or} \quad Y = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (B.11)$$

It can be verified that

$$(ABB^{-1})Y(XB) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} = AB. \quad (B.12)$$

Conclusion

The key agreement scheme proposed by Dawson and Wu is insecure.
Appendix C  Bounds on the Number of Functions Satisfying the Strict Avalanche Criterion

The Strict Avalanche Criterion (SAC) was introduced by Webster and Tavares [136] in a study of design criteria for certain cryptographic functions. As in the case with any criterion of cryptographic significance, it is of interest to count the functions which satisfy the criterion. Many recent papers (for example [76], [32]) have been concerned with counting functions that satisfy the SAC of various orders. It is easier to count the functions satisfying the SAC of the largest order, because relatively few functions exist which satisfy these stringent criteria.

O'Connor [95] gave an upper bound for the number of functions satisfying the SAC. In [31] Cusick gave a lower bound for the number of functions satisfying the SAC. He also gave a conjecture that provided an improvement of the lower bound. In this section, we give a constructive proof for this conjecture. Moreover, we provide an improved lower bound.

Notation

Throughout this section, let

\[ f_n : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2 \] describes a boolean function with \( n \) input variables.

\[ \mathcal{V} = \{ \mathbf{v}_i \mid 0 \leq i \leq 2^n - 1 \} \] denotes the set of vectors in \( \mathbb{Z}_2^n \) in lexicographical order.

A boolean function \( f_n(x) \) is specified by \( f_n(x) = [b_0, b_1, ..., b_{2^n-1}] \), where \( b_i = f_n(v_i) \).

\( \mathbf{e} \): denotes any element of \( \mathbb{Z}_2^n \) with Hamming weight 1. Let \( \mathbf{e}, \mathbf{v}_i \in \mathbb{Z}_2^{n-1} \) denote the \( n-1 \) least significant bits of \( \mathbf{e} \) and \( v_i \) respectively.

\( \mathbf{a} \): denotes any element of \( \mathbb{Z}_2^{n-1} \) with odd Hamming weight.

\[ g_{n-1} : \mathbb{Z}_2^{n-1} \rightarrow \mathbb{Z}_2 \] denotes the boolean function \( 1 \cdot x \oplus b, b \in \mathbb{Z}_2 \) where \( 1 \) denotes the all ones vector in \( \mathbb{Z}_2^{n-1} \), \( \cdot \) denotes the dot product operation over \( \mathbb{Z}_2 \) and \( \oplus \) denotes
the XOR operation. It is easy to see that \( g_{n-1} \) satisfies

\[
g_{n-1}(x) = g_{n-1}(x \oplus a) \oplus 1. \tag{C.1}
\]

\( MSB(\cdot) \) denotes the most significant bit of the enclosed argument.

\( SAC^n \) denotes the number of functions with \( n \) input bits that satisfy the SAC.

**Cusick’s Conjecture**

The following conjecture is given in [31]. This conjecture implies that there are at least \( 2^{2^n-1} \) boolean functions of \( n \) variables which satisfy the SAC.

**Conjecture** [31]: Given any choice of the values \( f_n(v_i), \; 0 \leq i \leq 2^{n-1} - 1 \), there exists a choice of \( f_n(v_i), \; 2^{n-1} \leq i \leq 2^n - 1 \), such that the resulting function \( f_n(x) \) satisfies the SAC.

We prove this conjecture below. After completing our proof, we learned that Cusick and Stănică [33] independently proved the conjecture. Also, D. Biss [19] has proved a much stronger result by a much more complicated argument. If we let \( L_n = \frac{\log_2 SAC^n}{2^n} \). \( L = \lim_{n \to \infty} L_n \), then Biss proved that \( L = 1 \). The conjecture, of course, only says that \( L_n \geq \frac{1}{2} \).

For \( n = 1 \), it is trivial to show that if \( f_1(1) = f_1(0) \oplus 1 \) then the resulting function satisfies the SAC. In the following lemma we prove that, for \( n \geq 2 \), there exist at least two choices for \( f_n(v_i), \; 2^{n-1} \leq i \leq 2^n - 1 \), such that the resulting function satisfies the SAC.

**Lemma C.1**

Let \( f_n = [h_{n-1} \; [h_{n-1} \oplus g_{n-1}]] \) where \( h_{n-1} \) is an arbitrary boolean function with \( n - 1 \) input variables, \( n \geq 2 \), and \( g_{n-1} \) is constructed as above to satisfy equation (C.1), then \( f_n \) satisfies the SAC.
Proof: Case 1: $MSB(e) = 0$:

\[
\sum_{i=0}^{2^{n-1}-1} f_n(v_i) \oplus f_n(v_i \oplus e) = \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) + \sum_{i=0}^{2^{n-1}-1} f_n(v_i) \oplus f_n(v_i \oplus e) \]

\[
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) + \sum_{i=0}^{2^{n-1}-1} f_n(v_i) \oplus f_n(v_i \oplus e) + \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) \]

\[
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) + \sum_{i=0}^{2^{n-1}-1} (h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e})) = 2^{n-1}.
\]

Case 2: $MSB(e) = 1$:

\[
\sum_{i=0}^{2^{n-1}-1} f_n(v_i) \oplus f_n(v_i \oplus e) = 2 \sum_{i=0}^{2^{n-1}-1} f_n(v_i) \oplus f_n(v_i \oplus e) \]

\[
= 2 \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) \oplus g_{n-1}(\hat{v}_i)
\]

\[
= 2 \sum_{i=0}^{2^{n-1}-1} g_{n-1}(\hat{v}_i) = 2^{n-1}.
\]

which proves the lemma.

From lemma C.1 above, and by noting that we have two choices for $g_n$, we conclude that, for $n \geq 2$, the number of function satisfying the SAC is lower bounded by $2^{2^{n-1}+1}$.

Using the following lemma, one can provide some improvement to the above bound.

**Lemma C.2**

Let $f_n = [h_{n-1} \ [l_{n-1} \oplus g_{n-1}]]$ where $h_{n-1}$ is an arbitrary boolean function with $n - 1$ input variables, $l_{n-1}(x) = h_{n-1}(x \oplus a)$, $n \geq 2$, and $g_{n-1}$ is constructed as above to satisfy equation (C.1), then $f_n$ satisfies the SAC.
Proof: Case 1: $MSB(e) = 0$:

$$
\sum_{i=0}^{2^n-1} f_n(v_i) \oplus f_n(v_i \oplus e)
= \sum_{i=0}^{2^n-1} f_n(v_i) \oplus f_n(v_i \oplus e) + \sum_{i=2^{n-1}}^{2^n-1} f_n(v_i) \oplus f_n(v_i \oplus e)
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) + \\
\sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i \oplus a) \oplus h_{n-1}(\hat{v}_i \oplus a \oplus \hat{e}) \oplus g_{n-1}(\hat{v}_i) \oplus g_{n-1}(\hat{v}_i \oplus \hat{e})
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus \hat{e}) + \sum_{i=0}^{2^{n-1}-1} (\overline{h_{n-1}(\hat{v}_i)} \oplus h_{n-1}(\hat{v}_i \oplus \hat{e})) = 2^{n-1}.

Case 2: $MSB(e) = 1$:

$$
\sum_{i=0}^{2^n-1} f_n(v_i) \oplus f_n(v_i \oplus e)
= 2 \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus a) \oplus g_{n-1}(\hat{v}_i)
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus a) \oplus g_{n-1}(\hat{v}_i) + \\
\sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i \oplus a) \oplus h_{n-1}(\hat{v}_i) \oplus g_{n-1}(\hat{v}_i \oplus a)
= \sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i) \oplus h_{n-1}(\hat{v}_i \oplus a) \oplus g_{n-1}(\hat{v}_i) + \\
\sum_{i=0}^{2^{n-1}-1} h_{n-1}(\hat{v}_i \oplus a) \oplus h_{n-1}(\hat{v}_i) \oplus g_{n-1}(\hat{v}_i) = 2^{n-1}.
$$

which proves the lemma. \qed

Note that if the function $f_{n-1}$ does not have any linear structures, then all the functions generated by $l_{n-1} \oplus g_{n-1}$ will be unique for all the $2^{n-2}$ choices of $a$. From lemma
C.1 and lemma C.2 we have \( 2^{n-1} + 2 \) distinct choices for \( f_{n-1}(v_i), \) \( 2^{n-1} \leq i \leq 2^n - 1. \) Thus we have the following corollary:

**Corollary C.1**

The number of functions satisfying the SAC is lower bounded by

\[
\left( 2^{2^{n-1}} - \mathcal{L}S^{n-1} \right) (2^{n-1} + 2) + 2\mathcal{L}S^{n-1}
\]  

where \( \mathcal{L}S^{n-1} \) is the number of functions with \( n - 1 \) input bits having any linear structure.

A complicated formula for \( \mathcal{L}S^n \) is given in [100]. It can also be shown [100] that \( \mathcal{L}S^n \) is asymptotic to \( (2^n - 1)2^{2^{n-1}+1}. \)

One should note that while this bound provides some improvement over the proved bound in [31], exhaustive search (see Table C.1) shows that the quality of this bound degrades as \( n \) increases. One can improve this bound slightly by identifying special classes of functions \( f_n(v_i), 0 \leq i \leq 2^{n-1} - 1 \) for which there is a large number of choices for \( f_n(v_i), 2^{n-1} \leq i \leq 2^n - 1 \) such that the resulting function, \( f_n, \) satisfies the SAC. For example, if the function \( h_{n-1} \) satisfies the SAC, then the function \( f_n = [h_{n-1} \oplus c \cdot x \oplus b], \) \( b \in \mathbb{Z}_2 \) also satisfies the SAC. Thus our bound is slightly improved to

\[
\left( 2^{2^{n-1}} - \mathcal{L}S^{n-1} - SAC^{n-1} \right) (2^{n-1} + 2) + 2^n SAC^{n-1} + 2\mathcal{L}S^{n-1}.
\]

We now give a lower bound on the number of balanced functions that satisfy the SAC.

**Lemma C.3**

Let \( f_n = [h_{n-1} \oplus l_{n-1}] \) where \( h_{n-1} \) is an arbitrary boolean function with \( n - 1 \) input variables that satisfies \( \sum_{wt(v_i)} \text{odd} h_{n-1}(v_i) = 2^{n-3}, l_{n-1}(x) = h(x \oplus a), n \geq 2, \) and \( g_{n-1} \) is constructed as above to satisfy equation (C.1), then \( f_n \) is a balanced function that satisfies the SAC.

**Proof:** From lemma C.3 , it follows that \( f_n \) satisfies the SAC. Here we will prove that \( f_n \) is a balanced function.
\[
\sum_{i=0}^{2^n-1} f_n(v_i) = \sum_{i=0}^{2^n-1} h_{n-1}(v_i) + \sum_{i=0}^{2^n-1} h_{n-1}(\bar{v}_i \oplus a) \oplus g_{n-1}(\bar{v}_i)
\]

\[
= \sum_{i=0}^{2^n-1} h_{n-1}(v_i) + \sum_{i=0}^{2^n-1} h_{n-1}(v_i) \oplus g_{n-1}(v_i) \oplus a
\]

\[
= \sum_{i=0}^{2^n-1} h_{n-1}(v_i) + \sum_{i=0}^{2^n-1} \overline{h_{n-1}(v_i)} \oplus 1 \cdot \overline{v}_i
\]

\[
= \sum_{\text{wt}(v_i) \text{ even}} \left( h_{n-1}(v_i) + \overline{h_{n-1}(v_i)} \right) + 2 \sum_{\text{wt}(v_i) \text{ odd}} h_{n-1}(\bar{v}_i) = 2^{n-2} + 2 \cdot 2^{n-3} = 2^{n-1}.
\]

which proves the lemma.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
n & 2 & 3 & 4 & 5 \\
\hline
\mathcal{L}S^{n-1} & 4 & 8 & 128 & 4992 \\
\hline
\text{Old Bound [31]} & 2 & 4 & 16 & 256 \\
\hline
\text{New Bound (exp. (C.6))} & 8 & 64 & 1536 & 1099776 \\
\hline
\text{New Bound (exp. (C.7))} & 8 & 64 & 1920 & 1157568 \\
\hline
\text{Exact Number} & 8 & 64 & 4128 & 27522560 \\
\hline
\end{array}
\]

\textbf{Table C.1} Exact Number of Functions Satisfying SAC versus the Derived Lower Bounds.

Similarly, one can also show that the function \( f_n = [h_{n-1} \ [h_{n-1} \oplus g_{n-1}]] \) where \( h_{n-1} \) is an arbitrary boolean function that satisfies \( \sum_{\text{wt}(v_i) \text{ even}} h_{n-1}(v_i) = 2^{n-3} \) is a balanced function that satisfies the SAC. From the lemma above, it follows that the number of balanced SAC functions is lower bounded by

\[
\left( \frac{2^{n-2}}{2^{n-3}} \right) 2^{2^{n-2}+1}. \quad (C.9)
\]
References


