

LINEAR CRYPTANALYSIS OF
SUBSTITUTION-PERMUTATION NETWORKS

by

LIAM KELIHER

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada

October 2003

Copyright © Liam Keliher, 2003

Abstract

The subject of this thesis is linear cryptanalysis of substitution-permutation networks (SPNs). We focus on the rigorous form of linear cryptanalysis, which requires the concept of linear hulls.

First, we consider SPNs in which the s-boxes are selected independently and uniformly from the set of all bijective $n \times n$ s-boxes. We derive an expression for the expected linear probability values of such an SPN, and give evidence that this expression converges to the corresponding value for the true random cipher. This adds quantitative support to the claim that the SPN structure is a good approximation to the true random cipher. We conjecture that this convergence holds for a large class of SPNs.

In addition, we derive a lower bound on the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis after a given number of rounds. For common block sizes, experimental evidence indicates that this probability rapidly approaches 1 with an increasing number of rounds.

We then consider SPNs with fixed s-boxes. We present two new algorithms for upper bounding the maximum average linear hull probability for SPNs. These algorithms, named KMT1 and KMT2, are the first completely general algorithms for this purpose—they can be applied to any SPN, and they compute an upper bound that

is a function of the number of encryption rounds being evaluated. In contrast, other approaches to this problem either require that the SPN linear transformation have a specific structure, or compute a single value independent of the number of rounds. By applying KMT1 and KMT2 to the AES, we establish the provable security of the AES against linear cryptanalysis.

As a straightforward application of our work with linear hulls, we analyze the Q cipher, an SPN submitted to the European Commission's NESSIE cryptographic competition. By using linear characteristics, not linear hulls, the designer of Q evaluates the cipher to be secure against linear cryptanalysis. However, we prove that Q can be broken using linear cryptanalysis based on linear hulls. To our knowledge, this is the first use of linear hulls to break a proposed cipher.

Acknowledgments

Hitherto hath the Lord helped us.

I Samuel 7:12

The Holy Bible

I am grateful to my co-supervisors, Henk Meijer (School of Computing) and Stafford Tavares (Electrical and Computer Engineering), for a steady source of ideas, enthusiasm, and collegiality. I feel that I could not have had a better supervisory arrangement.

I acknowledge the funding agencies that have provided financial support for the research in this thesis: the Natural Sciences and Engineering Research Council of Canada (NSERC), and Communications and Information Technology Ontario (CITO).

And I will always be thankful to my wife, Ronda, for her love and unwavering support during the years that went into this thesis (and for her sharp editorial eye).

Statement of Originality

I, Liam Keliher, certify that this Ph.D. dissertation is original, and that all the ideas attributable to others have been properly referenced.

Notation

N	block size (number of plaintext/ciphertext bits)
n	s-box input/output size
M	number of s-boxes per round
R	number of SPN rounds
K	number of key bits
$\{0, 1\}^d$	set of all binary vectors of length d
$wt(\mathbf{x})$	Hamming weight of binary vector \mathbf{x}
\oplus	exclusive OR operation (XOR)
$\mathbf{x} \bullet \mathbf{y}$	inner (dot) product of binary vectors \mathbf{x} and \mathbf{y}
$\mathbf{0}$	all-zero binary vector
\mathcal{M}'	transpose of matrix \mathcal{M}
$E[\mathbf{Z}]$	expectation of random variable \mathbf{Z}
$\text{Prob}_{\mathbf{Z}}\{\dots\}$	probability over random variable \mathbf{Z}
$\#\mathcal{A}$	number of elements in set \mathcal{A}
$GF(2^m)$	Galois field of size 2^m

SPN	substitution-permutation network
AES	Advanced Encryption Standard
LP	linear probability
ELP	expected linear probability
LCP	linear characteristic probability
ELCP	expected linear characteristic probability
ALH	approximate linear hull
MALHP	maximum average linear hull probability
DP	differential probability
EDP	expected differential probability
T	number of “core” SPN rounds under consideration
\mathcal{N}_L	data complexity for linear cryptanalysis
\mathcal{B}_l	linear branch number
q	maximum nontrivial LP value over SPN s-boxes

Contents

Abstract	i
Acknowledgments	iii
Statement of Originality	iv
Notation	v
Contents	vii
List of Tables	xiii
List of Figures	xiv
1 Introduction	1
1.1 Motivation for Research	2
1.2 Contributions of Thesis	3
1.3 Outline of Thesis	4
2 Background and Previous Research	6
2.1 Cryptographic Context	6

2.1.1	Information Security Services	7
2.2	Cryptographic Primitives	8
2.3	Ciphers	9
2.3.1	Symmetric-Key Versus Public-Key Ciphers	10
2.3.2	Block Ciphers and Stream Ciphers	11
2.4	Block Cipher Architectures	12
2.4.1	Key-Scheduling Algorithms	12
2.4.2	Substitution-Permutation Networks	14
2.4.3	Feistel Networks	15
2.4.4	Other Block Cipher Architectures	17
2.4.5	The True Random Cipher	18
2.5	Block Cipher Standards	18
2.5.1	The Data Encryption Standard (DES)	19
2.5.2	The Advanced Encryption Standard (AES)	19
2.5.3	The NESSIE Project	22
2.6	Properties of Boolean Mappings	23
2.6.1	Linear Probability	23
2.6.2	Differential Probability	25
2.6.3	Algebraic Degree	26
2.6.4	Completeness	27
2.7	Attacks on Block Ciphers	28
2.7.1	Exhaustive Key Search	29
2.7.2	Linear Cryptanalysis	30
2.7.3	Differential Cryptanalysis	30
2.7.4	Higher-Order Differential Cryptanalysis	31

2.7.5	Algebraic Attacks	32
3	Linear Cryptanalysis	33
3.1	Markov Ciphers	33
3.2	Linear Cryptanalysis of Markov Ciphers	35
3.2.1	The Use of Expected Linear Probability	40
3.2.2	Linear Characteristics	41
3.2.3	Linear Hulls	43
3.3	SPN-Specific Considerations	45
3.4	Extensions of Linear Cryptanalysis	49
3.4.1	Key Ranking	50
3.4.2	Multiple Linear Approximations	51
3.4.3	Generalized Form of Linear Cryptanalysis	51
4	Expected Linear Probability Values for SPNs with Randomly Selected S-Boxes	53
4.1	Approximating the True Random Cipher	54
4.2	SPNs with Randomly Selected S-Boxes	55
4.2.1	Distribution of LP Values for Random S-Boxes	56
4.2.2	Expected ELP Values over all SPNs	58
4.2.3	Recursive Formulation for $C_{\mathbf{a},\mathbf{b}}(A)$	61
4.3	Application to Specific SPN Structure	64
4.3.1	Evaluating the Terms $C_{\mathbf{a},\mathbf{b}}(A)$	64
4.3.2	Computational Results	66
4.4	Conjectures	68
4.5	Summary	70

5	Practical Security Against Linear Cryptanalysis for SPNs with Randomly Selected S-Boxes	71
5.1	Practical Security for Fixed S-Boxes	72
5.2	Practical Security for Random S-Boxes	74
5.2.1	Distribution of ELCP values for Random S-Boxes	74
5.2.2	Practical Security Lower Bound	77
5.3	Computational Results	78
5.3.1	Application to Other SPNs	79
5.4	Summary	80
6	Provable Security Against Linear Cryptanalysis for SPNs with Fixed S-Boxes	81
6.1	Upper Bounding the MALHP for SPNs	82
6.2	Technical Lemmas	87
6.3	General Approach for KMT1 and KMT2	89
6.3.1	Recursive Method for Computing $UB^{[1\dots T]}[\gamma, \hat{\gamma}]$	91
6.3.2	$T = 2$ Case	97
6.3.3	$T \geq 3$ Case	101
6.3.4	Minimum Lengths of $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$	103
6.3.5	Complexity Analysis	103
6.4	The KMT1 Algorithm	105
6.4.1	Complexity Analysis of KMT1	107
6.5	The KMT2 Algorithm	108
6.5.1	Upper Bounding LP Values for One S-Box	108
6.5.2	Upper Bounding ELP Values for One Round	109

6.5.3	Complexity Analysis of KMT2	111
6.6	Summary	111
7	Analysis of Specific SPN Ciphers	115
7.1	Application of KMT1/KMT2 to the AES	115
7.1.1	Computation of $W[\]$ Entries	116
7.1.2	Parallel/Distributed Processing	117
7.1.3	Considerations Specific to KMT1	117
7.1.4	Considerations Specific to KMT2	118
7.2	Linear Cryptanalysis of Q	120
7.2.1	Basic Components of Q	120
7.2.2	High-Level Structure of Q	121
7.2.3	Select LP Values for the Q S-Boxes	123
7.2.4	High Probability Linear Hulls in Q	123
7.2.5	Computational Results	127
7.2.6	Recovering the Full Key	129
7.2.7	Reasons for the Success of Our Attack	130
8	Conclusions	134
8.1	Summary of Thesis	134
8.2	Ideas for Future Research	136
	Bibliography	138
A	Duality Between Linear and Differential Cryptanalysis	149
A.1	Elements of the Duality	149
A.2	Maximum Expected Differential Probability	150

A.3	Upper Bounding the MEDP for SPNs	151
A.4	The KMT1-DC Algorithm	155
A.4.1	Application of KMT1-DC to the AES	156
A.5	The KMT2-DC Algorithm	156
A.5.1	Application of KMT2-DC to the AES	156
Vita		158

List of Tables

3.1	Success rates for linear cryptanalysis (Algorithm 2)	36
5.1	Practical security lower bound for 64-bit SPN with Kam and Davida permutation and randomly selected s-boxes	79
6.1	Upper bound from KMT2 for the AES	85
7.1	Distribution of LP values for the AES s-box	118
7.2	LP values for s-box S	124
7.3	LP values for s-box A	124
7.4	LP values for s-box B	124
7.5	Best ELP values	128
7.6	Corresponding best EDP values from Biham et al. [15]	128
7.7	Best linear hulls for attacking the bytes of $(KB \oplus KW2)$	130
A.1	Elements of duality between linear and differential cryptanalysis . . .	150
A.2	Upper bound from KMT2-DC for the AES	154
A.3	Distribution of DP values for the AES s-box	156

List of Figures

2.1	Basic scenario for two communicating parties	6
2.2	Taxonomy of cryptographic primitives	8
2.3	Operation of a cipher	9
2.4	SPN structure with $N = 16$, $M = n = 4$, $R = 3$	15
2.5	Basic Feistel network architecture	16
2.6	One round of Rijndael (the AES)	20
3.1	Summary of linear cryptanalysis (Algorithm 2)	37
4.1	Distribution of LP values for random bijective 8×8 s-box	57
4.2	SPN with $M = n = 4$ ($N = 16$), $R = 3$, and the permutation of Kam and Davida [50]	65
4.3	$E_{\text{SPN}} \left[\text{ELP}^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for $M = n = 4$ and $\mathbf{a} = \text{D000(hex)}$, $\mathbf{b} =$ $0050(\text{hex})$	67
4.4	$E_{\text{SPN}} \left[\text{ELP}^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for $M = n = 8$ and $wt(\gamma_{\mathbf{a}}) = wt(\gamma_{\mathbf{b}}) = 1$	69
6.1	Upper bounds from KMT1 and KMT2 for the AES	84
6.2	Important values for KMT1 and KMT2	92
6.3	Important values for KMT1 and KMT2 ($T = 2$ case)	97

6.4	General algorithm for upper bounding the MALHP ($T = 2$ case) . . .	112
6.5	General algorithm for upper bounding the MALHP ($T \geq 3$ case) . . .	113
6.6	KMT1 Algorithm ($T \geq 3$ case)	114
7.1	PreSerpent() bitwise permutation	121
7.2	Structure of a full round of Q	123
7.3	Pseudocode for computation of linear hulls over 23 core stages	132
7.4	Pseudocode for other subroutines	133
A.1	Upper bounds from KMT1-DC and KMT2-DC for the AES	153

Chapter 1

Introduction

The past few decades have witnessed the rapid proliferation of computing devices of all sizes and capacities, together with the accompanying growth of networks connecting these devices (the most famous of which is the Internet). This has resulted in unparalleled increases in efficiency and productivity, but has also accentuated the need for techniques to protect digital information in storage and in transmission.

This thesis is concerned with cryptology, the design and analysis of mathematical techniques for securing information. Many such techniques exist, but we will focus our attention on block ciphers, which are considered the “workhorses of cryptology.” A block cipher is a particular kind of algorithm used for scrambling (encrypting) and unscrambling (decrypting) information in order to protect it from an adversary. There is a rich collection of research dealing with block ciphers, and the following thesis contributes to this body of work.

1.1 Motivation for Research

One of the main block cipher architectures is the substitution-permutation network (SPN). Although SPNs have been studied for many years, many mathematical properties of these structures are not fully understood. In particular, some of the most powerful attacks on block ciphers have not been applied to SPNs as extensively as to other cipher architectures (specifically, Feistel networks). The recent adoption of the SPN Rijndael as the U.S. Government Advanced Encryption Standard (AES) indicates widespread confidence in the security of SPNs. At the same time, the vast amount of information that will be protected using the AES and other SPN-based ciphers mandates continued rigorous analysis of this cipher structure.

It is also the case that many attacks on block ciphers have themselves only been partially explored. Often an attack is based on one or more important insights into aspects of cipher behavior that deviate from perfect randomness (according to some measure). Such deviation can be exploited to deduce information about the cipher key. However, in order to make the attack computationally feasible, significant values are frequently approximated, often without bounding the error involved. The implications of these approximations may be critical for the security of the cipher, but may not be entirely understood until long after the attack is first discovered. As a consequence, there are many opportunities for theoretical and computational analysis of existing attacks. One of the most powerful attacks, and an attack to which the above comments apply, is linear cryptanalysis.

An important theoretical underpinning for linear cryptanalysis is the concept of linear hulls. Linear hulls obviate the need for a widely used approximation, allowing a more accurate evaluation of the resistance of ciphers to linear cryptanalysis. However,

because they appear to introduce significant computational complexity, linear hulls have not been adequately studied. As a result, there is a largely unexplored area of research concerning the analysis and application of linear hulls.

1.2 Contributions of Thesis

The principal goal of this thesis is to extend the current understanding of linear cryptanalysis of SPNs. We approach this goal from two main directions. First, we consider SPNs in which the component substitution boxes (s-boxes) are randomly selected. This is an elegant model, and is relevant in light of the fact that a number of block ciphers incorporate pseudorandomly generated s-boxes. We derive a lower bound on the probability that an SPN based on this model is practically secure against linear cryptanalysis, and we give experimental evidence that this probability rapidly approaches 1 with an increasing number of cipher rounds.

The single most important value used in linear cryptanalysis is expected linear probability (ELP). We derive an exact expression for ELP values of an SPN with randomly selected s-boxes, and show experimentally that this expression approaches the corresponding value for the true random cipher, which is generally taken to be the ideal cipher model. This adds quantitative support to the claim that the SPN structure is a good approximation to the true random cipher. We conjecture that this convergence holds for a large class of SPNs.

The second direction from which we approach the principal goal of this thesis is to consider SPNs with fixed s-boxes. Here, our most significant contribution is the introduction of two new algorithms for evaluating the provable security of such SPNs against linear cryptanalysis. These algorithms, named KMT1 and KMT2, are

the first completely general algorithms for this purpose—they can be applied to any SPN, and they yield a measure of provable security that is a function of the number of encryption rounds being evaluated. In contrast, other approaches to this problem either require that the SPN linear transformation have a specific structure, or they compute a single value independent of the number of rounds. By applying KMT1 and KMT2 to the AES, we establish the provable security of the AES against linear cryptanalysis.

The main theoretical basis for much of the new work in this thesis is the concept of linear hulls. One of our aims was to explore the extent of the efficacy of linear hulls; we did so, and found a rich source of ideas. A surprisingly simple example of the applicability of linear hulls arose from our analysis of the Q cipher, an SPN submitted to the European Commission’s NESSIE competition for cryptographic primitives. Without considering linear hulls, the designer of Q evaluates the cipher to be secure against linear cryptanalysis. However, we prove that Q can be broken using linear cryptanalysis based on linear hulls. To our knowledge, this is the first use of linear hulls to break a proposed cipher.

1.3 Outline of Thesis

This thesis is organized as follows.

- In Chapter 2 we survey background material related to cryptology in general, and block ciphers in particular, emphasizing relevant previous research.
- In Chapter 3 we present a thorough explanation of linear cryptanalysis, with particular emphasis on its application to the SPN structure.

- In Chapter 4 we examine expected linear probability values for SPNs with randomly selected s-boxes. We derive an exact expression for these values, and show experimentally that this expression converges to the corresponding value for the true random cipher.
- In Chapter 5 we derive a new lower bound on the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis. We give experimental evidence that this probability rapidly approaches 1 with an increasing number of rounds.
- In Chapter 6 we introduce two new algorithms, KMT1 and KMT2, for evaluating provable security against linear cryptanalysis for SPNs with fixed s-boxes. These are the first completely general algorithms for this purpose, and we use them to establish the provable security of the AES against linear cryptanalysis.
- In Chapter 7 we analyze specific SPNs. First we present detailed information about our application of KMT1 and KMT2 to the AES. We then explain our use of linear hulls to break the Q cipher.
- In Chapter 8 we summarize the results of this thesis, and we give directions for future research.
- In Appendix A we briefly explain the duality between linear and differential cryptanalysis, and we present the dual versions of KMT1 and KMT2.

Chapter 2

Background and Previous Research

2.1 Cryptographic Context

Figure 2.1 depicts the basic scenario relevant to this thesis. Two communicating parties, a sender and a receiver, wish to communicate over an *insecure channel*, such as a phone line or the Internet. We assume the presence of an attacker who is able to interact with the communication channel. This attacker may be *passive* (eavesdropping on transmissions) or *active* (manipulating data in transit).

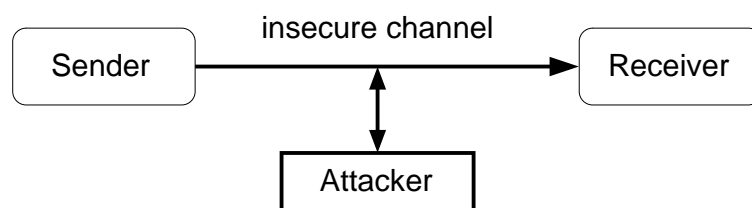


Figure 2.1: Basic scenario for two communicating parties

2.1.1 Information Security Services

Cryptography is the study of mathematical techniques used to provide *information security services* [81, 111]. A large number of information security services can be identified, but the following four are widely considered to be foundational.

1. *Secrecy* (also called *privacy* or *confidentiality*)—the assurance that information stored or being transmitted is inaccessible to unauthorized parties. Note that an attacker may be able to view certain data (e.g., via a wiretap), but cannot extract meaningful information from it.
2. *Integrity*—the protection of data from unauthorized manipulation. Data manipulation includes insertion, deletion, and substitution.
3. *Authentication*—the assurance that two communicating parties are who/what they claim to be (*entity authentication*), and that any data subsequently communicated in fact originates with the claimed sender (*data origin authentication*). Note that data origin authentication implies integrity, since if data has been manipulated it can no longer be attributed to the original sender.
4. *Non-repudiation*—the inability of a party to deny previous commitments or actions. Non-repudiation is important in situations in which disputes may arise over prior transactions. Protocols that ensure non-repudiation typically require the involvement of a trusted third party.

The term *cryptanalysis* refers to techniques used to thwart, or “break,” cryptographic techniques. *Cryptology* is the field of study that encompasses both cryptography and cryptanalysis.

2.2 Cryptographic Primitives

The building blocks of the cryptographic techniques used to provide information security services are called *cryptographic primitives*. Most primitives are functions whose inputs and outputs are elements of certain spaces of finite-length binary strings. Figure 2.2 (from [81]) gives a useful taxonomy of cryptographic primitives.

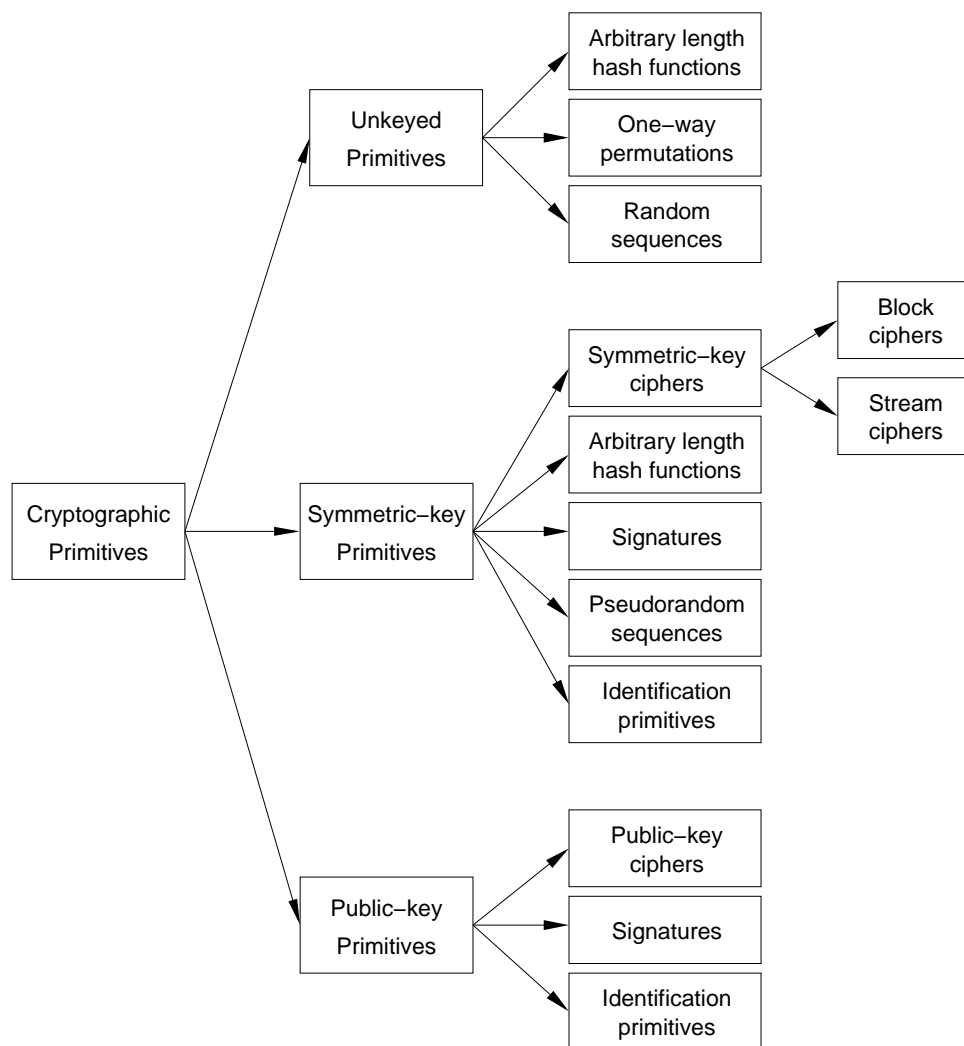


Figure 2.2: Taxonomy of cryptographic primitives

Each specific category of primitives in Figure 2.2 (i.e., any “leaf node” in the tree) represents an entire sub-field of study within cryptography. Apart from a general overview of ciphers in the next section, this thesis deals entirely with block ciphers; the remaining primitives are outside the scope of this work.

2.3 Ciphers

We illustrate ciphers using Figure 2.3. Suppose the sender has a large file to transmit to the receiver. The sender breaks this file into smaller pieces called *plaintexts* (usually of fixed length). Each plaintext (\mathbf{p}) is input into an *encryption algorithm*, which also takes an *encryption key*, \mathbf{k}_e , as a parameter; the resulting output is called a *ciphertext* (\mathbf{c}). The ciphertext is sent over the insecure channel, and the receiver recovers the plaintext using a *decryption algorithm*, which takes a *decryption key*, \mathbf{k}_d , as a parameter. The plaintexts are then reassembled into the original file. The term cipher (or *encryption scheme*) refers to the (parameterized) encryption/decryption algorithms.

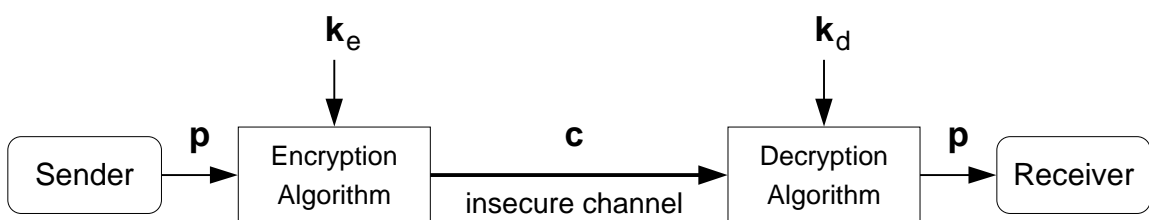


Figure 2.3: Operation of a cipher

2.3.1 Symmetric-Key Versus Public-Key Ciphers

There are two main categories of ciphers. In a *symmetric-key cipher*, either \mathbf{k}_e and \mathbf{k}_d are equal, or they are easily derived from each other; for the remainder of this thesis we will assume that $\mathbf{k}_e = \mathbf{k}_d \stackrel{\text{def}}{=} \mathbf{k}$. Clearly the use of a symmetric-key cipher implies that the sender and receiver must first establish a shared key—this is called the *key-distribution problem*.

In a *public-key cipher* [26] (also called an *asymmetric-key cipher*), there is an asymmetry between the encryption and decryption keys. Each communicating party wishing to receive information possesses a key pair $(\mathbf{k}_e, \mathbf{k}_d)$. The encryption key, \mathbf{k}_e (called the *public key*), can be widely distributed (e.g., on the receiver’s Web page), while the decryption key, \mathbf{k}_d (the *private key*), is generally known only to the receiver. It follows that any party can use \mathbf{k}_e to encrypt information and send it to the receiver, but only the receiver can use \mathbf{k}_d to decrypt this information.

Public-key ciphers provide an elegant solution to the key distribution problem. However public-key ciphers are typically much slower than symmetric-key ciphers (e.g., 1/1000 the speed [112]), and in many cases require much longer keys to achieve the same level of security (an issue in bandwidth-limited environments). As a result, hybrid techniques incorporating symmetric-key and public-key ciphers in a complementary fashion are common. For example, one party can randomly generate a key, \mathbf{k} , to be used in a symmetric-key cipher. This key is then encrypted with the second party’s public key, and sent over the channel. The second party decrypts \mathbf{k} with the corresponding private key, and then both parties switch to an agreed-upon symmetric-key cipher using \mathbf{k} .

2.3.2 Block Ciphers and Stream Ciphers

As shown in Figure 2.2, symmetric-key ciphers are further categorized into *block ciphers* and *stream ciphers*. A block cipher is a bijective mapping from $\{0, 1\}^N$ to $\{0, 1\}^N$, parameterized by $\mathbf{k} \in \{0, 1\}^K$ (N is called the *block size*). Typical block sizes are $N \in \{64, 128\}$, and typical key lengths are $K \in \{128, 192, 256\}$ (key lengths of 56 and 64 bits are common in older block ciphers). A block cipher has the obvious feature that, for a fixed key, a given plaintext will always map to the same ciphertext. (This is true for the straightforward application of a block cipher, called *electronic codebook (ECB) mode*. There are other block cipher modes for which this no longer holds—we do not deal with these [112].)

A stream cipher breaks a message to be encrypted into much smaller plaintexts, typically individual bits (sometimes bytes), $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots$. The key, \mathbf{k} , is expanded into a *keystream*, $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots$. The i^{th} ciphertext is obtained by combining \mathbf{x}_i and \mathbf{z}_i according to some rule (often the XOR operation, which requires that \mathbf{x}_i and \mathbf{z}_i have the same number of bits). It follows that identical plaintexts do not always encrypt to identical ciphertexts. A stream cipher derives its name from the fact that it can be viewed as processing its input as a continuous stream. Note that Figure 2.3 needs to be augmented to correctly depict the operation of a stream cipher, since the encryption and decryption algorithms now have *state*, namely the index i .¹

¹The variable i is adequate to store the state for a *synchronous* stream cipher; for a general stream cipher, the state also depends on the previous plaintexts, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{i-1}$ [112].

2.4 Block Cipher Architectures

Modern block ciphers can trace their roots to a landmark paper by Claude Shannon [107] in which the principles of *confusion* and *diffusion* were outlined. Confusion is the obscuring of the relationship between the plaintext and the ciphertext. Diffusion involves “spreading out” patterns in the plaintext so that they are no longer detectable in the ciphertext [104]. Shannon suggested that confusion and diffusion could be achieved through the use of *substitution* and *linear transformation*,² respectively. The two main block cipher architectures, *substitution-permutation networks* [28] and *Feistel networks* [29], both use substitution and linear transformation to implement Shannon’s principles. Both also are examples of *product ciphers*—ciphers that are constructed by composing two or more encryption operations. In general, a product cipher is stronger than each of its constituent operations. An *iterated cipher* is a product cipher that consists of repeated application of the same encryption step, called a *round* [81]. A round may itself consist of multiple encryption steps; in general, different keying material is used in each round.

Note that for the remainder of this thesis, the terms *cipher* and *block cipher* will be used interchangeably.

2.4.1 Key-Scheduling Algorithms

In most block ciphers, keying material is mixed with the intermediate block in each round. Typically, a separate *key-scheduling algorithm* is used to generate a series of *subkeys* (or *round keys*) from the original key, \mathbf{k} (sometimes called the *master key*).

²In fact, a linear transformation in a cipher is sometimes called a *diffusion layer* [99].

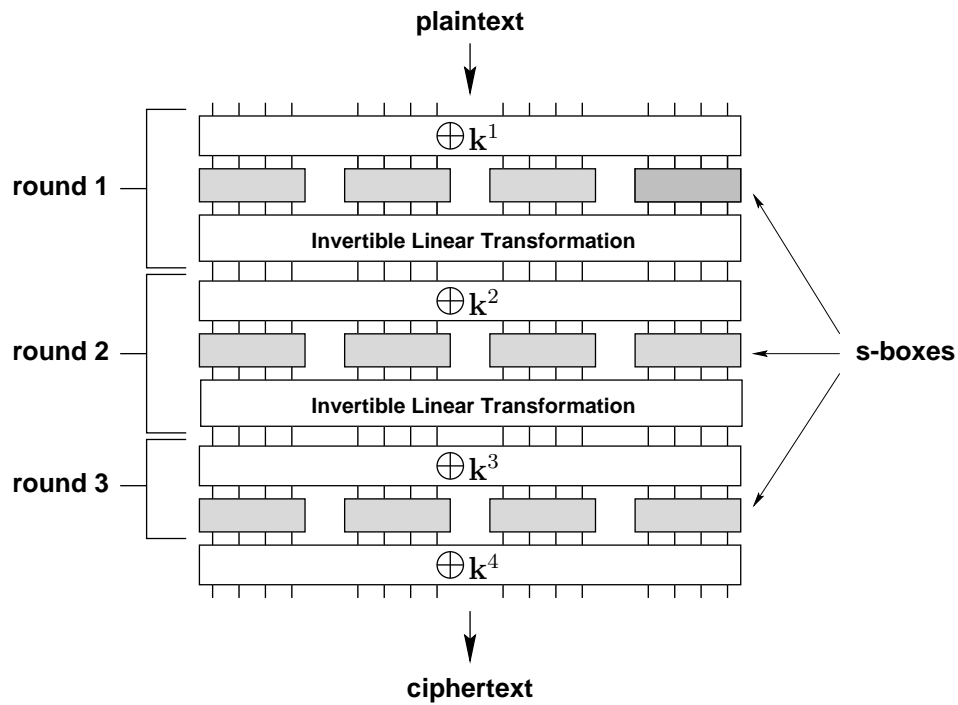
We denote these subkeys $\mathbf{k}^1, \mathbf{k}^2, \mathbf{k}^3, \dots$; subkey \mathbf{k}^r is incorporated into round r . Additional subkeys may also be generated and mixed at other points in the cipher, for example, before the first round or after the last round (this technique, intended to prevent an attacker from knowing the actual input to or output from some part of the cipher, is called *whitening* [104]). We do not focus on the design of key-scheduling algorithms as they are not relevant to our analysis in later chapters. However, it is worth noting that a poorly designed key-scheduling algorithm may introduce significant weaknesses into a cipher, opening the door for certain attacks (e.g., *related-key attacks* [9]). Many cipher designers build cryptographically strong key-scheduling algorithms by incorporating features of the cipher itself—this approach is used by the AES [25], Camellia [6], Twofish [105], and Serpent [4], among others.

Unless stated otherwise, we assume the most general situation for the key, namely that \mathbf{k} is an *independent key* [10], a concatenation of (the appropriate number of) subkeys chosen independently from the uniform distribution on $\{0, 1\}^N$. This assumption has the advantage of simplifying many kinds of analysis. It generally represents the most difficult keying situation to attack, since key-scheduling algorithms typically generate only a small subset of all possible vectors of subkeys, and, as noted above, may introduce weaknesses that can be exploited separately from the encryption/decryption algorithms. Therefore, it is often prudent for a cryptanalyst to assume the use of an independent key, since an attack that is successful in this model may have a higher success rate when there are correlations among the subkeys. As well, the assumption of an independent key is frequently made by a cipher designer when evaluating resistance to various attacks, but features of the key-scheduling algorithm should also be given careful consideration.

2.4.2 Substitution-Permutation Networks

An R -round substitution-permutation network (SPN) [28] requires $(R+1)$ N -bit subkeys, $\mathbf{k}^1, \mathbf{k}^2, \dots, \mathbf{k}^R, \mathbf{k}^{R+1}$. Each round consists of three *stages*, or *layers*. In the *key-mixing stage*, the N -bit round input is bitwise XOR'd with the subkey for that round. In the *substitution stage*, the resulting block is partitioned into M subblocks of size n ($N = Mn$), and each subblock becomes the input to a bijective $n \times n$ *substitution box* (*s-box*)—a bijective mapping from $\{0, 1\}^n$ to $\{0, 1\}^n$. In the *linear transformation stage*, the output from the substitution stage is processed through an invertible N -bit linear transformation. (Classically, the linear transformation was a bitwise permutation, hence the origin of the name *substitution-permutation network* [28].) If we represent the linear transformation as an invertible $N \times N$ binary matrix, we will use \mathcal{L} to denote this matrix. The linear transformation is usually omitted from the last round, since it is easily shown that its inclusion adds no cryptographic strength to the SPN. A final subkey, \mathbf{k}^{R+1} , is XOR'd with the output of round R to form the ciphertext. We will assume that the same linear transformation is used in each round. Unless specified otherwise, no restriction is placed on the choice of s-boxes. Figure 2.4 depicts an example SPN with $N = 16$, $M = n = 4$, and $R = 3$.

Decryption is accomplished by running the SPN “backwards.” Subkey \mathbf{k}^{R+1} is first XOR'd with the ciphertext, and then in each round r (from R down to 1), the inverse linear transformation is applied, followed by the inverse s-boxes, and the resulting block is XOR'd with \mathbf{k}^r .

Figure 2.4: SPN structure with $N = 16$, $M = n = 4$, $R = 3$

2.4.3 Feistel Networks

A Feistel network [29] is a block cipher that modifies *half* of the current block in each round (this requires an even block size). An R -round Feistel network makes use of R subkeys, $\mathbf{k}^1, \mathbf{k}^2, \dots, \mathbf{k}^R$ (the length of the subkeys depends on the round structure). Let the left and right halves of the N -bit input to round r be denoted \mathbf{x}_L^r and \mathbf{x}_R^r , respectively. The right half, \mathbf{x}_R^r , becomes the input to a *round function*, $f_r : \{0, 1\}^{N/2} \rightarrow \{0, 1\}^{N/2}$, which also takes \mathbf{k}^r as a parameter. The output from f_r is XOR'd with \mathbf{x}_L^r to form \mathbf{x}_R^{r+1} (the right half of the input to the next round), while \mathbf{x}_R^r is preserved unchanged as \mathbf{x}_L^{r+1} . This swapping of half blocks occurs in every round except the last. With the above notation, the plaintext is given by $\mathbf{p} = \langle \mathbf{x}_L^1, \mathbf{x}_R^1 \rangle$

and the ciphertext is given by $\mathbf{c} = \langle \mathbf{x}_L^{R+1}, \mathbf{x}_R^{R+1} \rangle$. Figure 2.5 depicts the basic Feistel network structure.

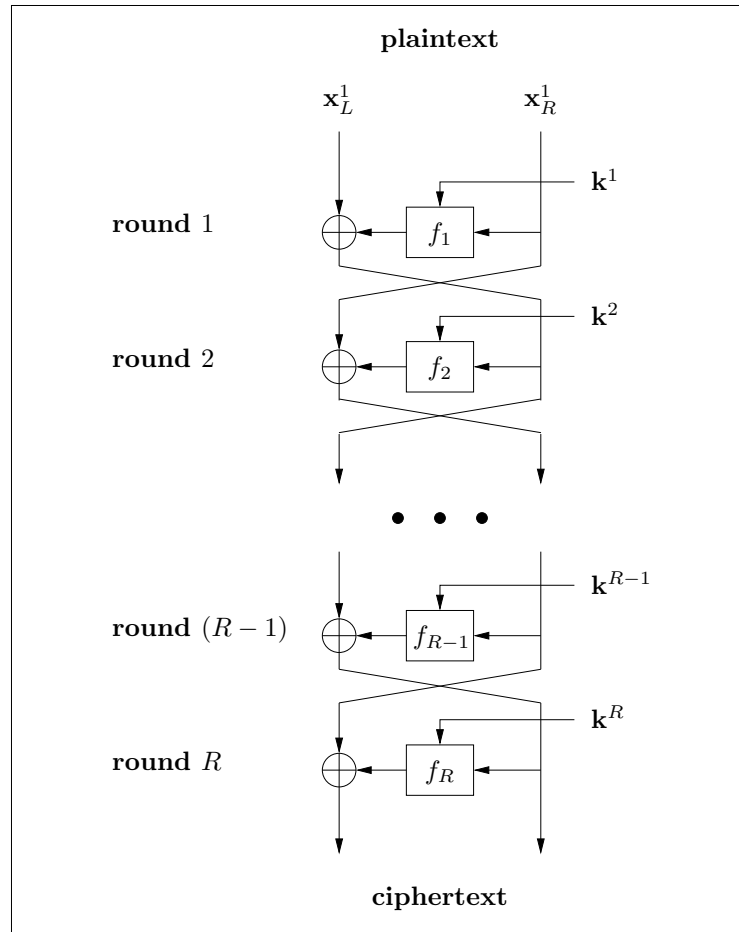


Figure 2.5: Basic Feistel network architecture

There are many approaches to the design of Feistel network round functions, but a common theme is to incorporate the basic features of an SPN round, i.e., some arrangement of s-boxes and linear transformations [6, 29, 51, 52, 53]. It is easy to show, however, that a Feistel network round function does not need to be invertible, allowing greater flexibility in its design.

In terms of implementation, one of the main advantages of the Feistel network structure is that encryption and decryption are essentially identical operations—a ciphertext is decrypted by processing it through the encryption algorithm, but reversing the order of the round functions (and corresponding subkeys). If the same round function is used in each round (a common approach), only the order of the subkeys needs to be reversed. This eliminates the need to generate/store inverse components.

Schneier and Kelsey introduced the concept of *unbalanced Feistel networks* [103], or UFNs. In a UFN, \mathbf{x}_L^r and \mathbf{x}_R^r are not equal in size (conventional Feistel networks are called *balanced* in [103]). If the lengths of \mathbf{x}_L^r and \mathbf{x}_R^r are s and t bits, respectively ($s + t = N$), then $f_r : \{0, 1\}^t \rightarrow \{0, 1\}^s$. The inputs to the next round, \mathbf{x}_L^{r+1} and \mathbf{x}_R^{r+1} (lengths s and t bits, respectively), are defined such that

$$\mathbf{x}_L^{r+1} \parallel \mathbf{x}_R^{r+1} = \mathbf{x}_R^r \parallel (f_r(\mathbf{x}_R^r) \oplus \mathbf{x}_L^r) ,$$

where \parallel is the concatenation operator. Clearly this is a modified version of the structure in Figure 2.5. Schneier and Kelsey give preliminary arguments that in certain cases a UFN may have increased resistance to certain attacks. Variations of the UFN approach are used in ciphers such as CAST-256 [3] and MARS [19].

2.4.4 Other Block Cipher Architectures

Although SPNs and Feistel networks are the most common block cipher architectures, there are a number of ciphers that do not adhere to either structure. However, most retain the basic concept of constructing a cipher from repeated rounds. We mention a couple of examples here.

In the block cipher IDEA [69, 70], which has a 64-bit block size and consists of 8

rounds, the round input is split into four 16-bit words, and these are combined with each other and with six 16-bit subkeys using a combination of binary operations on $\{0, 1\}^{16}$ from different algebraic groups. IDEA has been extensively analyzed and widely implemented [104]; it appears that this mixing of algebraic groups is a good source of security.

The block cipher RC6 [100] has a 128-bit block size and consists of 20 rounds. RC6 can essentially be viewed as two 64-bit Feistel networks operating in parallel, with interactions occurring in each round. RC6 makes extensive use of *data-dependent rotations*—bitwise rotations of data words in which the amount of rotation depends on other intermediate values.

2.4.5 The True Random Cipher

For a given block size N , the *true random cipher* (or *ideal cipher*) is the key-parameterized family of all bijective mappings from $\{0, 1\}^N$ to $\{0, 1\}^N$ such that each mapping is realized by exactly one key. For common block sizes, the true random cipher cannot be practically implemented since it would require a key of astronomical length (approximately $N \times 2^N$ bits) [81]. However, the true random cipher is important theoretically, and is generally considered to be the ideal block cipher model [31].

2.5 Block Cipher Standards

In the history of modern block ciphers, standardization initiatives by governments and by various national and international bodies have played a significant role. We briefly discuss three important examples.

2.5.1 The Data Encryption Standard (DES)

The first major initiative was a 1973 call for proposals for a cipher standard by the National Bureau of Standards (NBS) (now the National Institute of Standards and Technology (NIST)) of the U.S. Department of Commerce. At this time, cryptography outside of military establishments was in its infancy; the NBS received only one serious candidate—a block cipher called Lucifer [110], developed by IBM in the late 1960s and early 1970s. On January 15, 1977, the NBS published a modified version of Lucifer called the Data Encryption Standard (DES) [32]. DES is a 16-round Feistel network with a 64-bit block size and a 56-bit key (from the beginning, the small key size was a source of criticism [27]). The publication of DES marked the beginning of the widespread study of block ciphers. Many cryptanalytic attacks have been developed in the context of trying to find weaknesses in DES [14, 66, 71, 74, 113]. In addition, many DES-like block ciphers have since been proposed and studied [2, 18, 64, 108].

2.5.2 The Advanced Encryption Standard (AES)

In September 1997, NIST began a process to select a replacement for DES, to be called the Advanced Encryption Standard (AES) [86]. Candidates for the AES were required to be block ciphers supporting a block size of 128 bits and key lengths of 128, 192, and 256 bits. Unofficially, the AES was expected to be at least as efficient as DES, and significantly more secure. NIST received 15 candidate algorithms, and these were evaluated through an open process that included a series of public conferences. In August 1999, the following five finalists were announced: MARS (IBM Corp.), RC6 (RSA Laboratories), Rijndael (Daemen and Rijmen), Serpent (Anderson et al.), and Twofish (Schneier et al.) [87]. In October 2000, NIST selected Rijndael as the AES.

On November 26, 2001, the AES was published in the U.S. Federal Register [33].

The Rijndael Block Cipher

Rijndael (pronounced “Rhine-doll”) is an SPN with 16 8×8 s-boxes in each substitution stage. A single fixed s-box is used throughout the cipher. The linear transformation consists of two steps: a byte permutation, and the parallel application of four copies of a highly diffusive 32-bit linear transformation. A single Rijndael round is depicted in Figure 2.6. (Technically, in a Rijndael round the subkey is XOR’d *after* the linear transformation [25], but since an additional subkey is XOR’d before the first round, Rijndael can be viewed as conforming to the SPN structure as given in Section 2.4.2.)

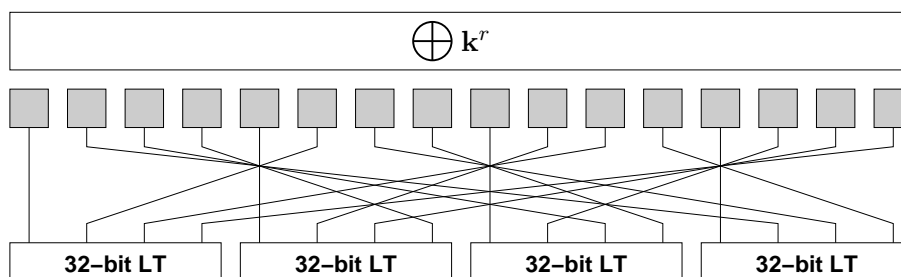


Figure 2.6: One round of Rijndael (the AES)

Rijndael actually supports block sizes and key lengths from 128 to 256 bits, in increments of 32 bits. However, as stated above, the AES is restricted to a 128-bit block size, and key lengths of 128, 192, and 256 bits; this is the only difference between Rijndael and the AES [25]. The number of rounds varies according to block size and key length. For a 128-bit block size, the possible values are: 128-bit key \rightarrow 10 rounds, 192-bit key \rightarrow 12 rounds, 256-bit key \rightarrow 14 rounds. Note that we will usually refer

to the AES instead of to Rijndael.

Prior to the announcement of the AES, Feistel networks were generally more widely studied than SPNs, although many important results about SPNs were published [1, 7, 8, 21, 38, 39, 40, 41, 109, 117, 118]. (The two architectures, although similar, are sufficiently different that results may not translate readily from one to the other.) With the adoption of Rijndael as the AES, there has been an increased interest in SPNs. In the past few years, a number of publications dealing with the security of SPNs have appeared [16, 55, 96, 97, 101], including several works by the author [58, 59, 60, 61, 62]. In this thesis we focus exclusively on SPNs. In Chapter 6 and Chapter 7 we discuss analysis specific to the AES.

AES-Like SPNs

An *AES-like* (or *Rijndael-like*) SPN [96] has the following structure. Each substitution stage contains 16 8×8 s-boxes (not necessarily identical), and the linear transformation consists of two steps:

1. A byte permutation $\pi : \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ with the property that if the input and output for π are viewed as consisting of four 32-bit words, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ and $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4)$, respectively, then each of the four bytes in \mathbf{y}_i comes from a different \mathbf{x}_j . (At the bit level, this kind of permutation was investigated by Kam and Davida [50].)
2. A linear transformation $\theta : \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ that consists of the parallel application of four 32-bit invertible linear transformations, $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$, with the condition that each θ_i is maximally diffusive (see Remark 3.3.12).

The ciphers SQUARE [24] and CRYPTON [72] are examples of AES-like SPNs.

2.5.3 The NESSIE Project

NESSIE was a three-year project within the European Commission's Information Society Technologies (IST) Programme; NESSIE stands for *New European Schemes for Signature, Integrity, and Encryption*. The main goal of the NESSIE project was to select a set of strong cryptographic primitives through an open submission and evaluation process, with the intention that these will play an important role in European industry, and will be adopted by various European and international standardization bodies.

The NESSIE call for primitives was published in March 2000, resulting in 42 submissions in a variety of categories (including block ciphers, stream ciphers, and public-key ciphers). In September 2001, this was reduced to 24 candidates based on initial evaluation. Finally, in February 2003, the NESSIE process ended with the recommendation of 12 primitives. The recommended portfolio included three block ciphers: MISTY1 [77] (Mitsubishi Electric Corp.), Camellia [6] (Nippon Telegraph and Telephone Corp.), and SHACAL-2 [35] (Gemplus). MISTY1 is a Feistel network with a block size of 64 bits; Camellia is a Feistel network with a block size of 128 bits; SHACAL-2 has a block size of 160 bits and is based directly on the Secure Hash Algorithm (SHA) [34]. The final NESSIE portfolio was also augmented with five existing standard primitives—the single block cipher added was the AES.

In Section 7.2 we describe our analysis of one of the NESSIE candidates, an SPN called Q [79]. Applying linear cryptanalysis based on linear hulls (Chapter 3), we discovered a significant weakness in this cipher.

2.6 Properties of Boolean Mappings

Since a block cipher consists of a relatively small number of distinct components (Boolean mappings), the security of the cipher is critically dependent on the mathematical properties of these components. Failure to satisfy certain criteria can lead directly to attacks (this is the case for our attack on the Q cipher in Section 7.2), or at least to predictability of behavior that may be exploited in the future. Much research has been devoted to individual properties of Boolean mappings [17, 20, 83, 88, 89, 93, 94, 115], as well as to interrelationships among these properties (good surveys are given in [80, 92, 106]). As this subject is vast, we limit our consideration to a small number of properties here (those that will be applicable later in the thesis). Note that a Boolean mapping $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ can be viewed as consisting of d functions $B = (f_1, f_2, \dots, f_d)$, where $f_j : \{0, 1\}^d \rightarrow \{0, 1\}$.

Remark 2.6.1. In an SPN, all of the component mappings are bijective. Since SPNs are the subject of this thesis, we define the properties in this section in the context of bijective mappings $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$. However, all of these properties extend naturally to arbitrary mappings from $\{0, 1\}^d$ to $\{0, 1\}^e$, where d and e are any positive integers.

2.6.1 Linear Probability

Often s-boxes are the only nonlinear components of a cipher (this is true for SPNs); if linear s-boxes were used, the entire cipher would be an affine mapping, and thus trivially broken [37]. Several definitions exist that capture the extent to which a Boolean mapping is nonlinear (some of which are equivalent) [80, 94]. We will use *linear probability* (LP) [114].

Definition 2.6.2. Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be bijective, and let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^d$ be fixed. If $\mathbf{X} \in \{0, 1\}^d$ is a uniformly distributed random variable, then the linear probability $LP(\mathbf{a}, \mathbf{b})$ is defined as

$$LP(\mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} (2 \cdot \text{Prob}_{\mathbf{X}} \{\mathbf{a} \bullet \mathbf{X} = \mathbf{b} \bullet B(\mathbf{X})\} - 1)^2. \quad (2.1)$$

If B is parameterized by a key, \mathbf{k} , we write $LP(\mathbf{a}, \mathbf{b}; \mathbf{k})$, and the expected linear probability $ELP(\mathbf{a}, \mathbf{b})$ is defined as

$$ELP(\mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} E_{\mathbf{K}} [LP(\mathbf{a}, \mathbf{b}; \mathbf{K})],$$

where \mathbf{K} is a random variable uniformly distributed over the space of keys.

We can view the terms $LP(\mathbf{a}, \mathbf{b})$ ($ELP(\mathbf{a}, \mathbf{b})$) as entries in a $2^d \times 2^d$ table in the obvious way. The values \mathbf{a} and \mathbf{b} in Definition 2.6.2 are called input and output *masks*, respectively (Daemen and Rijmen [25] use the term *selection pattern*). For our purposes, the bijective mapping B will be an s-box, a single encryption round, or a sequence of consecutive encryption rounds.

Note that LP values lie in the interval $[0, 1]$ (hence the use of the word “probability,” although they are not actual probabilities). The intuition for Definition 2.6.2 is as follows. If \mathbf{X} and $B(\mathbf{X})$ are uncorrelated in the sense we want to quantify, then $\mathbf{a} \bullet \mathbf{X}$ and $\mathbf{b} \bullet B(\mathbf{X})$ will be equal exactly half the time, i.e., with probability 0.5, and the corresponding LP value will be 0. A nonzero LP value indicates a correlation between the input and output of B , with a higher value indicating a stronger correlation.³ If $LP(\mathbf{a}, \mathbf{b}) = 1$, then either $\mathbf{a} \bullet \mathbf{X} = \mathbf{b} \bullet B(\mathbf{X})$ with probability 1, or $\mathbf{a} \bullet \mathbf{X} \neq \mathbf{b} \bullet B(\mathbf{X})$ with probability 1. It is worth stating some trivial cases: $LP(\mathbf{0}, \mathbf{0}) = 1$, $LP(\mathbf{a}, \mathbf{0}) = 0$ for $\mathbf{a} \neq \mathbf{0}$, and $LP(\mathbf{0}, \mathbf{b}) = 0$ for $\mathbf{b} \neq \mathbf{0}$.

³In the terminology of Daemen et al., $LP(\mathbf{a}, \mathbf{b})$ is the square of entry $[\mathbf{b}, \mathbf{a}]$ in the *correlation matrix* for B [23].

The following lemma derives immediately from Parseval's Theorem [80].

Lemma 2.6.3. *Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be a bijective mapping parameterized by a key, \mathbf{k} , and let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^d$. Then*

$$\begin{aligned} \sum_{\mathbf{x} \in \{0, 1\}^d} LP(\mathbf{a}, \mathbf{x}; \mathbf{k}) &= \sum_{\mathbf{x} \in \{0, 1\}^d} LP(\mathbf{x}, \mathbf{b}; \mathbf{k}) = 1 \\ \sum_{\mathbf{x} \in \{0, 1\}^d} ELP(\mathbf{a}, \mathbf{x}) &= \sum_{\mathbf{x} \in \{0, 1\}^d} ELP(\mathbf{x}, \mathbf{b}) = 1. \end{aligned}$$

Large linear probability values are exploited by linear cryptanalysis (Section 2.7.2, Chapter 3, and the remainder of this thesis).

2.6.2 Differential Probability

Differential probability (DP) [114] quantifies correlations between input and output XOR differences for a Boolean mapping.

Definition 2.6.4. *Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be bijective, and let $\Delta\mathbf{x}, \Delta\mathbf{y} \in \{0, 1\}^d$ be fixed. If $\mathbf{X} \in \{0, 1\}^d$ is a uniformly distributed random variable, then the differential probability $DP(\Delta\mathbf{x}, \Delta\mathbf{y})$ is defined as*

$$DP(\Delta\mathbf{x}, \Delta\mathbf{y}) \stackrel{\text{def}}{=} \text{Prob}_{\mathbf{x}} \{B(\mathbf{x}) \oplus B(\mathbf{x} \oplus \Delta\mathbf{x}) = \Delta\mathbf{y}\}. \quad (2.2)$$

If B is parameterized by a key, \mathbf{k} , we write $DP(\Delta\mathbf{x}, \Delta\mathbf{y}; \mathbf{k})$, and the expected differential probability $EDP(\Delta\mathbf{x}, \Delta\mathbf{y})$ is defined as

$$EDP(\Delta\mathbf{x}, \Delta\mathbf{y}) \stackrel{\text{def}}{=} E_{\mathbf{K}} [DP(\Delta\mathbf{x}, \Delta\mathbf{y}; \mathbf{K})],$$

where \mathbf{K} is a random variable uniformly distributed over the space of keys.

We can view the terms $DP(\Delta\mathbf{x}, \Delta\mathbf{y})$ ($EDP(\Delta\mathbf{x}, \Delta\mathbf{y})$) as entries in a $2^d \times 2^d$ table. Large differential probability values are exploited by differential cryptanalysis (Section 2.7.3). Note that linear and differential probability are related according to the following theorem [114].

Theorem 2.6.5. *Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be bijective, and let $\mathbf{a}, \mathbf{b}, \Delta\mathbf{x}, \Delta\mathbf{y} \in \{0, 1\}^d$ be fixed. Then*

$$\begin{aligned} LP(\mathbf{a}, \mathbf{b}) &= \frac{1}{2^d} \sum_{\mathbf{u}, \mathbf{v} \in \{0, 1\}^d} (-1)^{(\mathbf{a} \bullet \mathbf{u}) + (\mathbf{b} \bullet \mathbf{v})} DP(\mathbf{u}, \mathbf{v}) \\ DP(\Delta\mathbf{x}, \Delta\mathbf{y}) &= \frac{1}{2^d} \sum_{\mathbf{u}, \mathbf{v} \in \{0, 1\}^d} (-1)^{(\Delta\mathbf{x} \bullet \mathbf{u}) + (\Delta\mathbf{y} \bullet \mathbf{v})} LP(\mathbf{u}, \mathbf{v}). \end{aligned}$$

Remark 2.6.6. Many researchers refer to the *XOR table* of $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ [93]. This is a $2^d \times 2^d$ table, denoted $XOR(\cdot, \cdot)$, defined by

$$XOR(\Delta\mathbf{x}, \Delta\mathbf{y}) \stackrel{\text{def}}{=} \# \{ \mathbf{x} \in \{0, 1\}^d : B(\mathbf{x}) \oplus B(\mathbf{x} \oplus \Delta\mathbf{x}) = \Delta\mathbf{y} \},$$

for $\Delta\mathbf{x}, \Delta\mathbf{y} \in \{0, 1\}^d$. This is obviously equivalent to Definition 2.6.4, since

$$DP(\Delta\mathbf{x}, \Delta\mathbf{y}) = \frac{XOR(\Delta\mathbf{x}, \Delta\mathbf{y})}{2^d}.$$

2.6.3 Algebraic Degree

Any boolean function $f : \{0, 1\}^d \rightarrow \{0, 1\}$ can be written as a polynomial in the input bits,

$$f(x_1, x_2, \dots, x_d) = a_0 + \sum_{1 \leq i \leq d} a_i x_i + \sum_{1 \leq i < j \leq d} a_{i,j} x_i x_j + \dots + a_{1,2,\dots,d} x_1 x_2 \dots x_d,$$

for some coefficients $a_0, \dots, a_{1,2,\dots,d} \in \{0, 1\}$, where multiplication and addition (in $GF(2)$) correspond to the bitwise AND and XOR operations, respectively. This

is called the *algebraic normal form* of f , and can be obtained by a simple matrix operation (the *algebraic normal transform*) [98]. The *degree* of a term in the algebraic normal form of f is the number of distinct x_i in the term. The degree of f , denoted $\text{deg}(f)$, is the highest degree of any term with a nonzero coefficient (this is also called the *nonlinear order* of f [80]), and the degree of a mapping $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ is the highest degree of its d constituent functions. Ciphers whose s-boxes have low algebraic degree may be vulnerable to *higher-order differential cryptanalysis* (Section 2.7.4).

2.6.4 Completeness

Kam and Davida [50] defined the property of *completeness*.

Definition 2.6.7. *Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$. Then B is called complete if every output bit depends on every input bit. Formally, B is complete if, for all $1 \leq i, j \leq d$, there exist $\mathbf{x}, \mathbf{y} \in \{0, 1\}^d$ such that \mathbf{x} and \mathbf{y} differ in exactly bit i , and $B(\mathbf{x})$ and $B(\mathbf{y})$ differ in at least bit j .*

Clearly if $B = (f_1, f_2, \dots, f_d)$ is complete, then the algebraic normal transform (Section 2.6.3) of each $f_j : \{0, 1\}^d \rightarrow \{0, 1\}$ has the following property: every input bit x_i appears in at least one term with a nonzero coefficient.

Lemma 2.6.8. *Let $d \geq 2$, and let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be linear and invertible. Then B is not complete.*

Proof. Let the input to $B = (f_1, f_2, \dots, f_d)$ be denoted $\mathbf{x} = (x_1, x_2, \dots, x_d)$. Since B is linear, the algebraic normal transform of f_j must be a sum of terms x_i . Since B is complete, every x_i must appear, i.e.,

$$f_j(x_1, x_2, \dots, x_d) = x_1 + x_2 + \dots + x_d.$$

It follows that all the f_j are identical, contradicting the invertibility of B . Therefore, B cannot be complete. \square

From Lemma 2.6.8 we know that an SPN linear transformation is not complete. It is important to choose s-boxes and linear transformations so that an SPN (or any block cipher) is complete after some (relatively small) number of rounds. Otherwise, the cipher can be decomposed into simpler mappings that may be vulnerable to attack [50]. The completeness property is important to our work in Chapter 4, specifically Section 4.4.

2.7 Attacks on Block Ciphers

There exists a large (and growing) collection of attacks on block ciphers. Generally, the goal of an attack is to gain the ability to decrypt any ciphertext. Typically this involves deriving the key (a *total break*), although it may be possible to construct an algorithm that decrypts ciphertexts without knowledge of the key (*global deduction*) [67].

Attacks on block ciphers can be categorized as follows, depending on the information available to the attacker [81].

1. *Ciphertext-only*: attacker possesses one or more ciphertexts
2. *Known-plaintext*: attacker possesses one or more plaintexts and the corresponding ciphertexts
3. *Chosen-plaintext (Chosen-ciphertext)*: attacker is able to choose a set of plaintexts (ciphertexts) to be submitted for encryption (decryption) in order to obtain the corresponding ciphertexts (plaintexts)

4. *Adaptive chosen-plaintext (Adaptive chosen-ciphertext)*: attacker is able to submit plaintexts (ciphertexts) for encryption (decryption), with the freedom to base later choices on the results of earlier submissions

The *data complexity* of an attack is the number of data items required (ciphertexts or $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs, as appropriate). The *time complexity* (or *work factor*) of an attack is the number of steps required, where a “step” is often a single encryption, but may be some other appropriate computational unit [104].

Remark 2.7.1. Most authors use *time complexity* to denote the number of computational steps other than those required to process the data items used. However, this can give the misleading impression that the time required for such processing is negligible, which it is not, since at least a constant (nonzero) amount of time is needed for each data item. It follows that the “real” time complexity is the maximum of the data complexity and the number of additional processing steps. However, given this clarification, we will adhere to conventional terminology.

We now describe several attacks on block ciphers. We continue to use K to denote the number of bits in the key.

2.7.1 Exhaustive Key Search

Given a known $\langle \text{plaintext}, \text{ciphertext} \rangle$ pair, $\langle \mathbf{p}, \mathbf{c} \rangle$, *exhaustive key search* involves encrypting \mathbf{p} with each of the 2^K keys, discarding any key that does not produce the matching ciphertext, \mathbf{c} . A small number of additional $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs may be required to uniquely identify the correct key. Exhaustive key search is generally considered the “benchmark” against which other attacks are measured. A *theoretical break* (or *academic break*) against a block cipher is an attack with time

complexity less than that of exhaustive key search, i.e., less than 2^K .

2.7.2 Linear Cryptanalysis

Linear cryptanalysis, introduced by Matsui in 1993 [74], is a known-plaintext attack (ciphertext-only in certain cases) that is considered to be one of the most powerful attacks on block ciphers. Linear cryptanalysis was the first attack actually implemented to break DES [76]—Matsui carried out this experimental break using 2^{43} known $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs and time complexity 2^{30} . A precursor to linear cryptanalysis was introduced in 1992 by Matsui and Yamagishi, and used to attack the block cipher FEAL (a DES-like cipher) [78]. Linear cryptanalysis requires the existence of relatively large expected linear probability values (Section 2.6.1) over the entire cipher minus one or more outer rounds. In Chapter 3 we give a detailed description of linear cryptanalysis, and in the remainder of this thesis we focus on linear cryptanalysis of SPNs.

2.7.3 Differential Cryptanalysis

Differential cryptanalysis is a chosen-plaintext attack presented by Biham and Shamir in 1990 [11, 12, 13, 14]. (A differential-like attack was also published by Murphy in 1990, and applied to FEAL [84].) Differential cryptanalysis was the first attack able to break DES faster than exhaustive key search, with data complexity 2^{47} and time complexity 2^{37} [14]. Differential cryptanalysis makes use of relatively large differential probability values (Section 2.6.2) over the entire cipher minus one or more outer rounds. Given sufficiently many chosen $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs, the subkeys for these outer rounds can be determined with high probability. These rounds can then

be stripped off, and differential cryptanalysis can be reapplied to obtain the remaining subkeys (or other techniques can be used to derive the remaining subkeys from the known subkey(s)).

2.7.4 Higher-Order Differential Cryptanalysis

Higher-order differential cryptanalysis makes use of the concept of the *derivative* of a Boolean mapping, due to Lai [68].

Definition 2.7.2. Let $B : \{0, 1\}^d \rightarrow \{0, 1\}^d$, and let $\mathbf{a} \in \{0, 1\}^d$. The derivative of B at \mathbf{a} is the mapping $\Delta_{\mathbf{a}}B : \{0, 1\}^d \rightarrow \{0, 1\}^d$, given by

$$\Delta_{\mathbf{a}}B(\mathbf{x}) \stackrel{\text{def}}{=} B(\mathbf{x} \oplus \mathbf{a}) \oplus B(\mathbf{x}).$$

The i^{th} derivative of B at $(\mathbf{a}_i, \dots, \mathbf{a}_1)$ is defined recursively by

$$\Delta_{\mathbf{a}_i, \dots, \mathbf{a}_1}^{(i)} B \stackrel{\text{def}}{=} \Delta_{\mathbf{a}_i} \left(\Delta_{\mathbf{a}_{i-1}, \dots, \mathbf{a}_1}^{(i-1)} B \right).$$

Lai proved the following inequality concerning algebraic degree:

$$\text{deg}(\Delta_{\mathbf{a}}B) \leq \text{deg}(B) - 1.$$

It follows that if $\text{deg}(B) = m$, then the $(m + 1)^{\text{st}}$ derivative of B is 0. Higher-order differential cryptanalysis, due to Knudsen [66], makes use of this observation to break ciphers whose component mappings have low algebraic degree. The attack involves setting up a series of equations involving such derivatives and incorporating certain subkey bits (these equations are greatly simplified by the low algebraic degree), and then using exhaustive search to determine the correct values of these subkey bits. Jakobsen and Knudsen [44, 45] demonstrate that ciphers that are secure against traditional differential cryptanalysis may be vulnerable to higher-order differential cryptanalysis.

2.7.5 Algebraic Attacks

Algebraic attacks exploit cipher components that can be represented by operations in certain algebraic structures (e.g., groups, rings, fields [42]). These attacks attempt to derive and solve equations relating various input, output, and key values, with the goal of determining key bits. Interest in such attacks has increased since the introduction of the AES—this is due to the fact that operations in the finite field $GF(2^8)$ were fundamental to the design of the AES [25]. The AES s-box is based on inversion in a particular representation of $GF(2^8)$ (together with an affine mapping), and the 32-bit linear transformation inside the 128-bit AES linear transformation (Figure 2.6) consists of multiplication by a 4×4 matrix of elements from $GF(2^8)$.

Ferguson et al. [30] show that any AES ciphertext byte can be written as an equation involving plaintext and key bytes with the operations addition, multiplication, and inversion in $GF(2^8)$. This equation contains approximately 2^{50} terms, which is feasible to store and manipulate. At this time, there is no practical algorithm for solving such an equation, but the existence of such a (relatively) simple representation is significant.

Murphy and Robshaw [85] prove that the AES can be embedded within a new block cipher (the BES) that consists exclusively of operations in $GF(2^8)$. This allows an AES encryption to be written as a system of 5248 equations, all of which have algebraic degree at most 2 (quadratic), and all of which are very sparse. There is currently no efficient algorithm for solving such a system of equations, but it follows that the security of the AES depends on the continued hardness of this problem.

Chapter 3

Linear Cryptanalysis

In Sections 3.1 and 3.2 we develop the theory of linear cryptanalysis in the general context of *Markov ciphers*. In Section 3.3 we focus on linear cryptanalysis of SPNs, and in Section 3.4 we describe extensions to the basic linear cryptanalytic attack.

3.1 Markov Ciphers

Definition 3.1.1 ([70]). *Let $\mathcal{E} : \{0, 1\}^N \rightarrow \{0, 1\}^N$ be an R -round block cipher for which round r is given by the function $\mathbf{y} = \epsilon_r(\mathbf{x}; \mathbf{k}^r)$ ($\mathbf{x} \in \{0, 1\}^N$ is the round input, and \mathbf{k}^r is the round- r subkey). Then \mathcal{E} is a Markov cipher with respect to the group operation XOR (\oplus) on $\{0, 1\}^N$ if, for $1 \leq r \leq R$ and any $\mathbf{x}, \Delta\mathbf{x}, \Delta\mathbf{y} \in \{0, 1\}^N$,*

$$\begin{aligned} \text{Prob}_{\mathbf{K}} \{ \epsilon_r(\mathbf{x}; \mathbf{K}) \oplus \epsilon_r(\mathbf{x} \oplus \Delta\mathbf{x}; \mathbf{K}) = \Delta\mathbf{y} \} = \\ \text{Prob}_{\mathbf{X}, \mathbf{K}} \{ \epsilon_r(\mathbf{X}; \mathbf{K}) \oplus \epsilon_r(\mathbf{X} \oplus \Delta\mathbf{x}; \mathbf{K}) = \Delta\mathbf{y} \}, \end{aligned} \quad (3.1)$$

where \mathbf{X} and \mathbf{K} are independent and uniformly distributed over $\{0, 1\}^N$ and the space of all subkeys, respectively.

According to Definition 3.1.1, a block cipher is a Markov cipher if, for each round, the probability (over the uniform distribution of subkeys) that a pair of round inputs with XOR difference $\Delta\mathbf{x}$ will produce a pair of outputs with XOR difference $\Delta\mathbf{y}$ is independent of the actual choice of inputs, but depends only on $\Delta\mathbf{x}$ and $\Delta\mathbf{y}$.

The terminology “Markov cipher” is based on the concept of a *Markov chain* from probability theory; the connection is made in Theorem 3.1.3 below.

Definition 3.1.2 ([70]). *A sequence of discrete random variables $\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2, \dots$ is a Markov chain if, for any $i \geq 1$ and any fixed values $\alpha_0, \alpha_1, \dots, \alpha_i$,*

$$\text{Prob}\{\mathbf{Z}_i = \alpha_i \mid \mathbf{Z}_{i-1} = \alpha_{i-1}, \dots, \mathbf{Z}_0 = \alpha_0\} = \text{Prob}\{\mathbf{Z}_i = \alpha_i \mid \mathbf{Z}_{i-1} = \alpha_{i-1}\}.$$

That is, \mathbf{Z}_i depends only on \mathbf{Z}_{i-1} , not on any of the other previous random variables. A Markov chain is called homogeneous if these conditional probabilities are independent of i , i.e., if, for any fixed values α, β and for all $i, j \geq 1$,

$$\text{Prob}\{\mathbf{Z}_i = \alpha \mid \mathbf{Z}_{i-1} = \beta\} = \text{Prob}\{\mathbf{Z}_j = \alpha \mid \mathbf{Z}_{j-1} = \beta\}.$$

Theorem 3.1.3 ([70]). *Given an R -round Markov cipher, consider the encryption of pairs of plaintexts. Define the random variables $\Delta\mathbf{Y}_0, \Delta\mathbf{Y}_1, \dots, \Delta\mathbf{Y}_R$ as follows: $\Delta\mathbf{Y}_0$ represents the XOR of the current plaintext pair, and $\Delta\mathbf{Y}_r$ represents the XOR of the corresponding pair of outputs from round r , for $1 \leq r \leq R$. If the plaintexts and all subkeys are chosen independently and uniformly from their respective domains, then $\Delta\mathbf{Y}_0, \Delta\mathbf{Y}_1, \dots, \Delta\mathbf{Y}_R$ is a homogeneous Markov chain.*

It is easy to show that the SPN structure we are using is a Markov cipher (see Lemma 3.3.1), as are certain Feistel networks, such as DES [70]. Markov ciphers provide a general context in which to define and discuss cipher properties, and facilitate the analysis of a number of attacks.

Remark 3.1.4. The Markov cipher property was originally formulated in the examination of differential cryptanalysis [70], which is based directly on the idea of XOR differences. However, it is also relevant for linear cryptanalysis—the connection is via the relationship between linear probability and differential probability given in Theorem 2.6.5.

3.2 Linear Cryptanalysis of Markov Ciphers

Note that we continue to assume the use of independent keys (Section 2.4.1).

Matsui introduced two versions of linear cryptanalysis [74]. The first version, called Algorithm 1, extracts only a single bit of key information. The second version, Algorithm 2, can be used to extract one or both of the outermost subkeys. We will focus on Algorithm 2, and assume that it is being used to obtain the first subkey, \mathbf{k}^1 . (We demonstrate the use of linear cryptanalysis to attack both outermost subkeys simultaneously in Chapter 7, where we use linear cryptanalysis to break the Q cipher.) Once \mathbf{k}^1 is known, round 1 can be stripped off, and linear cryptanalysis can be reapplied to obtain \mathbf{k}^2 , and so on, until all subkeys are known.

The attacker views rounds $2 \dots R$ as a single key-dependent function mapping $\{0, 1\}^N \rightarrow \{0, 1\}^N$, where $(\tilde{\mathbf{k}} = \langle \mathbf{k}^2, \mathbf{k}^3, \dots, \mathbf{k}^R \rangle)$ is the key being used. Ideally, the attacker wants to precompute masks $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ that maximize $LP(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}})$ for this function. The number of known $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs required for a successful attack (the data complexity), denoted \mathcal{N}_L , is then determined from $LP(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}})$. Given an assumption about the behavior of round-1 output [74], Matsui

shows that if

$$\mathcal{N}_L = \frac{c}{LP(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}})}, \quad (3.2)$$

then Algorithm 2 has the success rates in Table 3.1, for various values of the constant c .

c	8	16	32	64
Success rate	48.6%	78.5%	96.7%	99.9%

Table 3.1: Success rates for linear cryptanalysis (Algorithm 2)

Remark 3.2.1. Note that Table 3.1 is the same as Table 3 given by Matsui in [74], except that the constant values in Table 3.1 are larger by a factor of 4, since Matsui uses *bias* values, not LP values.

Given the attacker's choice of \mathbf{a} and \mathbf{b} , linear cryptanalysis follows the steps outlined in Figure 3.1. The attacker first obtains \mathcal{N}_L \langle plaintext, ciphertext \rangle pairs,

$$\langle \mathbf{p}_1, \mathbf{c}_1 \rangle, \langle \mathbf{p}_2, \mathbf{c}_2 \rangle, \dots, \langle \mathbf{p}_{\mathcal{N}_L}, \mathbf{c}_{\mathcal{N}_L} \rangle.$$

The method by which these pairs are acquired is mostly outside the scope of this thesis, but two possibilities are worth mentioning. First, many files have standard header information, so eavesdropping on the ciphertexts produced by the encryption of such files will yield a number of known \langle plaintext, ciphertext \rangle pairs. Second, the attacker may be able to submit transactions to an entity (e.g., a server) that performs encryption as part of its processing; depending on the nature of the transactions, this may yield *chosen* \langle plaintext, ciphertext \rangle pairs.

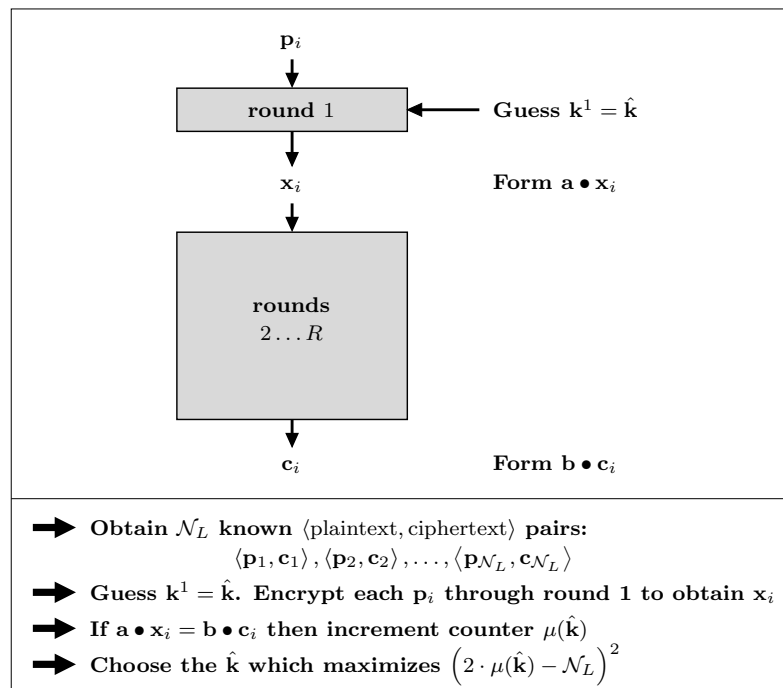


Figure 3.1: Summary of linear cryptanalysis (Algorithm 2)

The attacker then tries all possible values for the round-1 subkey, \mathbf{k}^1 —each such value is called a *guess*. Denote the current guess by $\hat{\mathbf{k}}$, and let $\mu(\hat{\mathbf{k}})$ denote a counter associated with $\hat{\mathbf{k}}$ (initially, $\mu(\hat{\mathbf{k}}) = 0$). The attacker proceeds as if $\hat{\mathbf{k}}$ is the correct value of \mathbf{k}^1 ; under this assumption, round 1 is completely known, so each plaintext \mathbf{p}_i ($1 \leq i \leq \mathcal{N}_L$) is encrypted through round 1 to obtain the corresponding intermediate value \mathbf{x}_i . There are now \mathcal{N}_L input/output pairs for the key-dependent function consisting of rounds $2 \dots R$, namely

$$\langle \mathbf{x}_1, \mathbf{c}_1 \rangle, \langle \mathbf{x}_2, \mathbf{c}_2 \rangle, \dots, \langle \mathbf{x}_{\mathcal{N}_L}, \mathbf{c}_{\mathcal{N}_L} \rangle.$$

For each pair $\langle \mathbf{x}_i, \mathbf{c}_i \rangle$, the attacker computes $\mathbf{a} \bullet \mathbf{x}_i$ and $\mathbf{b} \bullet \mathbf{c}_i$ and increments $\mu(\hat{\mathbf{k}})$ if $\mathbf{a} \bullet \mathbf{x}_i = \mathbf{b} \bullet \mathbf{c}_i$. Once this process has been carried out for all possible guesses, the

guess that maximizes $\left(2 \cdot \mu(\hat{\mathbf{k}}) - \mathcal{N}_L\right)^2$ is taken to be the correct value of \mathbf{k}^1 .

The theory behind linear cryptanalysis [74] states that if $\hat{\mathbf{k}}$ is an incorrect guess of the round-1 subkey, then \mathbf{x}_i will be an incorrect (and random) guess of the input to round 2, so the equation $\mathbf{a} \bullet \mathbf{x}_i = \mathbf{b} \bullet \mathbf{c}_i$ will hold with probability approximately $\frac{1}{2}$, and therefore, on average, $\mu(\hat{\mathbf{k}})$ will be close to $\frac{\mathcal{N}_L}{2}$ and $\left(2 \cdot \mu(\hat{\mathbf{k}}) - \mathcal{N}_L\right)^2$ will be close to 0. On the other hand, if $\hat{\mathbf{k}} = \mathbf{k}^1$, then, on average, $\mu(\hat{\mathbf{k}})$ will deviate significantly from $\frac{\mathcal{N}_L}{2}$ and $\left(2 \cdot \mu(\hat{\mathbf{k}}) - \mathcal{N}_L\right)^2$ will be significantly larger than 0—in this case, $\hat{\mathbf{k}}$ will maximize $\left(2 \cdot \mu(\hat{\mathbf{k}}) - \mathcal{N}_L\right)^2$ with probability as given in Table 3.1.

Reducing the Number of Counters

In the above, Algorithm 2 has been used to derive \mathbf{k}^1 in its entirety. Since the space of all subkeys may be large, it is usually not realistic to maintain a counter for each possible value of \mathbf{k}^1 . Instead, the following modified approach is typically used to derive a subset of the bits of \mathbf{k}^1 while maintaining a practical number of counters [74]. The ability to employ this modified approach depends on the structure of round 1 (or, in general, on the structure of the round whose subkey is being attacked).

Since computation of $\mathbf{a} \bullet \mathbf{x}_i$ only requires knowledge of the bits of \mathbf{x}_i for which the corresponding bits of \mathbf{a} are 1, it may suffice to guess a (strict) subset of the bits of \mathbf{k}^1 —these are called *effective key bits* [76]—and then to use these bits to perform a partial encryption through round 1 in order to obtain the desired bits of \mathbf{x}_i . If this is the case, then the attacker only needs to maintain a counter for each guess of the effective key bits.

Once the effective bits of \mathbf{k}^1 are known, various techniques can be used to finish breaking the cipher. In attacking DES, Matsui was able to determine 26 subkey bits

by the above method, and since these corresponded to 26 bits of the original 56-bit key (due to the simplicity of DES’s key-scheduling algorithm), the remaining 30 key bits could be determined by exhaustive search [76]. (We further discuss Matsui’s attack on DES in Section 3.4.1.) Another technique is to reapply linear cryptanalysis using new values for \mathbf{a} and \mathbf{b} in order to derive other subkey bits, until all the desired key bits are known or until exhaustive search becomes feasible. We employ this latter approach in our break of the Q cipher, as described in Chapter 7.

Notational Conventions

Above, we have discussed input and output masks and the associated LP values for rounds $2 \dots R$ of an R -round cipher. It is useful to consider these and other related concepts as applying to any $T \geq 2$ consecutive “core” rounds—we say that these are the rounds being *approximated*. Hereafter, unless specified otherwise, terms such as “first round” and “last round” will be relative to the T rounds under consideration. For Algorithm 2 as given above, $T = R - 1$, and the “first round,” or “round 1,” is actually the second round of the cipher.

We will use single-variable superscripts in our notation to refer to individual rounds, so $LP^t(\mathbf{a}, \mathbf{b}; \mathbf{k}^t)$ and $ELP^t(\mathbf{a}, \mathbf{b})$ denote LP and ELP values, respectively, for round t . We will use superscripts of the form $[x \dots y]$ to refer to a sequence of consecutive rounds being considered as a single unit. For example, $ELP^{[1 \dots t]}(\mathbf{a}, \mathbf{b})$ is an ELP value over rounds $1 \dots t$.

3.2.1 The Use of Expected Linear Probability

Let $\tilde{\mathbf{k}}$ denote the vector of subkeys used in the T core rounds under consideration. In terms of linear cryptanalysis, both the attacker and the cipher designer are interested in the following value:

$$\max_{\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}} LP^{[1\dots T]}(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}}). \quad (3.3)$$

The designer wants to prove that the value in (3.3) is sufficiently small that the corresponding data complexity (see (3.2)) is prohibitively large; the attacker, of course, wants the opposite, i.e., wants the value in (3.3) to be relatively large.

Direct computation of (3.3) is generally infeasible for two reasons: first, for all masks $\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}$ it requires encrypting all N -bit vectors through rounds $1 \dots T$ (see Definition 2.6.2), which is prohibitive for typical values of N (e.g., $N = 64$, $N = 128$); and second, it depends on an unknown key. Researchers have dealt with the latter problem by working instead with the expected value $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$, and making the assumption that for almost all values of $\tilde{\mathbf{k}}$,

$$LP^{[1\dots T]}(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}}) \approx ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}). \quad (3.4)$$

Harpes et al. call this the *Hypothesis of Fixed-Key Equivalence*, and present an argument for its effectiveness [36]. The data complexity of Algorithm 2 in (3.2) is now taken to be

$$\mathcal{N}_L = \frac{c}{ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})}. \quad (3.5)$$

For the purpose of this thesis, we will adopt the assumption in (3.4). In Section 8.2 a closer examination of (3.4) is listed as a possible direction for future research.

Given (3.4), the first problem listed above now becomes the computational infeasibility of computing

$$\max_{\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}} ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}). \quad (3.6)$$

Although (3.6) has the advantage that it does not depend on an unknown value, its direct computation is even more prohibitive than that of (3.3), since a single term $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ is computed over all N -bit inputs to rounds $1 \dots T$ and over all independent keys for rounds $1 \dots T$. Researchers have traditionally solved this complexity problem through the use of *linear characteristics* (or simply *characteristics*), which allow us to reduce this approximation of rounds $1 \dots T$ to a series of T one-round approximations, each of which is feasible to compute.

3.2.2 Linear Characteristics

Definition 3.2.2. A one-round characteristic for round t is a pair of N -bit masks, $\langle \mathbf{a}^t, \mathbf{b}^t \rangle$; we view \mathbf{a}^t and \mathbf{b}^t as input and output masks, respectively, for round t .

Definition 3.2.3. A T -round characteristic for rounds $1 \dots T$ is a $(T + 1)$ -tuple of N -bit masks, $\Omega = \langle \mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^T, \mathbf{a}^{T+1} \rangle$; we view \mathbf{a}^t and \mathbf{a}^{t+1} as input and output masks, respectively, for round t ($1 \leq t \leq T$).

Definition 3.2.4. Let $\Omega = \langle \mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^T, \mathbf{a}^{T+1} \rangle$ be a T -round characteristic for rounds $1 \dots T$, and let $\tilde{\mathbf{k}} = \langle \mathbf{k}^1, \mathbf{k}^2, \dots, \mathbf{k}^T \rangle$ be the vector of subkeys being used for

these rounds. The linear characteristic probability (*LCP*) and expected linear characteristic probability (*ELCP*) of Ω are defined as

$$\begin{aligned} LCP(\Omega; \tilde{\mathbf{k}}) &\stackrel{\text{def}}{=} \prod_{t=1}^T LP^t(\mathbf{a}^t, \mathbf{a}^{t+1}; \mathbf{k}^t) \\ ELCP(\Omega) &\stackrel{\text{def}}{=} \prod_{t=1}^T ELP^t(\mathbf{a}^t, \mathbf{a}^{t+1}). \end{aligned} \quad (3.7)$$

Feasibility of Computing $ELCP(\Omega)$

Let Ω be a T -round characteristic. Since a single cipher round usually exhibits a simple structure, often there is a shortcut for computing the values $ELP^t(\mathbf{a}^t, \mathbf{a}^{t+1})$ (this is true for SPNs, as we show in Section 3.3). Then $ELCP(\Omega)$ is obtained by the simple product in (3.7).

Choosing the Best Characteristic (Practical Security)

We now consider how characteristics are used to handle the computational complexity problem described at the end of Section 3.2.1. In carrying out linear cryptanalysis, the attacker typically runs a straightforward search algorithm to find the T -round characteristic, $\hat{\Omega}$, for which $ELCP(\hat{\Omega})$ is *maximal*; such a characteristic (not necessarily unique) is called the *best characteristic* [75]. If $\hat{\Omega} = \langle \mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^T, \mathbf{a}^{T+1} \rangle$, and if the input and output masks used in Algorithm 2 are taken to be $\mathbf{a} = \mathbf{a}^1$ and $\mathbf{b} = \mathbf{a}^{T+1}$, respectively, then the value $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$, which is used to determine the data complexity in (3.5), is approximated by

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \approx ELCP(\hat{\Omega}). \quad (3.8)$$

The approximation in (3.8) has been widely used to evaluate the security of block ciphers against linear cryptanalysis [39, 51, 75, 114]. Knudsen calls a block cipher

practically secure if the data complexity determined by this method is prohibitive [65]. For certain ciphers, the approximation in (3.8) is very good—this happens to be the case for DES [46]. However, by introducing the concept of *linear hulls*, Nyberg [90] showed that the approximation in (3.8) can result in an overestimation of the data complexity required for a given success rate—clearly this is advantageous for an attacker, but problematic for a cipher designer.

3.2.3 Linear Hulls

The following definition and theorem are due to Nyberg [90].

Definition 3.2.5. *Given N -bit masks \mathbf{a}, \mathbf{b} , the corresponding linear hull, denoted $\text{ALH}(\mathbf{a}, \mathbf{b})$,¹ is the set of all T -round characteristics (for the T rounds under consideration) having \mathbf{a} as the input mask for round 1 and \mathbf{b} as the output mask for round T , i.e., all characteristics of the form*

$$\Omega = \langle \mathbf{a}, \mathbf{a}^2, \mathbf{a}^3, \dots, \mathbf{a}^T, \mathbf{b} \rangle.$$

Theorem 3.2.6. *Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N$. Then*

$$\text{ELP}^{[1\dots T]}(\mathbf{a}, \mathbf{b}) = \sum_{\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})} \text{ELCP}(\Omega).$$

It follows immediately from Theorem 3.2.6 that the approximation in (3.8) does not hold in general, since $\text{ELP}^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ is seen to be equal to a sum of terms $\text{ELCP}(\Omega)$ over a (large) set of characteristics, and therefore, in general, the ELCP of any characteristic will be strictly *less than* the corresponding ELP value, resulting in an overestimation of the data complexity for a given success rate. This is referred to as the *linear hull effect*.

¹Nyberg originally used the term approximate linear hull, hence the abbreviation ALH.

Remark 3.2.7. The linear hull effect is significant for the AES, since the ELCP of any characteristic over $T = 8$ rounds is upper bounded by 2^{-300} [25], but the largest nontrivial ELP value has 2^{-128} as a lower bound.²

The next lemma follows easily from Definition 3.2.4 and Theorem 3.2.6.

Lemma 3.2.8. *Let $T \geq 2$, and let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N$. Then*

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{x} \in \{0, 1\}^N} ELP^{[1\dots(T-1)]}(\mathbf{a}, \mathbf{x}) \cdot ELP^T(\mathbf{x}, \mathbf{b}).$$

Maximum Average Linear Hull Probability (Provable Security)

Although linear characteristics have the advantage of computational tractability, because of the linear hull effect the ELCP of the best characteristic should be viewed only as a first approximation to the value we want to compute, namely the value in (3.6), which we repeat here:

$$\max_{\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}} ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}).$$

This value is called the *maximum average linear hull probability* (MALHP).³ A block cipher is considered to be *provably secure* [90] against linear cryptanalysis if the MALHP is sufficiently small that the resulting data complexity is prohibitive. For Algorithm 2 as described in Section 3.2, this must be true for $T = R - 1$. Since variations of linear cryptanalysis can be used to attack the first and last subkeys simultaneously, it may also be important that the data complexity remain prohibitive for $T = R - 2$.

²This follows by observing that Lemma 2.6.3 is contradicted if the maximum ELP value is less than 2^{-d} .

³It would be more consistent with our current terminology to use the name *maximum expected linear probability* (MELP), but we retain MALHP because of its use in existing literature, including works by the author.

Because of the computational difficulty of evaluating the MALHP, researchers have adopted the approach of upper bounding this value [5, 41, 55, 96, 97, 101]. If such an upper bound is sufficiently small, provable security can be claimed. Deriving upper bounds on the MALHP has been an important focus of the author's work [58, 60], and is the topic of Chapter 6.

Remark 3.2.9. The term *provable security* is well established in the context of linear cryptanalysis of block ciphers. However, to avoid confusion, it should be noted that this term has another common usage: a cryptographic method is called provably secure if the difficulty of attacking the method has been proven to be at least as great as the difficulty of solving some well-known problem that is believed to be hard (such as integer factorization). In this alternate terminology, our definition of provable security is an example of *computational security* [81].

3.3 SPN-Specific Considerations

We now adapt certain results related to linear cryptanalysis to the SPN structure. Note that where matrix multiplication is involved, we view all vectors as *column vectors*. Also, if \mathcal{M} is a matrix, \mathcal{M}' denotes the transpose of \mathcal{M} . We start with the following foundational lemma.

Lemma 3.3.1. *The SPN structure as given in Section 2.4.2 is a Markov cipher.*

Proof. This follows easily from the fact that key mixing in an SPN involves the XOR of an N -bit subkey at the beginning of a round. Let $1 \leq r \leq R$, and consider (3.1) in Definition 3.1.1. For round input $\mathbf{x} \in \{0, 1\}^N$ and subkey $\mathbf{k}^r \in \{0, 1\}^N$, the round function is $\epsilon_r(\mathbf{x} \oplus \mathbf{k}^r)$. Let $\mathbf{x}, \Delta\mathbf{x}, \Delta\mathbf{y} \in \{0, 1\}^N$ be fixed. The left-hand side of (3.1)

now becomes

$$\text{Prob}_{\mathbf{K}} \{ \epsilon_r(\mathbf{x} \oplus \mathbf{K}) \oplus \epsilon_r(\mathbf{x} \oplus \Delta \mathbf{x} \oplus \mathbf{K}) = \Delta \mathbf{y} \}. \quad (3.9)$$

If we replace \mathbf{K} with the random variable $\mathbf{X} = \mathbf{x} \oplus \mathbf{K}$, we get

$$\text{Prob}_{\mathbf{X}} \{ \epsilon_r(\mathbf{X}) \oplus \epsilon_r(\mathbf{X} \oplus \Delta \mathbf{x}) = \Delta \mathbf{y} \},$$

which yields the same value as (3.9). The result follows.

A final technical detail: if the last subkey (\mathbf{k}^{R+1}) is considered to be part of round R , then the above argument continues to hold for round R . \square

In the rest of this section, we again consider values over T core rounds.

Lemma 3.3.2. *Let $1 \leq t \leq T$, and let $\mathbf{a}, \mathbf{b}, \mathbf{k}^t \in \{0, 1\}^N$. Then $LP^t(\mathbf{a}, \mathbf{b}; \mathbf{k}^t)$ is independent of \mathbf{k}^t , and therefore*

$$LP^t(\mathbf{a}, \mathbf{b}; \mathbf{k}^t) = ELP^t(\mathbf{a}, \mathbf{b}).$$

Proof. Let $\mathbf{X} \in \{0, 1\}^N$ and $\hat{\mathbf{X}} = \mathbf{X} \oplus \mathbf{k}^t$ be random variables, and denote round t by $\epsilon_t(\cdot)$. It suffices to show that $LP^t(\mathbf{a}, \mathbf{b}; \mathbf{k}^t) = LP^t(\mathbf{a}, \mathbf{b}; \mathbf{0})$ for all $\mathbf{k}^t \in \{0, 1\}^N$. From Definition 2.6.2 we have

$$\begin{aligned} LP^t(\mathbf{a}, \mathbf{b}; \mathbf{k}^t) &= (2 \cdot \text{Prob}_{\mathbf{X}} \{ \mathbf{a} \bullet \mathbf{X} = \mathbf{b} \bullet \epsilon_t(\mathbf{X} \oplus \mathbf{k}^t) \} - 1)^2 \\ &= (2 \cdot \text{Prob}_{\hat{\mathbf{X}}} \{ \mathbf{a} \bullet (\hat{\mathbf{X}} \oplus \mathbf{k}^t) = \mathbf{b} \bullet \epsilon_t(\hat{\mathbf{X}}) \} - 1)^2 \\ &= (2 \cdot \text{Prob}_{\hat{\mathbf{X}}} \{ (\mathbf{a} \bullet \hat{\mathbf{X}}) \oplus (\mathbf{a} \bullet \mathbf{k}^t) = \mathbf{b} \bullet \epsilon_t(\hat{\mathbf{X}}) \} - 1)^2. \end{aligned} \quad (3.10)$$

If $\mathbf{a} \bullet \mathbf{k}^t = 0$, then the expression in (3.10) is equal to $LP^t(\mathbf{a}, \mathbf{b}; \mathbf{0})$. If $\mathbf{a} \bullet \mathbf{k}^t = 1$, then

$$\text{Prob}_{\hat{\mathbf{X}}} \{ (\mathbf{a} \bullet \hat{\mathbf{X}}) \oplus (\mathbf{a} \bullet \mathbf{k}^t) = \mathbf{b} \bullet \epsilon_t(\hat{\mathbf{X}}) \} = 1 - \text{Prob}_{\hat{\mathbf{X}}} \{ \mathbf{a} \bullet \hat{\mathbf{X}} = \mathbf{b} \bullet \epsilon_t(\hat{\mathbf{X}}) \},$$

so the expression in (3.10) again reduces to $LP^t(\mathbf{a}, \mathbf{b}; \mathbf{0})$. \square

Corollary 3.3.3. *Let Ω be a T -round characteristic, and let $\hat{\mathbf{k}}$ be the vector of subkeys for the T rounds under consideration. Then $LCP(\Omega; \hat{\mathbf{k}}) = ELCP(\Omega)$.*

Proof. Follows from Definition 3.2.4 and Lemma 3.3.2. \square

As in Section 2.4.2, let \mathcal{L} denote the SPN linear transformation represented as an invertible $N \times N$ binary matrix, i.e., if $\mathbf{x}, \mathbf{y} \in \{0, 1\}^N$ are the input and output, respectively, for the linear transformation, then $\mathbf{y} = \mathcal{L}\mathbf{x}$.

Lemma 3.3.4 ([25]). *If $\mathbf{a} \in \{0, 1\}^N$ is a mask applied to the inputs of the linear transformation, then there is a unique corresponding mask $\mathbf{b} \in \{0, 1\}^N$ applied to the outputs, i.e., there is a mask \mathbf{b} such that for all $\mathbf{x} \in \{0, 1\}^N$,*

$$\mathbf{a} \bullet \mathbf{x} = \mathbf{b} \bullet (\mathcal{L}\mathbf{x}).$$

The relationship between the masks \mathbf{a} and \mathbf{b} is given by $\mathbf{a} = \mathcal{L}'\mathbf{b}$.

It follows from Lemma 3.3.4 that if \mathbf{a}^t and \mathbf{a}^{t+1} are input and output masks for round t , respectively, then the resulting input and output masks for the *substitution stage* of round t are \mathbf{a}^t and $\mathbf{b}^t = \mathcal{L}'\mathbf{a}^{t+1}$. Further, \mathbf{a}^t and \mathbf{b}^t can be naturally partitioned to determine input and output masks for each s-box in round t . Number the s-boxes from left to right as $S_1^t, S_2^t, \dots, S_M^t$, and let the masks for S_i^t be denoted \mathbf{a}_i^t and \mathbf{b}_i^t . Then from Matsui's Piling-up Lemma [74] and Lemma 3.3.2,

$$ELP^t(\mathbf{a}^t, \mathbf{a}^{t+1}) = \prod_{i=1}^M LP^{S_i^t}(\mathbf{a}_i^t, \mathbf{b}_i^t). \quad (3.11)$$

From the above, any characteristic $\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})$ determines an input and an output mask for each s-box in rounds $1 \dots T$. If there is at least one s-box for which the input mask is zero and the output mask is nonzero, or vice versa, the linear probability

associated with that s-box will trivially be 0, and therefore $ELCP(\Omega) = 0$ by (3.11) and (3.7). We exclude such characteristics from consideration via the following two definitions.

Definition 3.3.5 ([114]). *Let Ω be a T -round characteristic for rounds $1 \dots T$. Then Ω is called consistent if, for each s-box in rounds $1 \dots T$, the input and output masks determined by Ω for that s-box are either both zero or both nonzero.*

Definition 3.3.6. *For $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N$, let $ALH(\mathbf{a}, \mathbf{b})^*$ consist of all the consistent characteristics in $ALH(\mathbf{a}, \mathbf{b})$.*

Definition 3.3.7 ([10]). *Given a characteristic $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, any s-box for which the resulting input and output masks are nonzero is called active.*

Definition 3.3.8. *Given $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, let $A(\Omega)$ denote the number of s-boxes made active by Ω .*

Definition 3.3.9. *Given $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, let \mathbf{v} be the input or output mask for the substitution stage of round t . Then the active s-boxes in round t can be determined from \mathbf{v} (without knowing the corresponding output/input mask). We define $\gamma_{\mathbf{v}}$ to be the M -bit vector that encodes this pattern of active s-boxes: $\gamma_{\mathbf{v}} = \gamma_1 \gamma_2 \dots \gamma_M$, where $\gamma_i = 1$ if the i^{th} s-box is active, and $\gamma_i = 0$ otherwise, for $1 \leq i \leq M$.*

Definition 3.3.10. *Let $\gamma, \hat{\gamma} \in \{0, 1\}^M$. Then*

$$W[\gamma, \hat{\gamma}] \stackrel{\text{def}}{=} \# \{ \mathbf{y} \in \{0, 1\}^N : \gamma_{\mathbf{x}} = \gamma, \gamma_{\mathbf{y}} = \hat{\gamma}, \text{ where } \mathbf{x} = \mathcal{L}'\mathbf{y} \}.$$

Informally, the value $W[\gamma, \hat{\gamma}]$ represents the number of ways the linear transformation can “connect” a pattern of active s-boxes in one round (γ) to a pattern of

active s-boxes in the next round ($\hat{\gamma}$). It is easy to see that $W[\mathbf{0}, \mathbf{0}] = 1$, and if $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, then $W[\gamma, \mathbf{0}] = W[\mathbf{0}, \hat{\gamma}] = 0$.

The diffusive power of a linear transformation is its ability to force some minimum number of s-boxes to be active over a sequence of rounds. We quantify this in the following definition.

Definition 3.3.11 ([25]). *The linear branch number, \mathcal{B}_l , of an SPN linear transformation is the minimum number of active s-boxes in two consecutive rounds for any $\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*$, i.e.,*

$$\begin{aligned} \mathcal{B}_l &\stackrel{\text{def}}{=} \min \{ wt(\gamma_{\mathbf{x}}) + wt(\gamma_{\mathbf{y}}) : \mathbf{y} \in \{0, 1\}^N \setminus \mathbf{0}, \mathbf{x} = \mathcal{L}'\mathbf{y} \} \\ &= \min \{ wt(\gamma) + wt(\hat{\gamma}) : \gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}, W[\gamma, \hat{\gamma}] > 0 \}. \end{aligned}$$

Remark 3.3.12. It is trivial to show that $2 \leq \mathcal{B}_l \leq (M + 1)$. If $\mathcal{B}_l = (M + 1)$, then the linear transformation is called *maximally diffusive* [41].

Definition 3.3.13. *Let q be the maximum nontrivial LP value over all SPN s-boxes. Symbolically,*

$$q \stackrel{\text{def}}{=} \max_{S \in \text{SPN}} \max_{\alpha, \beta \in \{0, 1\}^n \setminus \mathbf{0}} LP^S(\alpha, \beta).$$

3.4 Extensions of Linear Cryptanalysis

We now consider extensions to the basic linear cryptanalytic attack described in Section 3.2.

3.4.1 Key Ranking

Key ranking is a technique that was used by Matsui in applying linear cryptanalysis to DES [74]. In our exposition of linear cryptanalysis in Section 3.2, the attacker adopts the subkey guess (or, more commonly, the guess of the effective bits of the subkey), denoted $\hat{\mathbf{k}}$, that maximizes the statistic $\left(2 \cdot \mu(\hat{\mathbf{k}}) - \mathcal{N}_L\right)^2$. Key ranking extends this by first sorting the guesses based on nonincreasing values of this statistic—denote these sorted guesses by $\hat{\mathbf{k}}_1, \hat{\mathbf{k}}_2, \hat{\mathbf{k}}_3, \dots$. Then for each $\hat{\mathbf{k}}_i$, beginning with $\hat{\mathbf{k}}_1$, an exhaustive search is performed on the remaining key bits needed to break the cipher. For each value of the overall key determined by this method, a trial encryption is performed with a known ⟨plaintext, ciphertext⟩ pair, (\mathbf{p}, \mathbf{c}) . If the encryption of \mathbf{p} is equal to \mathbf{c} , the correct key has been found (it may be judicious to encrypt a few additional pairs to verify this conclusion). If the exhaustive search fails for $\hat{\mathbf{k}}_i$, then the process is repeated for $\hat{\mathbf{k}}_{i+1}$. The basic idea is that since the most likely guess may not be correct, the guesses should be tried in decreasing order of likelihood.

In his attack on DES, Matsui had the advantage that guessing 26 subkey bits immediately yielded 26 bits of the original 56-bit key, so exhaustive search of the remaining 30 key bits was feasible. However, if the number of remaining key bits is large, exhaustive search will not be possible. In this case, the attacker can assume that $\hat{\mathbf{k}}_1$ is the correct guess, and proceed to attack the rest of the cipher (e.g., by reapplication of linear cryptanalysis to determine the next subkey, etc.). If the guess of $\hat{\mathbf{k}}_1$ is wrong, it may be necessary to backtrack at some point.

Junod [46] and Junod and Vaudenay [47] have examined the theoretical underpinnings of key ranking, noting that key ranking for linear cryptanalysis is a particular example of Vaudenay’s more general *statistical cryptanalysis* concept [113].

3.4.2 Multiple Linear Approximations

Kaliski and Robshaw investigate the use of multiple linear approximations to improve the success rate of linear cryptanalysis and/or to reduce the data complexity [48, 49]. The basic idea is to use multiple pairs of $\langle \text{input}, \text{output} \rangle$ masks for the T rounds being approximated, $\langle \mathbf{a}_1, \mathbf{b}_1 \rangle, \langle \mathbf{a}_2, \mathbf{b}_2 \rangle, \langle \mathbf{a}_3, \mathbf{b}_3 \rangle, \dots$, such that each pair $\langle \mathbf{a}_i, \mathbf{b}_i \rangle$ attacks the *same* effective key bits. For a given data complexity, each pair of masks suggests the most likely guess of the effective key bits, or, if key ranking is employed, an ordering of the possible guesses. This information can be combined in such a way that the most likely guess has a significantly higher probability of being correct, or, for a fixed success rate, such that a lower data complexity is required.

The existing literature on multiple linear approximations is based on the use of linear characteristics. Further work is required to adapt this to an approach based on linear hulls. This is listed in Section 8.2 as a topic for future investigation.

3.4.3 Generalized Form of Linear Cryptanalysis

Harpes et al. present a generalized form of linear cryptanalysis that is more effective than the original linear cryptanalysis for certain ciphers [36]. The basic observation is that for a fixed mask $\mathbf{a} \in \{0, 1\}^N \setminus \mathbf{0}$, the inner product $\mathbf{a} \bullet \mathbf{x}$ defines a function mapping $\{0, 1\}^N \rightarrow \{0, 1\}$ that is 0 for exactly half the inputs—such a function is called *balanced* [80]. However, there are many balanced functions not defined by masks. Harpes et al. propose replacing the input and output masks for T core rounds with two general balanced binary functions, say f and g , for which the following value is relatively large:

$$|2 \cdot \text{Prob}_{\mathbf{X}} \{f(\mathbf{X}) = g(\mathbf{Y})\} - 1| \quad (3.12)$$

(here $\mathbf{X}, \mathbf{Y} \in \{0, 1\}^N$ are random variables representing the input and corresponding output for rounds $1 \dots T$). Harpes et al. refer to the value in (3.12) as *imbalance*—when f and g are determined by masks, the imbalance is the positive square root of the associated linear probability. Having found such f and g , the resulting attack closely resembles the original linear cryptanalysis.

For ciphers in which key mixing is via the XOR operation, as for SPNs, Harpes et al. note that their generalized approach essentially reduces to the original linear cryptanalysis. However, they construct a cipher in [36] that is secure against the original linear cryptanalysis, but vulnerable to their generalized attack. It follows that generalized linear cryptanalysis needs to be considered for certain cipher structures.

Chapter 4

Expected Linear Probability

Values for SPNs with Randomly

Selected S-Boxes

As stated in Section 2.4.5, the true random cipher is generally taken to be the ideal block cipher model. One important direction of research involves demonstrating that certain block cipher properties converge to the corresponding properties for the true random cipher as the number of cipher rounds is increased. In this chapter we show this convergence for the SPN structure when the values under consideration are expected linear probability (ELP) values. The results in this chapter were published in [62].

4.1 Approximating the True Random Cipher

A number of researchers have analyzed the relationship between SPNs and the true random cipher. Chen and Tavares [21] give statistical evidence that the distribution of entries in the XOR table of an SPN with a fixed key approaches the corresponding distribution for the true random cipher with an increasing number of rounds.

Heys and Tavares consider the *avalanche* properties of SPNs [38], developing a model for the number of bit changes in the ciphertext given a one-bit change in the plaintext. Based on this model, the expected number of bit changes in the ciphertext after R rounds appears to converge to $\frac{N}{2}$ —the value for the true random cipher—as R increases. Convergence is fairly fast: for a 64-bit SPN with 8×8 s-boxes and the permutation of Kam and Davida [50], the value from the model almost exactly equals $\frac{N}{2}$ for $R \geq 5$. Youssef [116] extends the analysis in [38] by considering a variety of linear transformations; in all cases, the same convergence is observed.

Heys and Tavares incorporate certain assumptions into their model, most notably the use of an “idealized” s-box with a uniform XOR table (no such s-box exists¹). However, identical results can be obtained by viewing the problem from another perspective that does not require any assumptions: the values derived in [38] are equal to the expected number of bit changes in the ciphertext given a one-bit change in the plaintext, where the expectation is over all independent keys *and all choices of SPN s-boxes*, where each s-box is chosen independently and uniformly from the set of all bijective $n \times n$ s-boxes. We adopt the latter approach in this chapter and in Chapter 5. In addition, instead of treating differential properties of SPNs, we consider linear properties.

¹Since XOR table entries are *even*, and the entries in any row sum to 2^n , the “most uniform” XOR table contains 2^{n-1} 0’s and 2^{n-1} 2’s in any row indexed by a nonzero input difference.

4.2 SPNs with Randomly Selected S-Boxes

We continue to deal with linear approximations over $T \geq 2$ core SPN rounds. Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ be input and output masks, respectively, for these core rounds. Recall that M is the number of s-boxes per round.

We have two expressions for the value $ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b})$ for an SPN with fixed s-boxes, the first from Definition 2.6.2, and the second from Nyberg's linear hull theorem (Theorem 3.2.6). Both expressions appear to be difficult to compute exactly. However, by considering SPNs in which the s-boxes are selected independently and uniformly from the set of all $2^n!$ bijective $n \times n$ s-boxes, we are able to derive a formula for the *expected* ELP value, where the outer expectation is over all SPNs generated by this random selection of s-boxes. In other words, we are able to answer the following question: *If the SPN s-boxes are randomly selected, and an independent key is randomly selected, what is the expected value of $LP^{[1 \dots T]}(\mathbf{a}, \mathbf{b})$?* We then compute our formula for some SPNs with practical block sizes, and make the observation that the resulting values converge to the corresponding value for the true random cipher as T increases. This gives quantitative support to the claim that the SPN structure is a practical approximation to the true random cipher. We conjecture that this convergence holds for any SPN whose s-boxes and linear transformation satisfy some simple conditions.

The study of SPNs in which the s-boxes are randomly selected is relevant in light of the fact that several block ciphers with pseudorandomly generated (key-dependent) s-boxes have been proposed and analyzed, including Khufu [82], Blowfish [102], and Twofish [105]. Some researchers argue for the advantage of this approach, based on the fact that randomly selected s-boxes of sufficient size (e.g., 8×8) possess good

cryptographic properties with high probability (see [56]). Further, when the s-boxes are key dependent, it appears that many attacks that rely on known s-boxes cannot be applied [104]. On the other hand, there exists a nonzero probability that randomly selected s-boxes will produce an SPN that is weak according to some measure, but given the use of a good pseudorandom number generator, this probability is vanishingly small. Additional steps can also be taken to avoid this contingency, for example, s-boxes can be screened for certain properties before being accepted (mathematically, of course, this means that we deviate from the original model).

Definition 4.2.1. *Let \mathcal{SPN} denote the set of SPNs generated by selecting each s-box independently and uniformly from the set of all bijective $n \times n$ s-boxes.*

Remark 4.2.2. Clearly, all SPNs in \mathcal{SPN} are equally probable, i.e., \mathcal{SPN} has the uniform distribution.

Remark 4.2.3. Although Definition 4.2.1 deals with the entire SPN, for our purposes it suffices to restrict the random selection of s-boxes to the T core rounds under consideration.

4.2.1 Distribution of LP Values for Random S-Boxes

Lemma 4.2.4 ([94]). *Let S be a bijective $n \times n$ s-box ($n \geq 2$), and let masks $\alpha, \beta \in \{0, 1\}^n \setminus \mathbf{0}$ be fixed. If S varies uniformly over the set of all bijective $n \times n$ s-boxes, then the resulting distribution of values $LP^S(\alpha, \beta)$ is given by the following set of ordered pairs of the form $\langle LP, \text{probability} \rangle$:*

$$\left\{ \left\langle 0, \frac{\binom{2^{n-1}}{2^{n-2}}}{\binom{2^n}{2^{n-1}}} \right\rangle \right\} \cup \left\{ \left\langle \left(\frac{i}{2^{n-2}} \right)^2, \frac{2 \binom{2^{n-1}}{2^{n-2}+i}}{\binom{2^n}{2^{n-1}}} \right\rangle : 1 \leq i \leq 2^{n-2} \right\}.$$

The distribution given in Lemma 4.2.4 is plotted in Figure 4.1 for $n = 8$ (with a \log_{10} scale on the y -axis).

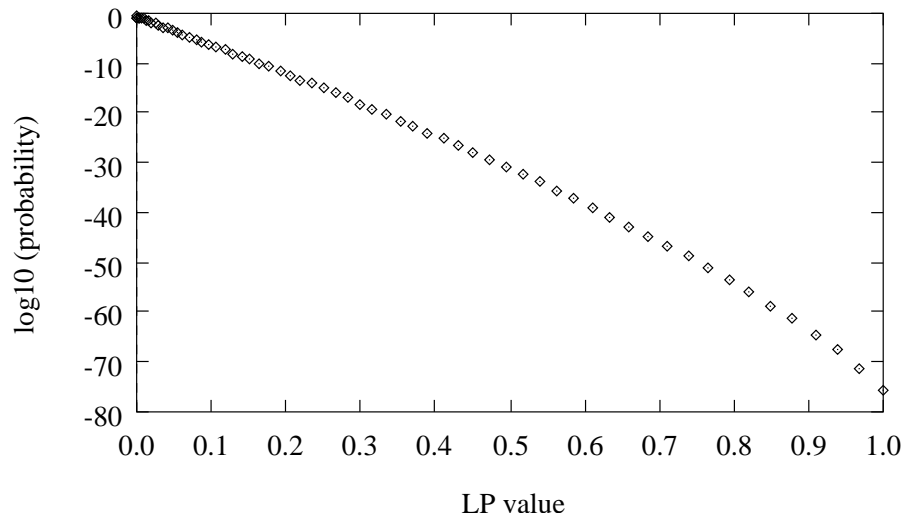


Figure 4.1: Distribution of LP values for random bijective 8×8 s-box

We now consider the expected LP value over all s-boxes for one active s-box, and for multiple active s-boxes (i.e., for a characteristic). We use the notation $A(\Omega)$ from Definition 3.3.8.

Lemma 4.2.5. *Let \mathbf{Z} be a random variable that has the distribution of Lemma 4.2.4.*

Then

$$E[\mathbf{Z}] = \frac{1}{(2^n - 1)} .$$

Proof. The distribution of Lemma 4.2.4 essentially consists of the squares of the terms in the *hypergeometric distribution* [63]. The result follows easily from the second moment of this distribution. \square

Lemma 4.2.6. *Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ be fixed, and let $\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*$. Let \mathbf{Z} be a random variable taking the value $ELCP(\Omega)$ for each SPN in \mathcal{SPN} . Then*

$$E[\mathbf{Z}] = \left(\frac{1}{2^n - 1} \right)^{A(\Omega)}.$$

Proof. Enumerate the s-boxes made active by Ω as $S_1, S_2, \dots, S_{A(\Omega)}$ (order is unimportant). It follows that Ω determines a pair of nonzero input and output masks for each S_a ($1 \leq a \leq A(\Omega)$). Let \mathbf{Z}_a be a random variable representing the LP value for S_a . (The s-boxes not in $\{S_1, S_2, \dots, S_{A(\Omega)}\}$ have zero input and output masks, and therefore LP values that are always equal to 1.) The distribution of \mathbf{Z}_a is given by Lemma 4.2.4 (as far as \mathbf{Z}_a is concerned, varying uniformly over \mathcal{SPN} is the same as varying uniformly over all choices of bijective $n \times n$ s-boxes for S_a), so from Lemma 4.2.5 we have

$$E[\mathbf{Z}_a] = \frac{1}{2^n - 1}.$$

Since we vary uniformly over \mathcal{SPN} , the \mathbf{Z}_a are independent. From (3.7) and (3.11),

$$\mathbf{Z} = \mathbf{Z}_1 \cdot \mathbf{Z}_2 \cdots \mathbf{Z}_{A(\Omega)},$$

and from the independence of the \mathbf{Z}_a ,

$$\begin{aligned} E[\mathbf{Z}] &= E[\mathbf{Z}_1] \cdot E[\mathbf{Z}_2] \cdots E[\mathbf{Z}_{A(\Omega)}] \\ &= \left(\frac{1}{2^n - 1} \right)^{A(\Omega)}. \end{aligned}$$

□

4.2.2 Expected ELP Values over all SPNs

In this section we state and prove the main result of this chapter (Theorem 4.2.9). Since we are dealing with an expectation over \mathcal{SPN} , we augment our notation with

subscripts when appropriate to indicate a dependence on the underlying SPN: specifically, \mathbf{spn} denotes a fixed value in \mathcal{SPN} and \mathbf{SPN} denotes a random variable over \mathcal{SPN} .

Convention

For the remainder of this chapter, whenever we are dealing with consecutive rounds $1 \dots t$, we adopt the convention that the linear transformation is omitted from round t . (If we then extend our consideration to rounds $1 \dots (t+1)$, the linear transformation is “put back” into round t but omitted from round $(t+1)$.) This convention has no cryptographic significance, but it has the advantage that given input and output masks, \mathbf{a}, \mathbf{b} , for rounds $1 \dots t$, we know that the pattern of active s-boxes in round t is given by $\gamma_{\mathbf{b}}$. Without this convention, we need to process \mathbf{b} backwards through the linear transformation (using Lemma 3.3.4) in order to obtain the pattern of active s-boxes in round t ; this would add unnecessary complexity to the development below.

Definition 4.2.7. For fixed masks $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ and positive integer A , let $C_{\mathbf{a}, \mathbf{b}}(A)$ denote the number of characteristics in $\text{ALH}(\mathbf{a}, \mathbf{b})^*$ that activate A s-boxes.

Lemma 4.2.8. If $\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*$, then $A_{\min} \leq A(\Omega) \leq A_{\max}$, where

$$\begin{aligned} A_{\min} &= wt(\gamma_{\mathbf{a}}) + wt(\gamma_{\mathbf{b}}) + (T - 2) \\ A_{\max} &= wt(\gamma_{\mathbf{a}}) + wt(\gamma_{\mathbf{b}}) + M(T - 2). \end{aligned}$$

Proof. Clearly the numbers of s-boxes made active in round 1 and round T are $wt(\gamma_{\mathbf{a}})$ and $wt(\gamma_{\mathbf{b}})$, respectively. The result follows by observing that $A(\Omega)$ is minimized by an Ω that activates one s-box in each of rounds $2 \dots (T-1)$, and is maximized by an Ω that activates all M s-boxes in each of rounds $2 \dots (T-1)$. \square

Theorem 4.2.9. *Let $T \geq 2$, and $\mathbf{a}, \mathbf{b} \in \{0, 1\}^M \setminus \mathbf{0}$. If*

$$\begin{aligned} A_{\min} &= wt(\gamma_{\mathbf{a}}) + wt(\gamma_{\mathbf{b}}) + (T - 2) \\ A_{\max} &= wt(\gamma_{\mathbf{a}}) + wt(\gamma_{\mathbf{b}}) + M(T - 2), \end{aligned}$$

then

$$E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right] = \sum_{A=A_{\min}}^{A_{\max}} C_{\mathbf{a}, \mathbf{b}}(A) \cdot \left(\frac{1}{2^n - 1} \right)^A.$$

Proof.

$$\begin{aligned} E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right] &= \frac{1}{\#\text{SPN}} \sum_{\text{spn} \in \text{SPN}} ELP_{\text{spn}}^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \\ &= \frac{1}{\#\text{SPN}} \sum_{\text{spn} \in \text{SPN}} \sum_{\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*} ELCP_{\text{spn}}(\Omega) \quad (4.1) \end{aligned}$$

$$\begin{aligned} &= \sum_{\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*} \left[\frac{1}{\#\text{SPN}} \sum_{\text{spn} \in \text{SPN}} ELCP_{\text{spn}}(\Omega) \right] \\ &= \sum_{\Omega \in \text{ALH}(\mathbf{a}, \mathbf{b})^*} \left(\frac{1}{2^n - 1} \right)^{A(\Omega)} \quad (4.2) \end{aligned}$$

$$= \sum_{A=A_{\min}}^{A_{\max}} C_{\mathbf{a}, \mathbf{b}}(A) \cdot \left(\frac{1}{2^n - 1} \right)^A \quad (4.3)$$

In the above, (4.1) follows from Theorem 3.2.6, (4.2) follows from Lemma 4.2.6, and (4.3) is obtained by grouping characteristics in $\text{ALH}(\mathbf{a}, \mathbf{b})^*$ according to the numbers of s-boxes they make active. The limits on the summation in (4.3) are from Lemma 4.2.8. \square

In Section 4.3 we compute $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ directly for a specific SPN structure by using Theorem 4.2.9. We also approximate this value experimentally by pseudorandomly generating the SPN s-boxes. For such an approximation, the following lemma is useful.

Lemma 4.2.10. *Consider T core SPN rounds. Let the s -boxes and the independent key be fixed. Then for any other fixed independent key (e.g., the all-zero key), there is a selection of s -boxes that produces the equivalent SPN, i.e., the same mapping from plaintexts to ciphertexts. Moreover, varying uniformly over both SPN and the set of all independent keys results in the same distribution of SPNs as does fixing an independent key and varying uniformly over SPN.*

Proof. Let S be a bijective $n \times n$ s -box, let $\boldsymbol{\kappa} \in \{0, 1\}^n$, and define $S_{\boldsymbol{\kappa}}(\mathbf{x}) = S(\mathbf{x} \oplus \boldsymbol{\kappa})$ for all $\mathbf{x} \in \{0, 1\}^n$. Since the s -boxes in the T rounds are chosen independently, it suffices to prove that varying uniformly over all choices of S and all choices of $\boldsymbol{\kappa}$ produces the same distribution of s -boxes as simply varying uniformly over all choices of S . This follows easily from the well-known fact that if S is fixed, then the s -boxes $S_{\boldsymbol{\kappa}}$ form an equivalence class of 2^n bijective s -boxes [109] (the $S_{\boldsymbol{\kappa}}$ are called *cryptographically equivalent* s -boxes). \square

4.2.3 Recursive Formulation for $C_{\mathbf{a},\mathbf{b}}(A)$

In order to make use of Theorem 4.2.9, we need a method for computing the values $C_{\mathbf{a},\mathbf{b}}(A)$. We start by thinking of “constructing” a characteristic in $\text{ALH}(\mathbf{a}, \mathbf{b})^*$ in a round-by-round fashion. Clearly the active s -boxes in round 1 and round T are determined by \mathbf{a} and \mathbf{b} , respectively. In each round t ($2 \leq t \leq T - 1$), we select the s -boxes to be made active, ensuring that we are able to “connect” these active s -boxes to the active s -boxes in the previous round, i.e., verifying that the relevant $W[\gamma, \hat{\gamma}]$ entry is nonzero (recall Definition 3.3.10 and the comment immediately following). For $t = T - 1$, we also need to be able to connect to the active s -boxes in round T . This suggests a recursive formulation for $C_{\mathbf{a},\mathbf{b}}(A)$.

Definition 4.2.11. For $t \geq 2$ and $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, let $D[t, A, \gamma, \hat{\gamma}]$ denote the number of characteristics over rounds $1 \dots t$ that activate a total of A s-boxes, with the restriction that all the characteristics have the same (arbitrary) fixed input mask for round 1 that activates the pattern of s-boxes given by γ , and the same (arbitrary) fixed output mask for round t that activates the pattern of s-boxes given by $\hat{\gamma}$.

It follows immediately that

$$C_{\mathbf{a}, \mathbf{b}}(A) = D[T, A, \gamma_{\mathbf{a}}, \gamma_{\mathbf{b}}].$$

Remark 4.2.12. Although we refer to fixed input and output masks in Definition 4.2.11, these need not be specified, since we are only interested in the *number* of characteristics satisfying the given conditions, and this number depends entirely on the *patterns* of active s-boxes in round 1 and round t , i.e., on γ and $\hat{\gamma}$.

Lemma 4.2.13. Let $t \geq 2$, $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, and $A_{\min} \leq A \leq A_{\max}$, where

$$A_{\min} = wt(\gamma) + wt(\hat{\gamma}) + (t - 2)$$

$$A_{\max} = wt(\gamma) + wt(\hat{\gamma}) + M(t - 2).$$

If $t = 2$, then

$$D[t, A, \gamma, \hat{\gamma}] = \begin{cases} W[\gamma, \hat{\gamma}] & \text{if } A = wt(\gamma) + wt(\hat{\gamma}) \\ 0 & \text{otherwise} \end{cases}$$

If $t \geq 3$, then

$$D[t, A, \gamma, \hat{\gamma}] = \sum_{\bar{w}=\bar{w}_{\min}}^{\bar{w}_{\max}} \sum_{\substack{\bar{\gamma} \in \{0, 1\}^M \\ wt(\bar{\gamma})=\bar{w}}} W[\bar{\gamma}, \hat{\gamma}] \cdot D[t - 1, A - wt(\hat{\gamma}), \gamma, \bar{\gamma}],$$

where

$$\bar{w}_{\min} = \max \{1, A - wt(\gamma) - wt(\hat{\gamma}) - M(t - 3)\}$$

$$\bar{w}_{\max} = \min \{M, A - wt(\gamma) - wt(\hat{\gamma}) - (t - 3)\}.$$

Proof. If $t = 2$, then $A_{\min} = A_{\max} = wt(\gamma) + wt(\hat{\gamma})$. Clearly, $D[2, wt(\gamma) + wt(\hat{\gamma}), \gamma, \hat{\gamma}]$ is equal to the number of ways the active s-boxes in round 1 can be connected to the active s-boxes in round 2, that is,

$$D[2, wt(\gamma) + wt(\hat{\gamma}), \gamma, \hat{\gamma}] = W[\gamma, \hat{\gamma}].$$

It follows that if $A \neq wt(\gamma) + wt(\hat{\gamma})$,

$$D[2, A, \gamma, \hat{\gamma}] = 0.$$

If $t \geq 3$, then each of the t -round characteristics counted by $D[t, A, \gamma, \hat{\gamma}]$ can be viewed as consisting of a $(t - 1)$ -round characteristic for rounds $1 \dots (t - 1)$ that makes $(A - wt(\hat{\gamma}))$ s-boxes active, concatenated with a final mask that activates s-boxes in round t according to the pattern $\hat{\gamma}$. If $\bar{\gamma}$ is the pattern of active s-boxes in round $(t - 1)$, then $D[t - 1, A - wt(\hat{\gamma}), \gamma, \bar{\gamma}]$ is the number of such $(t - 1)$ -round characteristics. It follows that $D[t, A, \gamma, \hat{\gamma}]$ is given by a sum of terms of the form

$$W[\bar{\gamma}, \hat{\gamma}] \cdot D[t - 1, A - wt(\hat{\gamma}), \gamma, \bar{\gamma}],$$

where the summation is over all $\bar{\gamma} \in \{0, 1\}^M$ satisfying certain conditions. Trivially, we have $1 \leq wt(\bar{\gamma}) \leq M$. Now $(A - wt(\gamma) - wt(\hat{\gamma}))$ active s-boxes must be “distributed” among rounds $2 \dots (t - 1)$. Clearly $wt(\bar{\gamma})$ is minimized when *all* the s-boxes in rounds $2 \dots (t - 2)$ are active, and the remaining $(A - wt(\gamma) - wt(\hat{\gamma}) - M(t - 3))$ active s-boxes are located in round $(t - 1)$. It follows that

$$wt(\bar{\gamma}) \geq \bar{w}_{\min} = \max\{1, A - wt(\gamma) - wt(\hat{\gamma}) - M(t - 3)\}.$$

On the other hand, $wt(\bar{\gamma})$ is maximized when there is one active s-box in each of rounds $2 \dots (t - 2)$, forcing the remaining $(A - wt(\gamma) - wt(\hat{\gamma}) - (t - 3))$ active s-boxes

to be located in round $(t - 1)$, so

$$wt(\bar{\gamma}) \leq \bar{w}_{\max} = \min \{M, A - wt(\gamma) - wt(\hat{\gamma}) - (t - 3)\}.$$

□

4.3 Application to Specific SPN Structure

In this section we apply the results of the preceding sections to a specific SPN structure. We consider SPNs in which $M = n$, so $N = n^2$ (these are called *square* SPNs), and in which the linear transformation is a well-known permutation given by Kam and Davida [50]—the permutation connects output bit j of s-box i in round t to input bit i of s-box j in round $(t + 1)$ (here we number bits from left to right, beginning at 1). Figure 4.2 gives an example of such an SPN for the parameters $M = n = 4$ ($N = 16$), and $R = 3$. This SPN structure is useful for testing various results, since not only does it exhibit good cryptographic properties with an increasing number of rounds, but the simplicity of the linear transformation facilitates analysis.

4.3.1 Evaluating the Terms $C_{\mathbf{a},\mathbf{b}}(A)$

The main task in computing $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ using Theorem 4.2.9 is evaluating the terms $C_{\mathbf{a},\mathbf{b}}(A)$, and the challenge here is obtaining the values $W[\gamma, \hat{\gamma}]$. In theory, we can apply the expression in Definition 3.3.10 directly, but this is prohibitive for practical block sizes. However, the symmetry of the Kam and Davida permutation makes derivation of the values $W[\gamma, \hat{\gamma}]$ straightforward, as we show in the next lemma.

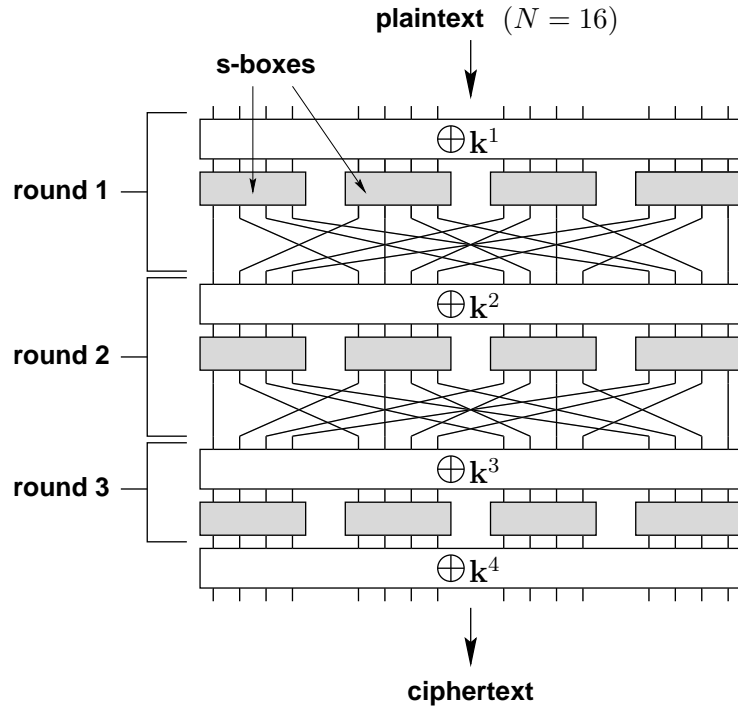


Figure 4.2: SPN with $M = n = 4$ ($N = 16$), $R = 3$, and the permutation of Kam and Davida [50]

Lemma 4.3.1. *Consider an SPN in which $M = n$, and in which the linear transformation is the permutation of Kam and Davida [50]. Let $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, and let $f = wt(\gamma)$, $\ell = wt(\hat{\gamma})$. Then*

$$W[\gamma, \hat{\gamma}] = \sum_{i=1}^{\ell} (-1)^{\ell-i} \binom{\ell}{i} (2^i - 1)^f.$$

Proof. Let $1 \leq t \leq (T - 1)$, let F be a fixed set of f active s-boxes in round t , and let L be a fixed set of ℓ active s-boxes in round $(t + 1)$. We want to determine the number of ways we can connect F to L through the linear transformation, i.e., we want to determine the number of output masks for round t that activate exactly the s-boxes in F , and which are transformed by the linear transformation into input masks for

round $(t + 1)$ that activate exactly the s-boxes in L .

Let $1 \leq i \leq \ell$, and suppose we “mark” i of the s-boxes in L . Let c_i be the number of output masks for round t that activate exactly the s-boxes in F , and which, when transformed by the linear transformation into input masks for round $(t + 1)$, activate *some subset of the i marked s-boxes*. Note that each s-box in round t has exactly one output “wire” connecting it to any given s-box in round $(t + 1)$. So if $S \in F$, an output mask counted by c_i will have 1’s or 0’s on the i wires connecting S to the i marked s-boxes in round $(t + 1)$, and will have 0’s on the remaining $(n - i)$ output wires for S . Therefore, there are $(2^i - 1)$ possible n -bit output masks for S (the all-zero mask is not allowed, since S is active). Since this same argument applies for each s-box in F , we have

$$c_i = (2^i - 1)^f.$$

Noting that there are $\binom{\ell}{i}$ ways to mark i s-boxes in L , and applying the inclusion-exclusion principle, we get

$$\begin{aligned} W[\gamma, \hat{\gamma}] &= \sum_{i=1}^{\ell} (-1)^{\ell-i} \binom{\ell}{i} c_i \\ &= \sum_{i=1}^{\ell} (-1)^{\ell-i} \binom{\ell}{i} (2^i - 1)^f. \end{aligned}$$

□

4.3.2 Computational Results

For the specific SPN structure given above, we used Theorem 4.2.9 to compute $E_{\text{SPN}} \left[\text{ELP}^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for a range of parameters:

$$2 \leq n \leq 10, \quad 2 \leq T \leq 16, \quad \text{and} \quad 1 \leq wt(\gamma_{\mathbf{a}}), wt(\gamma_{\mathbf{b}}) \leq M = n.$$

(For this SPN structure, the result of Theorem 4.2.9 does not depend on the specific values of $\gamma_{\mathbf{a}}$ and $\gamma_{\mathbf{b}}$, but only on $wt(\gamma_{\mathbf{a}})$ and $wt(\gamma_{\mathbf{b}})$.)

In Figure 4.3 we plot the values $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for $M = n = 4$ and $2 \leq T \leq 16$, in the case that $wt(\gamma_{\mathbf{a}}) = wt(\gamma_{\mathbf{b}}) = 1$ (with a \log_{10} scale on the y -axis). On the same graph we also plot experimental values, obtained as follows. For fixed masks $\mathbf{a} = \text{D000}$ (hex) and $\mathbf{b} = \text{0050}$ (hex) (chosen arbitrarily to satisfy $wt(\gamma_{\mathbf{a}}) = wt(\gamma_{\mathbf{b}}) = 1$), and for each value of T , we generated 1000 SPNs at random, and for each SPN we computed $LP_{\text{spn}}^{[1 \dots T]}(\mathbf{a}, \mathbf{b}; \mathbf{0})$ directly from Definition 2.6.2 (by fixing the all-zero key, we're making use of Lemma 4.2.10). We plot the average of these 1000 LP values. Note the strong correspondence between the theoretical values and the experimental values.

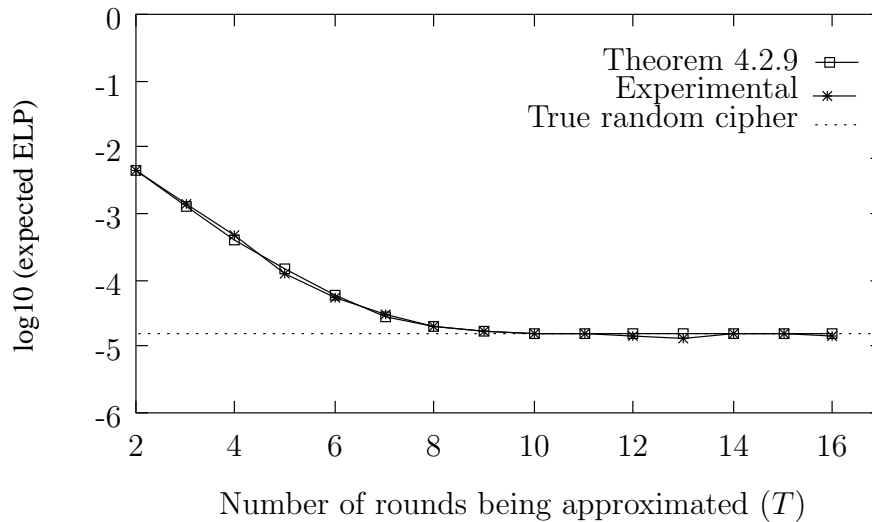


Figure 4.3: $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for $M = n = 4$ and $\mathbf{a} = \text{D000}$ (hex), $\mathbf{b} = \text{0050}$ (hex)

In Figure 4.3 we also observe the apparent convergence of the theoretical and experimental values to a limiting value. Applying Lemma 4.2.5 to an $N \times N$ s-box,

we know that $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ for the true random cipher is given by

$$\frac{1}{(2^N - 1)} = \frac{1}{(2^{16} - 1)} \approx 1.526 \times 10^{-5}.$$

We plot a horizontal line at $y = \log_{10}(1.526 \times 10^{-5}) \approx -4.82$ in Figure 4.3; this value indeed appears to be the limit approached by both the theoretical and experimental curves.

In Figure 4.4 we plot $E_{\text{SPN}} \left[ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for a 64-bit block size ($M = n = 8$). (It is not practical to compute experimental values of $LP_{\text{spn}}^{[1\dots T]}(\mathbf{a}, \mathbf{b}; \mathbf{0})$ for $N = 64$.) Again we observe a strong correspondence between what appears to be a limiting value for $E_{\text{SPN}} \left[ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \right]$ and the value $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ for the true random cipher, where the latter is given by

$$\frac{1}{(2^{64} - 1)} \approx 5.421 \times 10^{-20}.$$

A horizontal line at $y = \log_{10}(5.421 \times 10^{-20}) \approx -19.27$ is plotted in Figure 4.4.

4.4 Conjectures

In carrying out further experiments, the convergence seen in Figure 4.3 and Figure 4.4 was consistently observed for various values of $wt(\gamma_{\mathbf{a}})$ and $wt(\gamma_{\mathbf{b}})$. This leads us to the following conjecture.

Conjecture 4.4.1. *Consider an SPN for which $M = n$ and for which the linear transformation is the permutation of Kam and Davida [50]. Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$. Then*

$$\lim_{T \rightarrow \infty} E_{\text{SPN}} \left[ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \right] = \frac{1}{(2^N - 1)}.$$

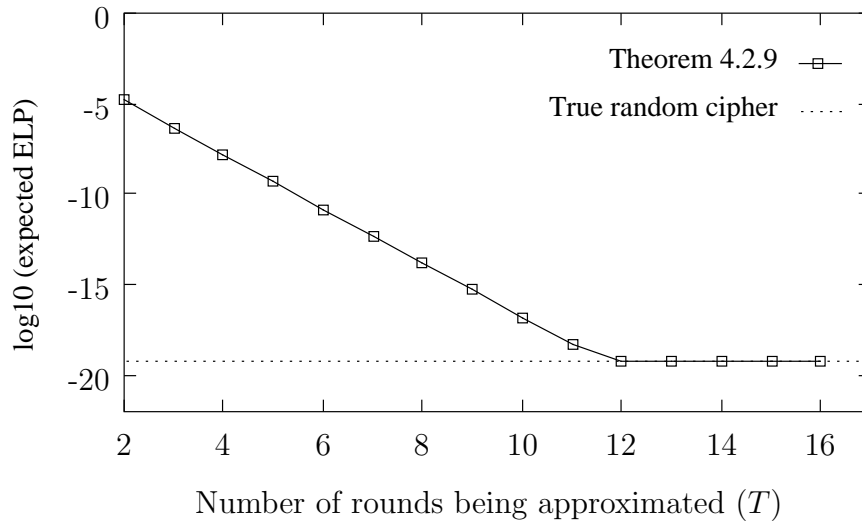


Figure 4.4: $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ for $M = n = 8$ and $wt(\gamma_{\mathbf{a}}) = wt(\gamma_{\mathbf{b}}) = 1$

We would like to generalize Conjecture 4.4.1 to SPNs based on a much larger class of linear transformations. It is easy to show that if an SPN does not satisfy the completeness property (Definition 2.6.7) after some number of rounds, then there are choices of masks \mathbf{a} and \mathbf{b} such that $E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right]$ does not converge to $\frac{1}{(2^N-1)}$. For example, suppose there are infinitely many values of T such that output bit j does not depend on input bit i over rounds $1 \dots T$. Let \mathbf{a} be the mask containing a single 1 in position i , and let \mathbf{b} be the mask containing a single 1 in position j . Then $ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) = 0$ for the same infinite set of values of T , and therefore convergence to $\frac{1}{(2^N-1)}$ is impossible. In light of the above, we introduce the following definition.

Definition 4.4.2. *The linear transformation component of an SPN is called high-level-complete, or HL-complete, if, when all the s-boxes of the SPN are complete, the SPN itself is complete after some number of rounds.*

Remark 4.4.3. Note that we cannot bypass Definition 4.4.2 simply by requiring that the linear transformation be complete, because Lemma 2.6.8 shows that an SPN linear transformation is *never* complete.

For any SPN linear transformation, the HL-completeness property is easy to verify. In particular, the permutation of Kam and Davida is *HL-complete*—for complete s-boxes, it yields a complete SPN after two rounds [50].

We now give our generalized conjecture.

Conjecture 4.4.4. *Consider an SPN whose linear transformation is HL-complete.*

Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$. Then

$$\lim_{T \rightarrow \infty} E_{\text{SPN}} \left[ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \right] = \frac{1}{(2^N - 1)}.$$

4.5 Summary

In this chapter we have considered SPNs with randomly selected s-boxes. For fixed input and output masks, we have derived an exact expression for the expected linear probability value over all choices of s-boxes and all independent keys, and have computed this expression for SPNs with practical block sizes. Further, we have given experimental evidence that the resulting values converge to the corresponding value for the true random cipher with an increasing number of rounds, and we conjecture that this convergence can be proven analytically for a large class of SPNs. This work provides additional quantitative evidence that the SPN structure is a good approximation to the true random cipher.

Chapter 5

Practical Security Against Linear Cryptanalysis for SPNs with Randomly Selected S-Boxes

In this chapter we consider the practical security of SPNs against linear cryptanalysis. We focus primarily on the situation in which the SPN s-boxes are chosen independently and uniformly from the set of all bijective $n \times n$ s-boxes. As explained in Section 4.2, this model is relevant in light of the fact that a number of block ciphers with pseudorandomly generated s-boxes have been proposed and analyzed. We derive a new lower bound on the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis. For common block sizes, this lower bound rapidly approaches 1 with an increasing number of rounds. Our work in this chapter was published in [57].

5.1 Practical Security for Fixed S-Boxes

We continue to let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ denote input and output masks, respectively, for $T \geq 2$ core SPN rounds, and we make use of several definitions from Section 3.3, including the value q , the maximum nontrivial LP value over the SPN s-boxes (Definition 3.3.13).

Most research concerning the practical security of SPNs against linear cryptanalysis has focused on the situation in which the s-boxes are fixed and public (as is the case for the majority of block ciphers). Heys and Tavares [39] make the initial observation that any characteristic $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$ activates at least one s-box in each round, which leads to the following theorem.

Theorem 5.1.1. *If $T \geq 2$ and $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, then $ELCP(\Omega) \leq q^T$.*

(In our development, the result in Theorem 5.1.1 follows from (3.7) and (3.11).) In general, however, if the linear transformation is well designed, any characteristic $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$ will activate (many) more than T s-boxes. The main tool here is the linear branch number, \mathcal{B}_l , given in Definition 3.3.11. The next theorem is due to Kang et al. [54, 55].

Theorem 5.1.2. *If $T \geq 2$, then for any $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, $ELCP(\Omega) \leq q^{\mathcal{B}_l}$.*

By partitioning any $T \geq 2$ core rounds into pairs of consecutive rounds, we get a useful corollary.

Corollary 5.1.3. *Let $T \geq 2$. Then for any $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$,*

$$ELCP(\Omega) \leq \begin{cases} q^{\mathcal{B}_l(\frac{T}{2})} & \text{if } T \text{ is even} \\ q^{\mathcal{B}_l\lfloor\frac{T}{2}\rfloor+1} & \text{if } T \text{ is odd.} \end{cases}$$

Clearly the common theme above is to derive a lower bound, B , on the number of active s-boxes for any $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, from which it follows that $ELCP(\Omega) \leq q^B$. Many results concerning practical security adopt this approach, often using arguments that are tailored to a specific linear transformation or to a class of linear transformations [39]. Daemen and Rijmen [25] use the phrase *wide trail strategy* to refer to a general design methodology that involves choosing a linear transformation to guarantee a large number of active s-boxes (here *trail* is another term for *characteristic*).

Daemen and Rijmen show that $q = 2^{-6}$ and $\mathcal{B}_l = 5$ for the AES [25]. It follows from Corollary 5.1.3 that when $T = 4$, $ELCP(\Omega) \leq q^{10} = 2^{-60}$ for any $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$. However, more involved analysis of the AES linear transformation proves that a minimum of 25 s-boxes must be active for any 4-round characteristic, so $ELCP(\Omega) \leq 2^{-150}$ for $T = 4$ [25]. (By extension, $ELCP(\Omega) \leq 2^{-300}$ for $T = 8$, as noted in Remark 3.2.7.)

Kanda et al. [52], Kanda [51], and Kang et al. [54, 55] consider 64-bit SPNs with 8×8 s-boxes in which the linear transformation has the following structure: each output byte is the XOR of a nonempty subset of the input bytes (such linear transformations are incorporated into the round functions of the Feistel networks Camellia [6] and E2 [53]). Kang et al. prove that for such linear transformations, $\mathcal{B}_l \leq 5$ (they do not show whether or not this upper bound is tight).

In light of Theorem 5.1.2 and Corollary 5.1.3, it is natural to consider maximally diffusive linear transformations, i.e., those for which $\mathcal{B}_l = (M + 1)$ (Remark 3.3.12).

One fruitful approach to the construction of maximally diffusive linear transformations is based on *maximum distance separable* (MDS) codes from the theory of error-correcting codes [73]. MDS codes are used as building blocks for the linear transformations in several SPNs, including the AES (for the smaller 32-bit linear transformation in Figure 2.6), SHARK [99] and SQUARE [24] (predecessors of the AES), and Hierocrypt [95].

5.2 Practical Security for Random S-Boxes

We now consider the practical security of SPNs in which the s-boxes are chosen independently and uniformly from the set of all $2^n!$ bijective $n \times n$ s-boxes. Specifically, we answer the following question: *If the SPN s-boxes are randomly selected, what is the probability that the resulting SPN is practically secure against linear cryptanalysis?* We derive a lower bound on this probability, and we give experimental evidence that this probability rapidly approaches 1 as the number of rounds is increased.

5.2.1 Distribution of ELCP values for Random S-Boxes

Lemma 4.2.4 gives the distribution of LP values for a randomly selected bijective s-box, for fixed input and output masks. In this section we extend Lemma 4.2.4 by deriving the distribution of ELCP values for a fixed (consistent) characteristic when the SPN s-boxes are randomly selected. (Note that Lemma 4.2.6 gives the expected value of this distribution.)

Definition 5.2.1. *Let $\mathcal{E}^{[1 \dots t]}$ denote the set of all consistent characteristics over rounds $1 \dots t$ (see Definition 3.3.5). For $A \geq 2$, let $\mathcal{E}_A^{[1 \dots t]} \subseteq \mathcal{E}^{[1 \dots t]}$ consist of those*

characteristics in $\mathcal{E}_A^{[1\dots t]}$ that make exactly A s -boxes active.

It is also useful to define the following family of sets, for $A \geq 2$:

$$\mathcal{I}_A = \{i_1 i_2 \cdots i_A : i_a \in \{1, 2, \dots, 2^{n-2}\}, \text{ for } 1 \leq a \leq A\}.$$

Lemma 5.2.2. *Let $A \geq 2$, and let $\Omega \in \mathcal{E}_A^{[1\dots T]}$ be fixed. If the SPN s -boxes are chosen independently and uniformly from the set of all bijective $n \times n$ s -boxes, then the set of nonzero values for $ELCP(\Omega)$ is*

$$\left\{ \left(\frac{I}{2^{(n-2)A}} \right)^2 : I \in \mathcal{I}_A \right\}, \quad (5.1)$$

where the value $\left(\frac{I}{2^{(n-2)A}} \right)^2$ occurs with probability

$$2^A \left(\sum_{\substack{i_1, i_2, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\} \\ i_1 i_2 \cdots i_A = I}} \prod_{a=1}^A \frac{\binom{2^{n-1}}{2^{n-2} + i_a}^2}{\binom{2^n}{2^{n-1}}} \right). \quad (5.2)$$

The probability that $ELCP(\Omega) = 0$ is given by

$$\left(\frac{\binom{2^{n-1}}{2^{n-2}}^2}{\binom{2^n}{2^{n-1}}} \right)^A + \sum_{B=1}^{A-1} \left[\binom{A}{B} 2^B \left(\sum_{i_1, i_2, \dots, i_B \in \{1, 2, \dots, 2^{n-2}\}} \prod_{b=1}^B \frac{\binom{2^{n-1}}{2^{n-2} + i_b}^2}{\binom{2^n}{2^{n-1}}} \right) \left(\frac{\binom{2^{n-1}}{2^{n-2}}^2}{\binom{2^n}{2^{n-1}}} \right)^{A-B} \right].$$

Proof. The basic approach here is the same as in the proof of Lemma 4.2.6. Enumerate the s -boxes made active by Ω as S_1, S_2, \dots, S_A . Let \mathbf{Z}_a be a random variable representing the LP value for S_a ($1 \leq a \leq A$). Since we vary over all choices of SPN s -boxes, the \mathbf{Z}_a are independent, and the distribution of \mathbf{Z}_a is given by Lemma 4.2.4. If \mathbf{Z} is a random variable representing the value $ELCP(\Omega)$, then

$$\mathbf{Z} = \mathbf{Z}_1 \cdot \mathbf{Z}_2 \cdots \mathbf{Z}_A.$$

It remains to derive the distribution of \mathbf{Z} . Any nonzero value for \mathbf{Z} is the product of nonzero values for the \mathbf{Z}_a , and therefore has the form

$$\left(\frac{i_1}{2^{n-2}}\right)^2 \left(\frac{i_2}{2^{n-2}}\right)^2 \cdots \left(\frac{i_A}{2^{n-2}}\right)^2,$$

for some $i_1, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\}$, which simplifies to (5.1). Since the \mathbf{Z}_a are independent,

$$\text{Prob} \left\{ \mathbf{Z}_1 = \left(\frac{i_1}{2^{n-2}}\right)^2, \dots, \mathbf{Z}_A = \left(\frac{i_A}{2^{n-2}}\right)^2 \right\} = 2^A \prod_{a=1}^A \frac{\binom{2^{n-1}}{2^{n-2}+i_a}}{\binom{2^n}{2^{n-1}}}. \quad (5.3)$$

For fixed $I \in \mathcal{I}_A$, $\text{Prob} \{ \mathbf{Z} = I \}$ is obtained by summing all terms having the form in (5.3), for all choices of $i_1, i_2, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\}$ satisfying $i_1 i_2 \cdots i_A = I$; this gives (5.2).

If $\mathbf{Z} = 0$, then at least one of the \mathbf{Z}_a must be zero. Let B be the number of \mathbf{Z}_a that are *nonzero*. The probability that $B = 0$, i.e., that all the \mathbf{Z}_a are zero, is

$$\left(\frac{\binom{2^{n-1}}{2^{n-2}}}{\binom{2^n}{2^{n-1}}} \right)^A. \quad (5.4)$$

Now suppose $1 \leq B \leq (A - 1)$. There are $\binom{A}{B}$ ways to choose B of the A active s-boxes to have nonzero LP values; the probability that these B s-boxes will have nonzero LP values is given by

$$2^B \left(\sum_{i_1, i_2, \dots, i_B \in \{1, 2, \dots, 2^{n-2}\}} \prod_{b=1}^B \frac{\binom{2^{n-1}}{2^{n-2}+i_b}}{\binom{2^n}{2^{n-1}}} \right). \quad (5.5)$$

The probability that the remaining $(A - B)$ s-boxes will have zero LP values is

$$\left(\frac{\binom{2^{n-1}}{2^{n-2}}}{\binom{2^n}{2^{n-1}}} \right)^{A-B}. \quad (5.6)$$

Combining (5.4), (5.5), and (5.6) completes the proof. \square

5.2.2 Practical Security Lower Bound

Theorem 5.2.3. *Consider $T \geq 2$ core SPN rounds. Let $0 < \delta \leq 1$. If the SPN s-boxes are chosen independently and uniformly from the set of all bijective $n \times n$ s-boxes, then*

$$\text{Prob} \left\{ \max_{\Omega \in \mathcal{E}^{[1..T]}} ELCP(\Omega) \leq \delta \right\} \geq 1 - \sum_{A=T}^{MT} \left[\# \mathcal{E}_A^{[1..T]} \cdot 2^A \sum_{\substack{I \in \mathcal{I}_A \\ I > (\delta \cdot 2^{(n-2)A})}} \left(\sum_{\substack{i_1, i_2, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\} \\ i_1 i_2 \dots i_A = I}} \prod_{a=1}^A \frac{\binom{2^{n-1}}{2^{n-2} + i_a}^2}{\binom{2^n}{2^{n-1}}} \right) \right]. \quad (5.7)$$

Proof.

$$\begin{aligned} & \text{Prob} \left\{ \max_{\Omega \in \mathcal{E}^{[1..T]}} ELCP(\Omega) \leq \delta \right\} \\ &= 1 - \text{Prob} \left\{ \exists \Omega \in \mathcal{E}^{[1..T]} \text{ such that } ELCP(\Omega) > \delta \right\} \\ &\geq 1 - \sum_{A=T}^{MT} \text{Prob} \left\{ \exists \Omega \in \mathcal{E}_A^{[1..T]} \text{ such that } ELCP(\Omega) > \delta \right\} \\ &\geq 1 - \sum_{A=T}^{MT} \left[\# \mathcal{E}_A^{[1..T]} \cdot \text{Prob} \left\{ ELCP(\Omega) > \delta \text{ for an arbitrary } \Omega \in \mathcal{E}_A^{[1..T]} \right\} \right] \\ &= 1 - \sum_{A=T}^{MT} \left[\# \mathcal{E}_A^{[1..T]} \cdot 2^A \sum_{\substack{I \in \mathcal{I}_A \\ I > (\delta \cdot 2^{(n-2)A})}} \left(\sum_{\substack{i_1, i_2, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\} \\ i_1 i_2 \dots i_A = I}} \prod_{a=1}^A \frac{\binom{2^{n-1}}{2^{n-2} + i_a}^2}{\binom{2^n}{2^{n-1}}} \right) \right]. \end{aligned}$$

The limits on the summation indexed by A are based on the obvious fact that the number of s-boxes made active by any nontrivial $\Omega \in \mathcal{E}^{[1..T]}$ is in the range $T \dots MT$. The final equality follows from application of Lemma 5.2.2. Note that the inequality $I > (\delta \cdot 2^{(n-2)A})$ is simply a convenient form of $(\frac{I}{2^{(n-2)A}} > \delta)$. \square

Remark 5.2.4. The definition of practical security given in Section 3.2.2 requires that the data complexity associated with the best linear characteristic be prohibitive. To

guarantee practical security, it is common to stipulate that the data complexity be greater than the total number of $\langle \text{plaintext}, \text{ciphertext} \rangle$ pairs available, i.e., greater than 2^N . For Theorem 5.2.3, it suffices to choose $\delta = \frac{c}{2^N}$ for some $c < 8$ (this gives a data complexity greater than 2^N for all the success rates in Table 3.1).

Deriving the Terms $\mathcal{E}_A^{[1\dots T]}$

In order to apply Theorem 5.2.3, we need to compute the values $\#\mathcal{E}_A^{[1\dots T]}$. This is easily done using the values $D[t, A, \gamma, \hat{\gamma}]$ from Definition 4.2.11.

Lemma 5.2.5.

$$\#\mathcal{E}_A^{[1\dots T]} = \sum_{\gamma, \hat{\gamma} \in \{0,1\}^M \setminus \mathbf{0}} D[T, A, \gamma, \hat{\gamma}] \cdot (2^n - 1)^{wt(\gamma) + wt(\hat{\gamma})}. \quad (5.8)$$

Proof. Since there are no restrictions on the patterns of active s-boxes in the first and last masks of the characteristics in $\mathcal{E}_A^{[1\dots T]}$, we sum over all $\gamma, \hat{\gamma} \in \{0,1\}^M \setminus \mathbf{0}$ in (5.8). Further, since the first and last masks are not required to be fixed (as in the definition of $D[t, A, \gamma, \hat{\gamma}]$), for each choice of $\gamma, \hat{\gamma}$ we multiply by the numbers of first and last masks that activate patterns of s-boxes given by γ and $\hat{\gamma}$, respectively; this yields the terms $(2^n - 1)^{wt(\gamma) + wt(\hat{\gamma})}$. \square

5.3 Computational Results

To test the effectiveness of the lower bound in Theorem 5.2.3, we return to the SPN structure introduced in Section 4.3. Table 5.1 gives our lower bound for the 64-bit version of this SPN ($M = n = 8$) for various numbers of rounds. We use the value $\delta = \frac{4}{2^N} = 2^{-62}$ in Theorem 5.2.3 (see Remark 5.2.4). No entries are given for 10 or 11 rounds, since the resulting lower bound is less than 0, and therefore not useful.

Clearly, for 12 or more rounds, the probability of generating an SPN that is practically secure against linear cryptanalysis rapidly approaches 1.

Number of rounds	Lower bound from Theorem 5.2.3
10	—
11	—
12	$1 - 2^{-8}$
13	$1 - 2^{-31}$
14	$1 - 2^{-60}$
15	$1 - 2^{-95}$
16	$1 - 2^{-136}$

Table 5.1: Practical security lower bound for 64-bit SPN with Kam and Davida permutation and randomly selected s-boxes

5.3.1 Application to Other SPNs

For SPNs with more diffusive linear transformations, we expect that the lower bound in Theorem 5.2.3 will approach 1 even more rapidly. The reasoning for this is based on the following two points.

1. For fixed $0 < \delta \leq 1$, if $\Omega \in \mathcal{E}_A^{[1 \dots T]}$ then it can be shown that $\text{Prob} \{ELCP(\Omega) > \delta\}$ decreases with increasing values of A (for A in the range $T \dots MT$).
2. For SPNs with more diffusive linear transformations, it is not possible to form consistent characteristics that activate small numbers of s-boxes, i.e., $\mathcal{E}_A^{[1 \dots T]} = 0$ for small values of A .

Overall, this results in the elimination of the larger terms in (5.7) of the form

$$2^A \sum_{\substack{I \in \mathcal{I}_A \\ I > (\delta \cdot 2^{(n-2)A})}} \left(\sum_{\substack{i_1, i_2, \dots, i_A \in \{1, 2, \dots, 2^{n-2}\} \\ i_1 i_2 \dots i_A = I}} \prod_{a=1}^A \frac{\binom{2^{n-1}}{2^{n-2} + i_a}^2}{\binom{2^n}{2^{n-1}}} \right).$$

Although a more diffusive linear transformation will increase the terms $\mathcal{E}_A^{[1 \dots T]}$ for larger values of A , in general this does not appear to compensate for the effect of the first point above, and therefore the outermost sum in (5.7) will be smaller, resulting in a larger lower bound.

5.4 Summary

In this chapter we have considered the practical security of SPNs. Our main result (Theorem 5.2.3) is a lower bound on the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis. This lower bound can be computed for SPNs with typical block sizes, and experimental evidence indicates that this lower bound rapidly approaches 1 with an increasing number of rounds. This lends further support to the cryptographic strength of SPNs with randomly selected s-boxes.

Chapter 6

Provable Security Against Linear Cryptanalysis for SPNs with Fixed S-Boxes

In this chapter we present two new algorithms for evaluating provable security against linear cryptanalysis for SPNs with fixed s-boxes—these algorithms are named KMT1 and KMT2.¹ Both algorithms compute an upper bound on the maximum average linear hull probability (MALHP) for an SPN. In general, KMT2 gives a tighter upper bound than KMT1, but is also, in general, more computationally expensive. KMT1 and KMT2 are the first completely general algorithms for upper bounding the MALHP for SPNs. By “general,” we mean that these algorithms can be applied to an SPN with any linear transformation, and they yield an upper bound that is a function of the number of core encryption rounds. In contrast, other approaches to upper bounding the MALHP either require that the linear transformation have a

¹From the initials of Keliher, Meijer, Tavares.

specific structure, or compute a single value independent of the number of rounds.

Section 6.1 summarizes existing results concerning upper bounding the MALHP for SPNs. After proving some technical lemmas in Section 6.2, we begin our main development in Section 6.3 with a thorough analysis of the underlying approach common to KMT1 and KMT2. In Section 6.4 we present the details specific to KMT1, and in Section 6.5 we do the same for KMT2.

Simplifying Assumption

For the remainder of this chapter, we make the simplifying assumption that the same set of s-boxes is used in every SPN round (these s-boxes may be distinct, or there may be repeated s-boxes within a round). It is straightforward to extend KMT1 and KMT2 to the more general situation in which all the SPN s-boxes are distinct. (It is also straightforward to extend these algorithms to apply to SPNs in which the linear transformation changes from round to round.)

6.1 Upper Bounding the MALHP for SPNs

We now survey the results related to upper bounding the MALHP for SPNs. Since the AES is generally considered to be the most important SPN-based block cipher at the present time, we will use it as a point of comparison. In Section 7.1 we give a much more detailed explanation of the application of KMT1 and KMT2 to the AES.

We continue to use T to denote a number of core SPN rounds, and we make use of \mathcal{B}_l , the linear branch number (Definition 3.3.11), and q , the maximum nontrivial LP value over the SPN s-boxes (Definition 3.3.13). We will focus on the variant of the AES that consists of 10 rounds and uses a 128-bit key (the most widely analyzed

variant so far). Recall that for the AES, $M = 16$, $q = 2^{-6}$, and $\mathcal{B}_l = 5$ (the latter holds because $\mathcal{B}_l = 5$ for the 32-bit linear transformation component of the AES).

Lemma 6.1.1. *Let $0 < \delta \leq 1$, and suppose $MALHP \leq \delta$ for T core SPN rounds. Then $MALHP \leq \delta$ for $T + 1$ core SPN rounds.*

Proof. Follows directly from Lemma 3.2.8 and Lemma 2.6.3. □

Hong et al. [41] presented the first upper bound on the MALHP for SPNs, focusing on SPNs with highly diffusive linear transformations. Their main result is given in the following theorem.

Theorem 6.1.2. *Let $T \geq 2$. If $\mathcal{B}_l = (M + 1)$, then $MALHP \leq q^M$. If $\mathcal{B}_l = M$, then $MALHP \leq q^{M-1}$.*

Since $M = 16$ and $\mathcal{B}_l = 5$ for the AES, Theorem 6.1.2 does not apply. However, Kang et al. [55] later published the following natural generalization of Theorem 6.1.2.

Theorem 6.1.3. *Let $T \geq 2$. Then $MALHP \leq q^{\mathcal{B}_l-1}$.*

Applying Theorem 6.1.3 to the AES gives an upper bound of $q^4 = 2^{-24}$. This corresponds to a data complexity of 2^{29} for a success rate of 96.7% (see Table 3.1), which is not prohibitive.

The next result to appear after that of Hong et al. was our work in [58], in which we presented the KMT1 algorithm. As stated above, KMT1 is completely general in that it can be applied to an SPN with any linear transformation, and it computes an upper bound as a (nonincreasing) function of the number of encryption rounds. The only information KMT1 extracts from the SPN s-boxes is the value q . The upper bound from KMT1 for the AES is plotted in Figure 6.1 (the upper curve). For

$T \geq 7$, the upper bound is 2^{-75} , corresponding to a data complexity of 2^{80} for a 96.7% success rate. This result established the provable security of the AES against linear cryptanalysis.

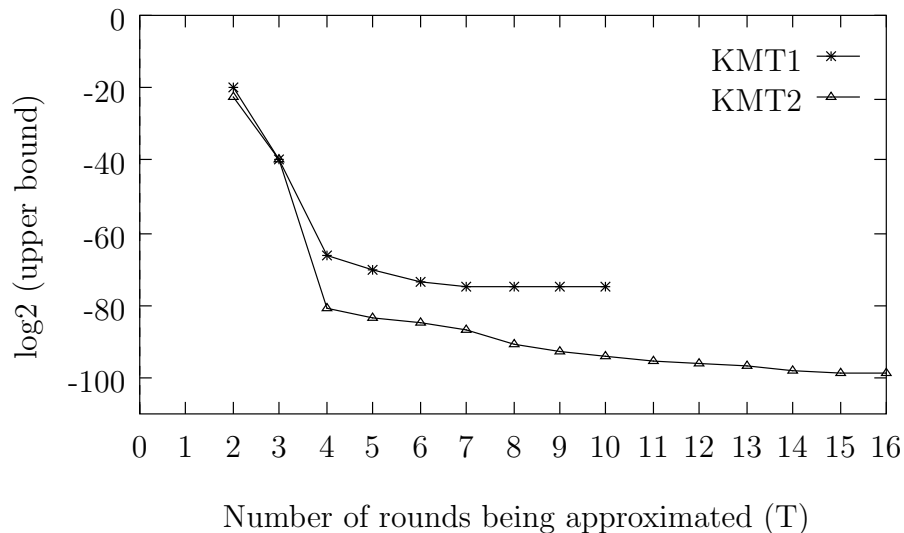


Figure 6.1: Upper bounds from KMT1 and KMT2 for the AES

In [60] we presented our improved algorithm, KMT2. The main difference between KMT1 and KMT2 is that KMT2 incorporates much more detailed information about the distribution of LP values for the s-boxes, resulting in a tighter upper bound in general (KMT2 is also more computationally expensive than KMT1 in general). The upper bound from KMT2 for the AES is plotted in Figure 6.1 (the lower curve). The values from KMT2 are also given in Table 6.1, for $2 \leq T \leq 16$ (the values for the larger values of T are included to illustrate the continued downward progression of the KMT2 upper bound).

Sano et al. [101] then published a result concerning provable security for *nested SPNs*. A nested SPN is an SPN with “large” s-boxes that are themselves small

Number of rounds	Upper bound from KMT2	Number of rounds	Upper bound from KMT2
1	—	9	$2^{-92.4}$
2	$2^{-22.6}$	10	$2^{-94.0}$
3	$2^{-40.0}$	11	$2^{-95.2}$
4	$2^{-80.8}$	12	$2^{-96.2}$
5	$2^{-83.4}$	13	$2^{-97.0}$
6	$2^{-84.7}$	14	$2^{-97.8}$
7	$2^{-87.0}$	15	$2^{-98.4}$
8	$2^{-90.6}$	16	$2^{-99.0}$

Table 6.1: Upper bound from KMT2 for the AES

SPNs. S-boxes and linear transformations are referred to as *high-level* or *low-level* as appropriate. Sano et al. gave the following theorem.

Theorem 6.1.4. *Consider a nested SPN with M_1 high-level s-boxes, each of which is a 2-round SPN containing M_2 low-level s-boxes in each round. Suppose all linear transformations are maximally diffusive, i.e., the high-level linear branch number is $(M_1 + 1)$ and the low-level linear branch number is $(M_2 + 1)$. Then for $T \geq 2$ core high-level rounds, $MALHP \leq q^{M_1 M_2}$.*

Sano et al. demonstrated that the AES can be made to fit the nested SPN structure, with $M_1 = 4$ and $M_2 = 4$. (In fact, this holds for any AES-like SPN (Section 2.5.2).) It follows from Theorem 6.1.4 that for $T \geq 4$ core AES rounds, the MALHP is upper bounded by $q^{16} = 2^{-96}$. This improves on the upper bound from KMT2 for $4 \leq T \leq 12$, but KMT2 gives a tighter upper bound for $T \geq 13$ (this is

relevant because the version of the AES that uses a 256-bit key consists of 14 rounds).

The next result to appear was that of Park et al. [96], which considers provable security for AES-like SPNs. Park et al. proved the following theorem.

Theorem 6.1.5. *Let $T \geq 4$ for an AES-like SPN. Then the MALHP is upper bounded by*

$$\max \{4q^{19} + 6q^{18} + 4q^{17} + q^{16}, 184q^{22} + 912q^{21} + 438q^{20} + 72q^{19} + 4q^{18} + q^{16}\}.$$

Applying Theorem 6.1.5 to the AES gives an upper bound of 1.06×2^{-96} for $T \geq 4$. This is almost identical to the upper bound of 2^{-96} obtained by Sano et al. [101], and was apparently obtained independently.

Park et al. then gave an improved result for the AES in [97]. The basis of their result is the following theorem, which applies to any SPN.

Theorem 6.1.6. *Let the s -boxes in the SPN substitution stage be enumerated from left to right as S_1, S_2, \dots, S_M . Then for $T \geq 2$, the MALHP is upper bounded by*

$$\max \left\{ \max_{\substack{1 \leq i \leq M \\ \alpha \in \{0,1\}^n \setminus \mathbf{0}}} \left[\sum_{\chi \in \{0,1\}^n \setminus \mathbf{0}} (LP^{S_i}(\alpha, \chi))^{B_i} \right], \max_{\substack{1 \leq i \leq M \\ \beta \in \{0,1\}^n \setminus \mathbf{0}}} \left[\sum_{\chi \in \{0,1\}^n \setminus \mathbf{0}} (LP^{S_i}(\chi, \beta))^{B_i} \right] \right\}.$$

When applied to the AES, Theorem 6.1.6 gives an upper bound of 1.44×2^{-27} for $T \geq 2$ rounds. Park et al. were able to use Theorem 6.1.6 in an analysis tailored to AES-like SPNs to obtain an upper bound of $(1.44 \times 2^{-27})^4 \approx 1.075 \times 2^{-106}$ for $T \geq 4$ AES rounds.

In conclusion for this section, there has been much interest recently in upper bounding the MALHP for SPNs. This has resulted in a series of advances, and, in particular, has established the provable security of the AES against linear cryptanalysis (beginning with our KMT1 algorithm). At the time of this writing, however, it

remains the case that KMT1 and KMT2 are the only completely general methods for upper bounding the MALHP for SPNs. We now proceed to an explanation of these two algorithms.

6.2 Technical Lemmas

Lemma 6.2.1. *Let $m \geq 2$, and suppose $\{c_i\}_{i=1}^m, \{d_i\}_{i=1}^m$ are sequences of nonnegative values. Let $\{\dot{c}_i\}_{i=1}^m, \{\dot{d}_i\}_{i=1}^m$ be the sequences obtained by sorting $\{c_i\}$ and $\{d_i\}$, respectively, in nonincreasing order. Then $\sum_{i=1}^m c_i d_i \leq \sum_{i=1}^m \dot{c}_i \dot{d}_i$.*

Proof. Without loss of generality, assume $\{d_i\}$ is sorted in nonincreasing order, so $\dot{d}_i = d_i$. If $m = 2$ and $\{c_i\}$ is not sorted, i.e., if $c_1 < c_2$, then $\dot{c}_1 = c_2$ and $\dot{c}_2 = c_1$, so

$$\begin{aligned} \sum_{i=1}^2 c_i d_i \leq \sum_{i=1}^2 \dot{c}_i \dot{d}_i &\iff c_1 d_1 + c_2 d_2 \leq c_2 \dot{d}_1 + c_1 \dot{d}_2 \\ &\iff (c_2 - c_1) d_2 \leq (c_2 - c_1) d_1 \\ &\iff d_2 \leq d_1, \end{aligned}$$

and $d_2 \leq d_1$ holds because $\{d_i\}$ was assumed to be sorted. Let $m \geq 3$ and assume the lemma holds for $(m - 1)$. Let s be the index of a minimal term in $\{c_i\}$, and let $\{\hat{c}_i\}_{i=1}^m$ be the sequence obtained by exchanging c_s and c_m in $\{c_i\}$. Then $\dot{c}_m = \hat{c}_m$, and therefore sorting $\{\hat{c}_i\}_{i=1}^{m-1}$ in nonincreasing order gives $\{\dot{c}_i\}_{i=1}^{m-1}$. By an argument similar to that of the $m = 2$ case, we have $\sum_{i=1}^m c_i \dot{d}_i \leq \sum_{i=1}^m \hat{c}_i \dot{d}_i$. Applying the induction hypothesis to the first $(m - 1)$ terms of $\{\hat{c}_i\}$ and $\{\dot{d}_i\}$ gives $\sum_{i=1}^{m-1} \hat{c}_i \dot{d}_i \leq \sum_{i=1}^{m-1} \dot{c}_i \dot{d}_i$. Combining these facts, we get

$$\sum_{i=1}^m c_i d_i \leq \sum_{i=1}^m \hat{c}_i \dot{d}_i = \sum_{i=1}^{m-1} \hat{c}_i \dot{d}_i + \hat{c}_m \dot{d}_m \leq \sum_{i=1}^{m-1} \dot{c}_i \dot{d}_i + \dot{c}_m \dot{d}_m = \sum_{i=1}^m \dot{c}_i \dot{d}_i.$$

□

Lemma 6.2.2. *Suppose $\{\dot{c}_i\}_{i=1}^m$, $\{\ddot{c}_i\}_{i=1}^m$, and $\{\dot{d}_i\}_{i=1}^m$ are sequences of nonnegative values, with $\{\dot{d}_i\}$ sorted in nonincreasing order. Suppose there exists \tilde{m} , $1 \leq \tilde{m} \leq m$, such that*

- (a) $\ddot{c}_i \geq \dot{c}_i$, for $1 \leq i \leq \tilde{m}$
- (b) $\ddot{c}_i \leq \dot{c}_i$, for $(\tilde{m} + 1) \leq i \leq m$
- (c) $\sum_{i=1}^m \dot{c}_i \leq \sum_{i=1}^m \ddot{c}_i$

Then $\sum_{i=1}^m \dot{c}_i \dot{d}_i \leq \sum_{i=1}^m \ddot{c}_i \dot{d}_i$.

Proof. If $\tilde{m} = m$, the lemma clearly holds, so assume $1 \leq \tilde{m} < m$. Let

$$\begin{aligned} \dot{A} &= \sum_{i=1}^{\tilde{m}} \dot{c}_i & \dot{B} &= \sum_{i=\tilde{m}+1}^m \dot{c}_i & \dot{C} &= \sum_{i=1}^m \dot{c}_i \\ \ddot{A} &= \sum_{i=1}^{\tilde{m}} \ddot{c}_i & \ddot{B} &= \sum_{i=\tilde{m}+1}^m \ddot{c}_i & \ddot{C} &= \sum_{i=1}^m \ddot{c}_i \end{aligned}$$

By assumption, $\dot{A} \leq \ddot{A}$, $\dot{B} \geq \ddot{B}$, and $\dot{C} \leq \ddot{C}$. Let $\Delta A = \ddot{A} - \dot{A}$ and $\Delta B = \dot{B} - \ddot{B}$.

Then $\Delta A \geq 0$ and $\Delta B \geq 0$. Note that $\Delta A - \Delta B = (\ddot{A} + \ddot{B}) - (\dot{A} + \dot{B}) = \ddot{C} - \dot{C} \geq 0$.

We have

$$\sum_{i=1}^{\tilde{m}} \ddot{c}_i \dot{d}_i \geq \sum_{i=1}^{\tilde{m}} \dot{c}_i \dot{d}_i + \Delta A \cdot \dot{d}_{\tilde{m}} \tag{6.1}$$

$$\sum_{i=\tilde{m}+1}^m \ddot{c}_i \dot{d}_i \geq \sum_{i=\tilde{m}+1}^m \dot{c}_i \dot{d}_i - \Delta B \cdot \dot{d}_{\tilde{m}+1}. \tag{6.2}$$

Adding (6.1) and (6.2), we get

$$\begin{aligned} \sum_{i=1}^m \ddot{c}_i \dot{d}_i &\geq \sum_{i=1}^m \dot{c}_i \dot{d}_i + \Delta A \cdot \dot{d}_{\tilde{m}} - \Delta B \cdot \dot{d}_{\tilde{m}+1} \geq \sum_{i=1}^m \dot{c}_i \dot{d}_i + \Delta A \cdot \dot{d}_{\tilde{m}+1} - \Delta B \cdot \dot{d}_{\tilde{m}+1} \\ &= \sum_{i=1}^m \dot{c}_i \dot{d}_i + (\Delta A - \Delta B) \cdot \dot{d}_{\tilde{m}+1} \\ &\geq \sum_{i=1}^m \dot{c}_i \dot{d}_i. \end{aligned}$$

□

6.3 General Approach for KMT1 and KMT2

In this section we develop the underlying approach common to KMT1 and KMT2. We continue to use the convention for superscripts introduced in Section 3.2, as well as the convention of omitting the linear transformation from the last round of a sequence of consecutive rounds under consideration (Section 4.2.2).

Consider the evaluation of the MALHP for $T \geq 2$ core SPN rounds:

$$MALHP = \max_{\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}} ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}). \quad (6.3)$$

Direct computation would involve encrypting all 2^N inputs through the T rounds for all possible choices of the T subkeys, and then applying the mask \mathbf{a} to each input and the mask \mathbf{b} to each corresponding output—and this would need to be done for all $\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}$. This is computationally infeasible for any practically sized SPN.

An alternate approach to computing each term $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ is to use Nyberg's linear hull theorem (Theorem 3.2.6):

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) = \sum_{\Omega \in ALH(\mathbf{a}, \mathbf{b})^*} ELCP(\Omega). \quad (6.4)$$

Computation of the right-hand side of (6.4) would involve constructing all characteristics $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$ and summing the terms $ELCP(\Omega)$. Again, this is infeasible for practically sized SPNs. However, viewing the problem from this perspective has proven to be very useful, giving us the basis for the KMT1 and KMT2 algorithms. We developed our intuition through the following stages.

1. For all $\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}$, derive an upper bound on $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ —clearly the maximum of these termwise upper bounds is an upper bound on the MALHP.

2. To upper bound $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$, derive a value $B(\mathbf{a}, \mathbf{b})$ that upper bounds $ELCP(\Omega)$ for all $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$, and compute $\#ALH(\mathbf{a}, \mathbf{b})^*$. Then

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \leq B(\mathbf{a}, \mathbf{b}) \cdot \#ALH(\mathbf{a}, \mathbf{b})^*.$$

3. To reduce computational complexity, note that the *number* of characteristics in $ALH(\mathbf{a}, \mathbf{b})^*$ depends only on $\gamma_{\mathbf{a}}$ and $\gamma_{\mathbf{b}}$, not on the specific values of \mathbf{a} and \mathbf{b} . Therefore, for $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, let $C(\gamma, \hat{\gamma})$ be the number of characteristics in $ALH(\mathbf{a}, \mathbf{b})^*$ for any $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ satisfying $\gamma_{\mathbf{a}} = \gamma$, $\gamma_{\mathbf{b}} = \hat{\gamma}$. Let $B(\gamma, \hat{\gamma})$ be a value that upper bounds $ELCP(\Omega)$ for all $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ satisfying $\gamma_{\mathbf{a}} = \gamma$, $\gamma_{\mathbf{b}} = \hat{\gamma}$, and for all $\Omega \in ALH(\mathbf{a}, \mathbf{b})^*$. If we define

$$UB^{[1\dots T]}[\gamma, \hat{\gamma}] \stackrel{\text{def}}{=} B(\gamma, \hat{\gamma}) \cdot C(\gamma, \hat{\gamma}),$$

it follows that

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \leq UB^{[1\dots T]}[\gamma_{\mathbf{a}}, \gamma_{\mathbf{b}}],$$

and therefore

$$MALHP \leq \max_{\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}} UB^{[1\dots T]}[\gamma, \hat{\gamma}]. \quad (6.5)$$

Note that the maximum in (6.5) is taken over approximately 2^{2M} elements, which is feasible for many modern block ciphers (e.g., $M = 16$ for the AES).

4. Computation of the values $B(\gamma, \hat{\gamma})$ and $C(\gamma, \hat{\gamma})$ can be simplified through a recursive approach, i.e., the values computed for $(T - 1)$ core rounds can be used to derive the corresponding values for T core rounds.
5. It remains to determine the values $B(\gamma, \hat{\gamma})$ and $C(\gamma, \hat{\gamma})$ in the “base case,” namely $T = 2$. The method for choosing the values $B(\gamma, \hat{\gamma})$ is especially critical here.

Remark 6.3.1. The above steps capture the essence of the KMT1 and KMT2 algorithms. However, additional refinements are necessary to ensure that the resulting upper bounds on the MALHP are sufficiently small to be useful. Because of these refinements, the basic structure above may not be immediately apparent in the pseudocode given later in the chapter.

It is worth singling out one of the main ideas in Step 3, namely the derivation of upper bound values $UB^{[1\dots T]}[\gamma, \hat{\gamma}]$ such that the following property holds:

UB Property (for T rounds). For all $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$,

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \leq UB^{[1\dots T]}[\gamma_{\mathbf{a}}, \gamma_{\mathbf{b}}].$$

6.3.1 Recursive Method for Computing $UB^{[1\dots T]}[\gamma, \hat{\gamma}]$

In what follows, we detail our method for computing the values $UB^{[1\dots T]}[\gamma, \hat{\gamma}]$, leaving certain parameters unspecified, and we prove the correctness of our approach. The KMT1 and KMT2 algorithms are defined by selecting particular values for these unspecified parameters.

Let $T \geq 2$, and let \mathbf{a} and \mathbf{b} be nonzero N -bit input and output masks, respectively, for T core SPN rounds. Consider the diagram in Figure 6.2. If $\mathbf{x} \in \{0, 1\}^N$ is an output mask for rounds $1 \dots (T - 1)$ (minus the linear transformation for round $(T - 1)$), and if \mathbf{y} is the corresponding input mask for round T (\mathbf{x} and \mathbf{y} are related according to Lemma 3.3.4), then from Lemma 3.2.8 we have

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) = \sum_{\mathbf{x} \in \{0, 1\}^N} ELP^{[1\dots(T-1)]}(\mathbf{a}, \mathbf{x}) \cdot ELP^T(\mathbf{y}, \mathbf{b}). \quad (6.6)$$

We can eliminate terms from the sum in (6.6) that are trivially zero by making the following observations based on Section 3.3.

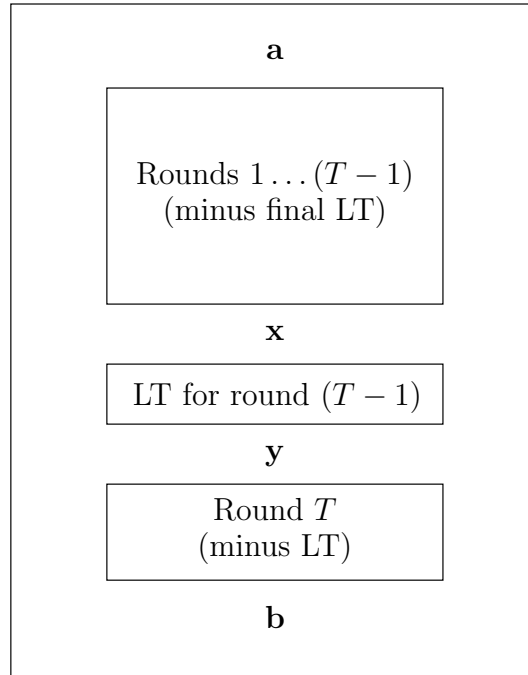


Figure 6.2: Important values for KMT1 and KMT2

Observation 6.3.2. For $T \geq 2$, if $\gamma_{\mathbf{y}} \neq \gamma_{\mathbf{b}}$, then $ELP^T(\mathbf{y}, \mathbf{b}) = 0$, so we can limit consideration to values \mathbf{x} for which the corresponding values \mathbf{y} satisfy $\gamma_{\mathbf{y}} = \gamma_{\mathbf{b}}$.

Observation 6.3.3. If $T = 2$ and $\gamma_{\mathbf{x}} \neq \gamma_{\mathbf{a}}$, then $ELP^{[1 \dots (T-1)]}(\mathbf{a}, \mathbf{x}) = ELP^1(\mathbf{a}, \mathbf{x}) = 0$, so we can eliminate all such \mathbf{x} .

Let L be the number of values $\mathbf{x} \in \{0, 1\}^N$ that are not eliminated by Observations 6.3.2 and 6.3.3. Enumerate these values as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L$, and let the corresponding \mathbf{y} values be $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L$, respectively. For $1 \leq i \leq L$, let

$$c_i = ELP^{[1 \dots (T-1)]}(\mathbf{a}, \mathbf{x}_i)$$

$$d_i = ELP^T(\mathbf{y}_i, \mathbf{b}).$$

Then (6.6) can be rewritten as

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^L c_i d_i. \quad (6.7)$$

Now let $\{\dot{c}_i\}$ and $\{\dot{d}_i\}$ be the sequences obtained by sorting $\{c_i\}$ and $\{d_i\}$, respectively, in nonincreasing order. It follows from Lemma 6.2.1 that

$$\sum_{i=1}^L c_i d_i \leq \sum_{i=1}^L \dot{c}_i \dot{d}_i. \quad (6.8)$$

Since computation of the values c_i , d_i is generally infeasible, it is not practical to compute the right-hand side of (6.8). However, suppose it *is* feasible to find nonincreasing sequences $\{u_i\}$ and $\{v_i\}$ (each of length at least L) that upper bound $\{\dot{c}_i\}$ and $\{\dot{d}_i\}$, respectively, i.e., $\dot{c}_i \leq u_i$ and $\dot{d}_i \leq v_i$, for $1 \leq i \leq L$. It follows that

$$\sum_{i=1}^L \dot{c}_i \dot{d}_i \leq \sum_{i=1}^L u_i v_i, \quad (6.9)$$

and therefore $\sum_{i=1}^L u_i v_i$ is an upper bound on $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$, i.e.,

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \leq \sum_{i=1}^L u_i v_i \quad (6.10)$$

(combine (6.7), (6.8), and (6.9)). Moreover, it is possible to refine this upper bound by application of Lemma 6.2.2. The key observation is this: if we wish to obtain an upper bound by replacing a nonincreasing sequence such as $\{\dot{c}_i\}$ with a nonincreasing sequence $\{\eta_i\}$, it is not necessary that $\{\eta_i\}$ “completely dominate” $\{\dot{c}_i\}$, i.e., it is not necessary that $\dot{c}_i \leq \eta_i$ for $1 \leq i \leq L$. Instead, under certain conditions, it suffices for an initial subsequence of $\{\eta_i\}$ to dominate the corresponding subsequence of $\{\dot{c}_i\}$ (see Lemma 6.2.2). Therefore, given a sequence $\{u_i\}$ satisfying $\dot{c}_i \leq u_i$ for all i , we can tighten the upper bound in (6.10) by *truncating* $\{u_i\}$ to obtain a sequence $\{\tilde{u}_i\}$

(similarly, we truncate $\{v_i\}$ to obtain $\{\tilde{v}_i\}$) in the following fashion. We know that $\sum_{i=1}^L \dot{c}_i = \sum_{i=1}^L c_i \leq 1$ (similarly, $\sum_{i=1}^L \dot{d}_i \leq 1$) by Lemma 2.6.3. If $\sum_{i=1}^L u_i \leq 1$, let $\tilde{u}_i = u_i$, for $1 \leq i \leq L$. If $\sum_{i=1}^L u_i > 1$, let L_u be maximum such that $\sum_{i=1}^{L_u} u_i \leq 1$, and let $\{\tilde{u}_i\}_{i=1}^L$ consist of the first L terms of

$$\left\langle u_1, u_2, \dots, u_{L_u}, \left(1 - \sum_{i=1}^{L_u} u_i\right), 0, 0, 0, \dots \right\rangle. \quad (6.11)$$

Since $\{\dot{c}_i\}$, $\{\tilde{u}_i\}$, and $\{\dot{d}_i\}$, respectively, satisfy the conditions on the three sequences in the statement of Lemma 6.2.2, we have

$$\sum_{i=1}^L \dot{c}_i \dot{d}_i \leq \sum_{i=1}^L \tilde{u}_i \dot{d}_i. \quad (6.12)$$

Similarly, since $\{\dot{d}_i\}$, $\{\tilde{v}_i\}$, and $\{\tilde{u}_i\}$, respectively, also satisfy the same conditions, we have

$$\sum_{i=1}^L \tilde{u}_i \dot{d}_i \leq \sum_{i=1}^L \tilde{u}_i \tilde{v}_i. \quad (6.13)$$

Combining (6.7), (6.8), (6.12), and (6.13) gives

$$ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b}) \leq \sum_{i=1}^L \tilde{u}_i \tilde{v}_i. \quad (6.14)$$

The inequality in (6.14) is the basis for the upper bound computed by KMT1 and KMT2. (Of course, it remains to specify the sequences $\{\tilde{u}_i\}$ and $\{\tilde{v}_i\}$.)

Clearly the upper bound in (6.14) improves on that in (6.10), since

$$\sum_{i=1}^L \tilde{u}_i \tilde{v}_i \leq \sum_{i=1}^L u_i v_i,$$

and in many cases the inequality is strict.

Remark 6.3.4. In most instances the upper bound on $ELP^{[1\dots T]}(\mathbf{a}, \mathbf{b})$ computed by KMT1/KMT2 is exactly $\sum_{i=1}^L \tilde{u}_i \tilde{v}_i$; however, there are certain cases in which the

upper bound will be *strictly larger than* $\sum_{i=1}^L \tilde{u}_i \tilde{v}_i$ —this contingency is discussed at length in the proof of Theorem 6.3.7 and again briefly in the proof of Theorem 6.3.8.

Conditions on $\{u_i\}$ and $\{v_i\}$

We require that the following conditions on $\{u_i\}$ and $\{v_i\}$ be satisfied:

1. The sequence $\{u_i\}$ depends only on $\gamma_{\mathbf{a}}$ (not on the specific value of \mathbf{a}). Augmenting our current notation, for each $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$, we require a nonincreasing sequence $\{u_i^\gamma\}$ such that for any $\mathbf{a} \in \{0, 1\}^N \setminus \mathbf{0}$ satisfying $\gamma_{\mathbf{a}} = \gamma$, if $\{\dot{c}_i\}_{i=1}^L$ consists of terms of the form $ELP^{[1 \dots (T-1)]}(\mathbf{a}, \mathbf{x})$ (each for a distinct value of \mathbf{x}) sorted in nonincreasing order, then $\dot{c}_i \leq u_i^\gamma$, for $1 \leq i \leq L$.
2. Similarly, the sequence $\{v_i\}$ depends only on $\gamma_{\mathbf{b}}$ —for each $\hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, we require a nonincreasing sequence $\{v_i^{\hat{\gamma}}\}$ such that for any $\mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ satisfying $\gamma_{\mathbf{b}} = \hat{\gamma}$, if $\{\dot{d}_i\}_{i=1}^L$ consists of terms of the form $ELP^T(\mathbf{y}, \mathbf{b})$ (each for a distinct value of \mathbf{y}) sorted in nonincreasing order, then $\dot{d}_i \leq v_i^{\hat{\gamma}}$, for $1 \leq i \leq L$.
3. Derivation of the values in the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ is computationally feasible.

Using this augmented notation, the truncated version of $\{u_i^\gamma\}$ is $\{\tilde{u}_i^\gamma\}$, the truncated version of $\{v_i^{\hat{\gamma}}\}$ is $\{\tilde{v}_i^{\hat{\gamma}}\}$, and (6.14) becomes

$$ELP^{[1 \dots T]}(\mathbf{a}, \mathbf{b}) \leq \sum_{i=1}^L \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}} \quad (6.15)$$

(where $\gamma = \gamma_{\mathbf{a}}, \hat{\gamma} = \gamma_{\mathbf{b}}$).

Remark 6.3.5. For all $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, clearly we require that the lengths of the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ be at least as large as any value L that may occur.

Repeated Terms in $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$

We address a final technical matter before presenting general algorithms for the two cases $T = 2$ and $T \geq 3$. Since L may be large, direct computation of the sum $\sum_{i=1}^L \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}}$ in (6.15) may not be possible, even if calculation of individual values \tilde{u}_i^γ and $\tilde{v}_i^{\hat{\gamma}}$ is feasible. However, actual sequences that we have used for $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ have had large numbers of repeated terms, greatly simplifying the computation. We now develop notation to handle this.

Let \overline{D}^γ be the number of *distinct* terms in $\{u_i^\gamma\}$; represent these distinct terms in nonincreasing order as

$$\langle \overline{\rho}_1^\gamma, \overline{\rho}_2^\gamma, \dots, \overline{\rho}_{\overline{D}^\gamma}^\gamma \rangle.$$

Let $\overline{\phi}_j^\gamma$ be the number of occurrences of $\overline{\rho}_j^\gamma$ in $\{u_i^\gamma\}$, for $1 \leq j \leq \overline{D}^\gamma$. With this notation, $\{u_i^\gamma\}$ is given by

$$\left\langle \underbrace{\overline{\rho}_1^\gamma, \dots, \overline{\rho}_1^\gamma}_{\overline{\phi}_1^\gamma \text{ terms}}, \underbrace{\overline{\rho}_2^\gamma, \dots, \overline{\rho}_2^\gamma}_{\overline{\phi}_2^\gamma \text{ terms}}, \dots, \underbrace{\overline{\rho}_{\overline{D}^\gamma}^\gamma, \dots, \overline{\rho}_{\overline{D}^\gamma}^\gamma}_{\overline{\phi}_{\overline{D}^\gamma}^\gamma \text{ terms}} \right\rangle. \quad (6.16)$$

Similarly, let $\underline{D}^{\hat{\gamma}}$ be the number of distinct terms in $\{v_i^{\hat{\gamma}}\}$; represent these distinct terms in nonincreasing order as

$$\langle \underline{\rho}_1^{\hat{\gamma}}, \underline{\rho}_2^{\hat{\gamma}}, \dots, \underline{\rho}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}} \rangle.$$

Let $\underline{\phi}_j^{\hat{\gamma}}$ be the number of occurrences of $\underline{\rho}_j^{\hat{\gamma}}$ in $\{v_i^{\hat{\gamma}}\}$, for $1 \leq j \leq \underline{D}^{\hat{\gamma}}$. Then $\{v_i^{\hat{\gamma}}\}$ is given by

$$\left\langle \underbrace{\underline{\rho}_1^{\hat{\gamma}}, \dots, \underline{\rho}_1^{\hat{\gamma}}}_{\underline{\phi}_1^{\hat{\gamma}} \text{ terms}}, \underbrace{\underline{\rho}_2^{\hat{\gamma}}, \dots, \underline{\rho}_2^{\hat{\gamma}}}_{\underline{\phi}_2^{\hat{\gamma}} \text{ terms}}, \dots, \underbrace{\underline{\rho}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}}, \dots, \underline{\rho}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}}}_{\underline{\phi}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}} \text{ terms}} \right\rangle. \quad (6.17)$$

Remark 6.3.6. The use of the *overline* symbol in notation such as \overline{D}^γ , $\overline{\rho}_j^\gamma$, and $\overline{\phi}_j^\gamma$ is a mnemonic indicating that these values relate to LP values that are derived using a fixed *input* mask (namely **a**). Correspondingly, the use of the *underline* symbol indicates values related to LP values derived using a fixed *output* mask (namely **b**).

We also define the partial sums $\overline{\Phi}_0^\gamma = \overline{\Lambda}_0^\gamma = \underline{\Phi}_0^{\hat{\gamma}} = \underline{\Lambda}_0^{\hat{\gamma}} = 0$, and

$$\begin{aligned} \overline{\Phi}_J^\gamma &= \sum_{j=1}^J \overline{\phi}_j^\gamma & \text{and} & & \overline{\Lambda}_J^\gamma &= \sum_{j=1}^J \overline{\rho}_j^\gamma \cdot \overline{\phi}_j^\gamma, & \text{for } 1 \leq J \leq \overline{D}^\gamma, \\ \underline{\Phi}_J^{\hat{\gamma}} &= \sum_{j=1}^J \underline{\phi}_j^{\hat{\gamma}} & \text{and} & & \underline{\Lambda}_J^{\hat{\gamma}} &= \sum_{j=1}^J \underline{\rho}_j^{\hat{\gamma}} \cdot \underline{\phi}_j^{\hat{\gamma}}, & \text{for } 1 \leq J \leq \underline{D}^{\hat{\gamma}}. \end{aligned}$$

6.3.2 $T = 2$ Case

In the case $T = 2$, the situation in Figure 6.2 simplifies to that in Figure 6.3. Specifically, the values $ELP^{[1 \dots (T-1)]}(\mathbf{a}, \mathbf{x})$, which are upper bounded by $\{u_i^\gamma\}$, are

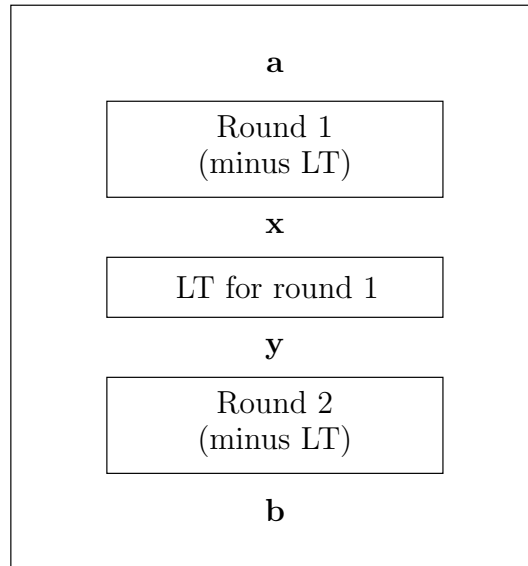


Figure 6.3: Important values for KMT1 and KMT2 ($T = 2$ case)

computed over a single substitution stage, as are the values $ELP^T(\mathbf{y}, \mathbf{b})$, which are upper bounded by $\{v_i^{\hat{\gamma}}\}$. As well, when $T = 2$, both Observations 6.3.2 and 6.3.3 apply, and therefore the number of terms in $\{c_i\}$ and $\{d_i\}$, namely L , is equal to the number of ways we can connect a pattern of s-boxes specified by $\gamma = \gamma_{\mathbf{a}}$ to a pattern of s-boxes specified by $\hat{\gamma} = \gamma_{\mathbf{b}}$ via the linear transformation, i.e., $L = W[\gamma, \hat{\gamma}]$ (see Definition 3.3.10).

Theorem 6.3.7. *Let $T = 2$, and suppose the values $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ are computed using the algorithm in Figure 6.4. Then the UB Property holds.*

Proof. For this proof, “Line X ” refers to the X^{th} line in Figure 6.4. We want to show that the value $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ computed in Figure 6.4 is either the sum $\sum_{i=1}^L \tilde{u}_i^{\gamma} \tilde{v}_i^{\hat{\gamma}}$ from (6.15) (the general case), or a value that upper bounds this sum (a contingency we discuss at the end of this proof). Recall that $\{\tilde{u}_i^{\gamma}\}$ is the possibly truncated version of $\{u_i^{\gamma}\}$, and $\{\tilde{v}_i^{\hat{\gamma}}\}$ is the possibly truncated version of $\{v_i^{\hat{\gamma}}\}$.

First we deal with the general case. We want to show that the algorithm computes the sum of the first $L = W[\gamma, \hat{\gamma}] \stackrel{\text{def}}{=} W$ term-by-term products of the sequences in (6.16) and (6.17), stopping if the partial sum for either sequence is ≥ 1 (appropriately handling the final term).

Note that the algorithm proceeds “group-by-group” through the groups of repeated terms in $\{u_i^{\gamma}\}$ and $\{v_i^{\hat{\gamma}}\}$, not term-by-term. The variable h is the index of the current group in $\{u_i^{\gamma}\}$. Since all the terms in this group have the same value, namely $\bar{\rho}_h^{\gamma}$, the function $\text{IncSum2}()$ computes the sum of the matching terms in $\{v_i^{\hat{\gamma}}\}$, and multiplies this sum (stored in $\Delta\lambda$) by $\bar{\rho}_h^{\gamma}$ (Line 20).

To ensure that we sum at most W terms of the form $u_i^{\gamma} v_i^{\hat{\gamma}}$, we use the check $(\bar{\Phi}_h^{\gamma} \leq W)$ in Line 6, and handle the case $(\bar{\Phi}_h^{\gamma} > W)$ in the conditional statement

beginning at Line 9. Since $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ are assumed to have length at least $W = L$ (see Remark 6.3.5), then either the condition $(\bar{\Phi}_h^\gamma \leq W)$ will fail for some h satisfying $1 \leq h \leq \bar{D}^\gamma$; or, if it happens that W is exactly equal to $\bar{\Phi}_{\bar{D}^\gamma}^\gamma$, h will be incremented to $\bar{D}^\gamma + 1$, and the condition $(h \leq \bar{D}^\gamma)$ in Lines 6 and 9 will fail (assuming no other condition fails first).

If $\{u_i^\gamma\}$ needs to be truncated, i.e., if $\sum_{i=1}^W u_i^\gamma > 1$, there will be an h ($1 \leq h \leq \bar{D}^\gamma$) for which $(\bar{\Lambda}_h^\gamma > 1)$; we check for this in Line 6, and handle this occurrence in the conditional statement beginning at Line 9. In Line 10, the value W' is calculated—this is the number of terms of $\{u_i^\gamma\}$ required for a partial sum equal to 1. The minimum of W and W' is then passed to $\text{IncSum2}()$ via the variable W_{end} . Note that W' may not be an integer, but may instead be equal to $m + \varepsilon$, where m is an integer and $0 < \varepsilon < 1$. Here ε represents the fraction of the term u_{m+1}^γ required for a partial sum equal to 1, that is, $(\sum_{i=1}^m u_i^\gamma) + (\varepsilon \cdot u_{m+1}^\gamma) = 1$. In the notation of (6.11), $m = L_u$ and $\varepsilon = \left(1 - \sum_{i=1}^{L_u} u_i^\gamma\right)$.

The variable λ is the partial sum of the terms in $\{v_i^{\hat{\gamma}}\}$ that have already been “used,” i.e., multiplied by the corresponding terms of $\{u_i^\gamma\}$. If $\{v_i^{\hat{\gamma}}\}$ needs to be truncated, $\text{IncSum2}()$ will set λ to 1, and the check $(\lambda < 1)$ in Lines 6 and 9 will fail.

Finally, we consider the technical situation in which the value $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ computed in Figure 6.4 is not the sum $\sum_{i=1}^W \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}}$, but instead is strictly larger than this sum (a contingency first mentioned in Remark 6.3.4). This occurs when:

1. both $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ need to be truncated, i.e., $\sum_{i=1}^W u_i^\gamma > 1$ and $\sum_{i=1}^W v_i^{\hat{\gamma}} > 1$,
2. both $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ need to be truncated at exactly the same point—in other words, if L_u is maximum such that $\sum_{i=1}^{L_u} u_i^\gamma \leq 1$, and L_v is maximum such that $\sum_{i=1}^{L_v} v_i^{\hat{\gamma}} \leq 1$, then $L_u = L_v$,

3. the residual terms for both sequences are nonzero, i.e., $\left[1 - \left(\sum_{i=1}^{L_u} u_i^\gamma\right)\right] > 0$
and $\left[1 - \left(\sum_{i=1}^{L_v} v_i^\gamma\right)\right] > 0$.

Define $\tilde{W} \stackrel{\text{def}}{=} L_u = L_v$ (clearly, $\tilde{W} < W$). Let $0 < \varepsilon_u < 1$ be the fraction of the term $u_{\tilde{W}+1}^\gamma$ required for a partial sum of $\{u_i^\gamma\}$ equal to 1, i.e.,

$$\left(\sum_{i=1}^{\tilde{W}} u_i^\gamma\right) + \left(\varepsilon_u \cdot u_{\tilde{W}+1}^\gamma\right) = 1. \quad (6.18)$$

Similarly, let $0 < \varepsilon_v < 1$ be the fraction of the term $v_{\tilde{W}+1}^{\hat{\gamma}}$ required for a partial sum of $\{v_i^{\hat{\gamma}}\}$ equal to 1, i.e.,

$$\left(\sum_{i=1}^{\tilde{W}} v_i^{\hat{\gamma}}\right) + \left(\varepsilon_v \cdot v_{\tilde{W}+1}^{\hat{\gamma}}\right) = 1. \quad (6.19)$$

Then

$$\sum_{i=1}^W \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}} = \sum_{i=1}^{\tilde{W}} u_i^\gamma v_i^{\hat{\gamma}} + \left(\varepsilon_u \cdot u_{\tilde{W}+1}^\gamma\right) \left(\varepsilon_v \cdot v_{\tilde{W}+1}^{\hat{\gamma}}\right). \quad (6.20)$$

However, the actual upper bound computed in Figure 6.4 will be one of the two values

$$\sum_{i=1}^{\tilde{W}} u_i^\gamma v_i^{\hat{\gamma}} + \left(\varepsilon_u \cdot u_{\tilde{W}+1}^\gamma \cdot v_{\tilde{W}+1}^{\hat{\gamma}}\right), \quad \sum_{i=1}^{\tilde{W}} u_i^\gamma v_i^{\hat{\gamma}} + \left(\varepsilon_v \cdot u_{\tilde{W}+1}^\gamma \cdot v_{\tilde{W}+1}^{\hat{\gamma}}\right),$$

both of which are strictly larger than the right-hand side of (6.20). To see this, note that there will be a value of h for which $(\bar{\Lambda}_h^\gamma > 1)$ (since $\sum_{i=1}^W u_i^\gamma > 1$). This will cause the While loop in Lines 6–8 to terminate, and execution will jump to Line 9. The value W' computed in Line 10 will be $(\tilde{W} + \varepsilon_u) < W$, and therefore $(\tilde{W} + \varepsilon_u)$ will be assigned to W_{end} and passed to $\text{IncSum2}()$. In Line 16, the value assigned to $\Delta\lambda$ will be such that

$$(\lambda + \Delta\lambda) = \sum_{i=1}^{\tilde{W}} v_i^{\hat{\gamma}} + \left(\varepsilon_u \cdot v_{\tilde{W}+1}^{\hat{\gamma}}\right) \quad (6.21)$$

(note the appearance of the term ε_u , not ε_v , in (6.21)). If $\varepsilon_u \leq \varepsilon_v$, then $(\lambda + \Delta\lambda) \leq 1$ (from (6.19) and (6.21)), so the condition in Line 17 fails and Line 18 is skipped. The value returned by $\text{IncSum2}()$ is $(\bar{\rho}_h^\gamma \cdot \Delta\lambda) = \left(u_{\bar{W}+1}^\gamma \cdot \Delta\lambda\right)$, and therefore the value assigned to $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ is

$$\sum_{i=1}^{\bar{W}} u_i^\gamma v_i^{\hat{\gamma}} + \left(\varepsilon_u \cdot u_{\bar{W}+1}^\gamma \cdot v_{\bar{W}+1}^{\hat{\gamma}}\right).$$

If $\varepsilon_u > \varepsilon_v$, then $(\lambda + \Delta\lambda) > 1$, so Line 18 is executed. It follows that the value assigned to $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ is

$$\sum_{i=1}^{\bar{W}} u_i^\gamma v_i^{\hat{\gamma}} + \left(\varepsilon_v \cdot u_{\bar{W}+1}^\gamma \cdot v_{\bar{W}+1}^{\hat{\gamma}}\right).$$

□

6.3.3 $T \geq 3$ Case

The structure of our approach in the case $T \geq 3$ is essentially the same as in the case $T = 2$, but takes into account the following differences. (As above, let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$ be fixed, and let $\gamma = \gamma_{\mathbf{a}}$, $\hat{\gamma} = \gamma_{\mathbf{b}}$.)

1. In order to upper bound a nonincreasing sequence $\{\dot{c}_i\}$ consisting of terms of the form $ELP^{[1\dots(T-1)]}(\mathbf{a}, \mathbf{x})$, we replace the sequence $\{u_i^\gamma\}$ used in the case $T = 2$ with a sequence of values of the form $UB^{[1\dots(T-1)]}[\]$ (computed recursively). We continue to use the sequence $\{v_i^{\hat{\gamma}}\}$.
2. Since Observation 6.3.2 applies in the case $T \geq 3$, but Observation 6.3.3 does not, we do not have a simple expression for the value L (the number of terms in (6.15)), so there is no condition in the pseudocode for the case $T \geq 3$ that corresponds to the condition $(\bar{\Phi}_h^\gamma \leq W)$ in Line 6 of Figure 6.4.

Theorem 6.3.8. *Let $T \geq 3$. Assume that the values $UB^{[1\dots(T-1)]}[\gamma, \hat{\gamma}]$ have been computed for all $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$ such that the UB Property holds (for $(T - 1)$ rounds). Let the values $UB^{[1\dots T]}[\gamma, \hat{\gamma}]$ be computed using the algorithm in Figure 6.5. Then the UB Property holds (for T rounds).*

Proof. For this proof, “Line X ” refers to the X^{th} line in Figure 6.5. We continue to use notation from Section 6.3.1. The sequence $\{v_i^{\hat{\gamma}}\}$ will be used to bound $\{d_i\}$ in exactly the same fashion as in the case $T = 2$. Now consider the sequence $\{\dot{c}_i\}$. For any i ($1 \leq i \leq L$), $\dot{c}_i = ELP^{[1\dots(T-1)]}(\mathbf{a}, \mathbf{x})$, for some $\mathbf{x} \in \{0, 1\}^N$. By assumption, $\dot{c}_i \leq UB^{[1\dots(T-1)]}[\gamma, \gamma_{\mathbf{x}}]$. Moreover, the number of elements $\mathbf{z} \in \{0, 1\}^N$ that activate the same set of s-boxes as \mathbf{x} (i.e., for which $\gamma_{\mathbf{z}} = \gamma_{\mathbf{x}}$), and that are *not* eliminated by Observation 6.3.2, is $W[\gamma_{\mathbf{x}}, \hat{\gamma}]$. Note that for any such \mathbf{z} , if $\dot{c}_k = ELP^{[1\dots(T-1)]}(\mathbf{a}, \mathbf{z})$, then \dot{c}_k is also upper bounded by $UB^{[1\dots(T-1)]}[\gamma, \gamma_{\mathbf{x}}]$. In other words, the sequence of terms of the form $UB^{[1\dots(T-1)]}[\]$ used to upper bound $\{\dot{c}_i\}$ consists of natural groups of repeated terms.

Consider the operations in Lines 3–7. We enumerate the elements $\xi \in \{0, 1\}^M \setminus \mathbf{0}$ for which $W[\xi, \hat{\gamma}] > 0$ and $UB^{[1\dots(T-1)]}[\gamma, \xi] > 0$ as $\gamma_1, \gamma_2, \dots, \gamma_H$, such that

$$UB^{[1\dots(T-1)]}[\gamma, \gamma_1] \geq UB^{[1\dots(T-1)]}[\gamma, \gamma_2] \geq \dots \geq UB^{[1\dots(T-1)]}[\gamma, \gamma_H].$$

We then set $U_h = UB^{[1\dots(T-1)]}[\gamma, \gamma_h]$ and $W_h = W[\gamma_h, \hat{\gamma}]$, for $1 \leq h \leq H$. It follows that $\{\dot{c}_i\}_{i=1}^L$ is upper bounded in a term-by-term fashion by the first L terms of the sequence

$$\left\langle \underbrace{U_1, \dots, U_1}_{w_1 \text{ terms}}, \underbrace{U_2, \dots, U_2}_{w_2 \text{ terms}}, \dots, \underbrace{U_H, \dots, U_H}_{w_H \text{ terms}} \right\rangle. \quad (6.22)$$

In other words, the sequence in (6.22) plays the role played by $\{u_i^{\gamma}\}$ in the case $T = 2$. The remainder of Figure 6.5 almost exactly parallels Figure 6.4. We use W_{total} and

ψ as counterparts of the partial sums $\overline{\Phi}_h^\gamma$ and $\overline{\Lambda}_h^\gamma$, respectively, and the role of λ is unchanged. Finally, the function `IncSum2()` in Figure 6.4 is replaced by `IncSumT()`, which is almost identical, except that ψ is also updated inside `IncSumT()`.

As in the case $T = 2$, there is a technical situation in which the upper bound computed is strictly larger than $\sum_{i=1}^L \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}}$ (again, with $\{\tilde{u}_i^\gamma\}$ being replaced by the sequence in (6.22)). The details exactly parallel those at the end of the proof of Theorem 6.3.7. \square

6.3.4 Minimum Lengths of $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$

In the case $T = 2$, since we have eliminated all \mathbf{x} for which $\gamma_{\mathbf{x}} \neq \gamma_{\mathbf{a}}$ via Observation 6.3.3, and all \mathbf{y} for which $\gamma_{\mathbf{y}} \neq \gamma_{\mathbf{b}}$ via Observation 6.3.2, it suffices for $\{u_i^\gamma\}$ to have length $(2^n - 1)^{wt(\gamma)}$, and for $\{v_i^{\hat{\gamma}}\}$ to have length $(2^n - 1)^{wt(\hat{\gamma})}$.

In the case $T \geq 3$, only Observation 6.3.2 applies, so we require that $\{v_i^{\hat{\gamma}}\}$ have minimum length $(2^n - 1)^{wt(\hat{\gamma})}$, but there is no corresponding restriction on the length of the sequence in (6.22); we simply use these recursively computed values.

6.3.5 Complexity Analysis

The algorithms in Figure 6.4 ($T = 2$ case) and Figure 6.5 ($T \geq 3$ case) both consist of double outer For loops that cause the inner lines to be executed $(2^M - 1) \times (2^M - 1) \approx 2^{2M}$ times. The complexity of executing these inner lines (Lines 3–13 in Figure 6.4 and Lines 3–17 in Figure 6.5) depends on the numbers of distinct terms in the sequences involved. In Figure 6.4, these sequences are $\{\tilde{u}_i^\gamma\}$ and $\{\tilde{v}_i^{\hat{\gamma}}\}$, and in Figure 6.5 they are $\{\tilde{v}_i^{\hat{\gamma}}\}$ and the sequence in (6.22). Let \mathcal{H} be the maximum number of distinct terms in any of these sequences.

It follows that Lines 3–13 in Figure 6.4 have complexity $O(\mathcal{H} \log_2 \mathcal{H})$. This is because the While loop in Lines 6–8 is executed at most \mathcal{H} times, and each call to `IncSum2()` has complexity $O(\log_2 \mathcal{H})$ due to the binary search required to find the minimum in Line 15. This gives an overall complexity of $O(2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H})$ for the algorithm in Figure 6.4.

For Figure 6.5, we also obtain an overall complexity of $O(2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H})$. The inner lines (Lines 3–17) have complexity $O(\mathcal{H} \log_2 \mathcal{H})$ for two reasons: first, because of the sorting in Lines 4–5,² and second, because the While loop in Lines 10–13 is executed at most \mathcal{H} times and each call to `IncSumT()` has complexity $O(\log_2 \mathcal{H})$ due to the binary search required to find the minimum in Line 19.

Finally, since computation of the terms $UB^{[1 \dots T]}[\gamma, \hat{\gamma}]$ in Figure 6.5 requires pre-computation of the terms $UB^{[1 \dots (T-1)]}[\gamma, \hat{\gamma}]$, the complexity of executing the entire algorithm for T core rounds is

$$O(T \cdot 2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H}).$$

Remark 6.3.9. The value \mathcal{H} depends on the choice of the sequences $\{\tilde{u}_i^\gamma\}$ and $\{\tilde{v}_i^{\hat{\gamma}}\}$, as well as on the values U_h defined in Lines 3–6 in Figure 6.5. The values U_h are derived recursively, and therefore precomputation of \mathcal{H} is difficult in general.

Remark 6.3.10. The feasibility of computing the values $W[\]$ depends on the linear transformation of the SPN. In Section 7.1.1 we show that it is feasible to compute these values for the AES.

²Here we assume the use of a *comparison sort*, for which a complexity lower bound of $\Omega(n \log n)$ is known [22] (n is the number of elements being sorted).

6.4 The KMT1 Algorithm

As stated earlier, in order to complete the descriptions of the KMT1 and KMT2 algorithms, it remains to specify the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$. For KMT1, we make use of q , the maximum nontrivial LP value over all SPN s-boxes (Definition 3.3.13). In general, $0 < q \leq 1$. We will assume that $0 < q < 1$, which will be the case for any reasonably designed SPN. Given $\mathbf{a} \in \{0, 1\}^N \setminus \mathbf{0}$, let f be the number of s-boxes made active by \mathbf{a} , that is, $f = wt(\gamma_{\mathbf{a}})$. For any round t , and for any $\mathbf{x} \in \{0, 1\}^N$, it follows from (3.11) that

$$ELP^t(\mathbf{a}, \mathbf{x}) \leq q^f.$$

In general, for any $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$, let $f = wt(\gamma)$, and define $\{u_i^\gamma\}$ to be the sequence

$$\left\langle \underbrace{q^f, q^f, q^f, \dots, q^f}_{(2^n - 1)^f \text{ terms}} \right\rangle.$$

Similarly, for any $\hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, let $\ell = wt(\hat{\gamma})$, and define $\{v_i^{\hat{\gamma}}\}$ to be the sequence

$$\left\langle \underbrace{q^\ell, q^\ell, q^\ell, \dots, q^\ell}_{(2^n - 1)^\ell \text{ terms}} \right\rangle. \quad (6.23)$$

Then $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ satisfy the conditions listed in Section 6.3.1. Section 6.3.4 explains why $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ have lengths $(2^n - 1)^f$ and $(2^n - 1)^\ell$, respectively.

With these choices for $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$, KMT1 is entirely specified by the pseudocode in Figure 6.4 and Figure 6.5. However, the simple nature of these sequences allows us to restate the case $T = 2$ in a concise form in Theorem 6.4.1 below. We also restate the case $T \geq 3$ in Theorem 6.4.2.

Theorem 6.4.1. Let $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$, $f = \text{wt}(\gamma)$, $\ell = \text{wt}(\hat{\gamma})$, and $W = W[\gamma, \hat{\gamma}]$. If

$$UB^{[1\dots 2]}[\gamma, \hat{\gamma}] \stackrel{\text{def}}{=} \begin{cases} \min \{q^f, q^\ell\} & \text{if } \max \{q^f, q^\ell\} \cdot W > 1 \\ q^{f+\ell} \cdot W & \text{if } \max \{q^f, q^\ell\} \cdot W \leq 1 \end{cases}$$

then the UB Property holds (for 2 rounds).

Proof. Let $\mathbf{a}, \mathbf{b} \in \{0, 1\}^N \setminus \mathbf{0}$, and set $\gamma = \gamma_{\mathbf{a}}$, $\hat{\gamma} = \gamma_{\mathbf{b}}$. Recall that the value $UB^{[1\dots 2]}[\gamma, \hat{\gamma}]$ computed in Figure 6.4 is the right-hand side of (6.15), namely $\sum_{i=1}^L \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}}$ (here $L = W$), where $\{\tilde{u}_i^\gamma\}$ and $\{\tilde{v}_i^{\hat{\gamma}}\}$ are the possibly truncated versions of $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$, respectively. If $\max \{q^f, q^\ell\} \cdot W \leq 1$, then no truncation is required, so

$$UB^{[1\dots 2]}[\gamma, \hat{\gamma}] = \sum_{i=1}^W u_i^\gamma v_i^{\hat{\gamma}} = q^{f+\ell} \cdot W.$$

Now suppose $\max \{q^f, q^\ell\} \cdot W > 1$. It follows that at least one of the two sequences will be truncated. If $f < \ell$ (so $q^f > q^\ell$), then either $\{u_i^\gamma\}$ will be truncated and $\{v_i^{\hat{\gamma}}\}$ will not, or both will be truncated, with $\{u_i^\gamma\}$ being truncated earlier than $\{v_i^{\hat{\gamma}}\}$. Suppose there are \tilde{W} nonzero terms in the truncated version of $\{u_i^\gamma\}$. Since the first \tilde{W} terms of $\{v_i^{\hat{\gamma}}\}$ are all equal to q^ℓ , we have

$$\sum_{i=1}^W \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}} = q^\ell \sum_{i=1}^{\tilde{W}} u_i^\gamma = q^\ell \cdot 1 = q^\ell = \min \{q^f, q^\ell\} = UB^{[1\dots 2]}[\gamma, \hat{\gamma}].$$

If $f > \ell$, a symmetric argument gives

$$UB^{[1\dots 2]}[\gamma, \hat{\gamma}] = q^f = \min \{q^f, q^\ell\}.$$

If $\max \{q^f, q^\ell\} \cdot W > 1$ and $f = \ell$, then

$$\sum_{i=1}^W \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}} \leq q^\ell \sum_{i=1}^{\tilde{W}} \tilde{u}_i^\gamma = q^\ell = q^f = \min \{q^f, q^\ell\}.$$

Here the upper bound may be strictly larger than $\sum_{i=1}^W \tilde{u}_i^\gamma \tilde{v}_i^{\hat{\gamma}}$. This is exactly the contingency discussed at the end of the proof of Theorem 6.3.7. \square

Theorem 6.4.2. *Let $T \geq 3$. Assume that values $UB^{[1 \dots (T-1)]}[\gamma, \hat{\gamma}]$ have been computed for all $\gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$ such that the UB Property holds (for $(T - 1)$ rounds). Let values $UB^{[1 \dots T]}[\gamma, \hat{\gamma}]$ be computed using the algorithm in Figure 6.6. Then the UB Property holds (for T rounds).*

Proof. Here “Line X ” refers to the X^{th} line in Figure 6.6. The algorithm in Figure 6.6 follows readily from the general algorithm in Figure 6.5. Consider lines 9 and 10. Clearly $S_u > 1$ if and only if the sequence in (6.22) needs to be truncated, and $S_q > 1$ if and only if $\{v_i^{\hat{\gamma}}\}$ (as given in (6.23)) needs to be truncated. In Lines 19–26 we deal with the four possible cases involving truncation/non-truncation of these sequences. \square

6.4.1 Complexity Analysis of KMT1

For KMT1, in the case $T = 2$, computation of a single term $UB^{[1 \dots 2]}[\gamma, \hat{\gamma}]$ requires constant time using Theorem 6.4.1, and therefore computing the maximum of all such terms has complexity $O(2^{2M})$. In the case $T \geq 3$, the overall complexity is the same as in Section 6.3.5, namely $O(T \cdot 2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H})$. The reason for this unchanged complexity is that the inner lines in Figure 6.6 (Lines 3–26) have complexity $O(\mathcal{H} \log_2 \mathcal{H})$ due to the sorting step in Lines 5–6. Recall that \mathcal{H} is the maximum number of distinct terms in the sequences involved; since $\{v_i^{\hat{\gamma}}\}$ consists of a single repeated term, \mathcal{H} will be the maximum value of H as defined in Line 5 in Figure 6.6 (or Line 4 in Figure 6.5). In Section 7.1.3 we discuss the actual running time of our implementation of KMT1 as applied to the AES.

6.5 The KMT2 Algorithm

As seen above, the KMT1 algorithm makes use of q , the maximum s-box LP value. The KMT2 algorithm is based on the following observation: *the upper bound obtained by KMT1 can potentially be improved by incorporating more detailed information about the distribution of LP values for the SPN s-boxes.* The KMT2 algorithm is, in general, more computationally expensive than KMT1, but has proven to be of manageable complexity for SPNs of practical size, such as the AES (Section 7.1.4).

We construct the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ by considering how to upper bound LP values for the SPN s-boxes. The KMT2 algorithm is then entirely specified by the pseudocode in Figure 6.4 and Figure 6.5.

6.5.1 Upper Bounding LP Values for One S-Box

Let S be an $n \times n$ s-box. For fixed input mask $\alpha \in \{0, 1\}^n \setminus \mathbf{0}$, let

$$\langle e_{\alpha,1}^S, e_{\alpha,2}^S, \dots, e_{\alpha,2^n-1}^S \rangle \quad (6.24)$$

be the sequence obtained by sorting the values $\{LP^S(\alpha, \beta) : \beta \in \{0, 1\}^n \setminus \mathbf{0}\}$ in non-increasing order. For $1 \leq i \leq (2^n - 1)$, define

$$u_i^S = \max_{\alpha \in \{0,1\}^n \setminus \mathbf{0}} e_{\alpha,i}^S.$$

It follows that if $\alpha \in \{0, 1\}^n \setminus \mathbf{0}$ is a fixed input mask for S , if $\langle \chi_1, \chi_2, \dots, \chi_L \rangle$ is a sequence of distinct output masks for S , and if $c_i = LP^S(\alpha, \chi_i)$, for $1 \leq i \leq L$, then the sequence $\{\dot{c}_i\}_{i=1}^L$ obtained by sorting $\{c_i\}$ in nonincreasing order is upper bounded by $\{u_i^S\}$ in a term-by-term fashion, i.e., $\dot{c}_i \leq u_i^S$, for $1 \leq i \leq L$. Let \bar{D}^S be the number of distinct terms in $\{u_i^S\}$, and let

$$\langle \bar{\rho}_1^S, \bar{\rho}_2^S, \dots, \bar{\rho}_{\bar{D}^S}^S \rangle$$

be the sequence of these distinct terms sorted in decreasing order. Finally, define $\bar{\phi}_j^S$ to be the frequency of occurrence of $\bar{\rho}_j^S$ in $\{u_i^S\}$, for $1 \leq j \leq \bar{D}^S$ (it follows trivially that $\bar{\phi}_1^S + \bar{\phi}_2^S + \dots + \bar{\phi}_{\bar{D}^S}^S = (2^n - 1)$). Then $\{u_i^S\}$ is the sequence

$$\left\langle \underbrace{\bar{\rho}_1^S, \dots, \bar{\rho}_1^S}_{\bar{\phi}_1^S \text{ terms}}, \underbrace{\bar{\rho}_2^S, \dots, \bar{\rho}_2^S}_{\bar{\phi}_2^S \text{ terms}}, \dots, \underbrace{\bar{\rho}_{\bar{D}^S}^S, \dots, \bar{\rho}_{\bar{D}^S}^S}_{\bar{\phi}_{\bar{D}^S}^S \text{ terms}} \right\rangle. \quad (6.25)$$

In the above development, if we begin instead with fixed *output* masks, we obtain the corresponding values \underline{D}^S , $\underline{\rho}_j^S$, and $\underline{\phi}_j^S$ ($1 \leq j \leq \underline{D}^S$), and we define $\{v_i^S\}$ to be the sequence

$$\left\langle \underbrace{\underline{\rho}_1^S, \dots, \underline{\rho}_1^S}_{\underline{\phi}_1^S \text{ terms}}, \underbrace{\underline{\rho}_2^S, \dots, \underline{\rho}_2^S}_{\underline{\phi}_2^S \text{ terms}}, \dots, \underbrace{\underline{\rho}_{\underline{D}^S}^S, \dots, \underline{\rho}_{\underline{D}^S}^S}_{\underline{\phi}_{\underline{D}^S}^S \text{ terms}} \right\rangle. \quad (6.26)$$

Remark 6.5.1. It is easy to see that $u_i \leq q$ and $v_i \leq q$, for all i . This is the reason that KMT2 produces a tighter upper bound than KMT1 in general.

6.5.2 Upper Bounding ELP Values for One Round

In this section we extend the sequences $\{u_i^S\}$ and $\{v_i^S\}$ to apply to multiple s-boxes. Enumerate the s-boxes in a substitution stage from left to right as S_1, S_2, \dots, S_M . Let $\gamma = \gamma_1 \gamma_2 \dots \gamma_M \in \{0, 1\}^M \setminus \mathbf{0}$ represent a pattern of active s-boxes, and let $A = wt(\gamma)$. Denote the indices of the nonzero γ_m (i.e., the active s-boxes) by m_1, m_2, \dots, m_A . Consider all terms of the form

$$\prod_{a=1}^A \bar{\rho}_{r_a}^{S_{m_a}}, \quad (6.27)$$

where r_a is free to range over $\{1, 2, \dots, \bar{D}^{S_{m_a}}\}$. Let \bar{D}^γ be the number of distinct terms of this form, and let

$$\left\langle \bar{\rho}_1^\gamma, \bar{\rho}_2^\gamma, \dots, \bar{\rho}_{\bar{D}^\gamma}^\gamma \right\rangle$$

be the sequence of these distinct terms sorted in decreasing order. For $1 \leq j \leq \overline{D}^\gamma$, define $\overline{\phi}_j^\gamma$ to be the frequency of occurrence of $\overline{\rho}_j^\gamma$ in the formation of the terms in (6.27), and define $\{u_i^\gamma\}$ to be the sequence

$$\left\langle \underbrace{\overline{\rho}_1^\gamma, \dots, \overline{\rho}_1^\gamma}_{\overline{\phi}_1^\gamma \text{ terms}}, \underbrace{\overline{\rho}_2^\gamma, \dots, \overline{\rho}_2^\gamma}_{\overline{\phi}_2^\gamma \text{ terms}}, \dots, \underbrace{\overline{\rho}_{\overline{D}^\gamma}^\gamma, \dots, \overline{\rho}_{\overline{D}^\gamma}^\gamma}_{\overline{\phi}_{\overline{D}^\gamma}^\gamma \text{ terms}} \right\rangle. \quad (6.28)$$

It follows that if \mathbf{a} is a fixed input mask for round t ($1 \leq t \leq T$), and if $\{\hat{c}_i\}$ consists of terms of the form $ELP^t(\mathbf{a}, \mathbf{x})$ sorted in nonincreasing order (with each such term based on a distinct value of \mathbf{x}), then $\{u_i^\gamma\}$ upper bounds $\{\hat{c}_i\}$ in a term-by-term fashion. (Notice that we have switched from the LP values in Section 6.5.1 to ELP values by making use of (3.11).)

The construction of the sequence $\{v_i^{\hat{\gamma}}\}$ is the counterpart of the above. Let $\hat{\gamma} = \hat{\gamma}_1 \hat{\gamma}_2 \dots \hat{\gamma}_M \in \{0, 1\}^M \setminus \mathbf{0}$. If $\hat{A} = wt(\hat{\gamma})$ and the indices of the nonzero $\hat{\gamma}_m$ are $m_1, m_2, \dots, m_{\hat{A}}$, then we consider all terms of the form

$$\prod_{a=1}^{\hat{A}} \rho_{r_a}^{S_{m_a}}. \quad (6.29)$$

We define $\underline{D}^{\hat{\gamma}}$, $\underline{\rho}_j^{\hat{\gamma}}$, and $\underline{\phi}_j^{\hat{\gamma}}$ to be the counterparts of \overline{D}^γ , $\overline{\rho}_j^\gamma$, and $\overline{\phi}_j^\gamma$, respectively, and $\{v_i^{\hat{\gamma}}\}$ is defined to be the sequence

$$\left\langle \underbrace{\underline{\rho}_1^{\hat{\gamma}}, \dots, \underline{\rho}_1^{\hat{\gamma}}}_{\underline{\phi}_1^{\hat{\gamma}} \text{ terms}}, \underbrace{\underline{\rho}_2^{\hat{\gamma}}, \dots, \underline{\rho}_2^{\hat{\gamma}}}_{\underline{\phi}_2^{\hat{\gamma}} \text{ terms}}, \dots, \underbrace{\underline{\rho}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}}, \dots, \underline{\rho}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}}}_{\underline{\phi}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}} \text{ terms}} \right\rangle. \quad (6.30)$$

It is not hard to see that

$$\begin{aligned} \overline{\phi}_1^\gamma + \overline{\phi}_2^\gamma + \dots + \overline{\phi}_{\overline{D}^\gamma}^\gamma &= (2^n - 1)^A = (2^n - 1)^{wt(\gamma)} \\ \underline{\phi}_1^{\hat{\gamma}} + \underline{\phi}_2^{\hat{\gamma}} + \dots + \underline{\phi}_{\underline{D}^{\hat{\gamma}}}^{\hat{\gamma}} &= (2^n - 1)^{\hat{A}} = (2^n - 1)^{wt(\hat{\gamma})}, \end{aligned}$$

satisfying the minimum lengths of $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ as given in Section 6.3.4.

Remark 6.5.2. The complexity of computing the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ depends to a large extent on the numbers of distinct terms in the s-box-level sequences $\{u_i^{S_m}\}$ and $\{v_i^{S_m}\}$, respectively. It also depends, of course, on M , the number of s-boxes per round. In the worst case, there are $(2^n - 1)^M$ distinct terms of the form given in (6.27) or (6.29). However, typically there will be repeated terms in the sequences $\{u_i^{S_m}\}$ and $\{v_i^{S_m}\}$, greatly simplifying computation. (For the AES, this simplification is particularly pronounced—see Section 7.1.4.)

6.5.3 Complexity Analysis of KMT2

It follows directly from Section 6.3.5 that the complexity of KMT2 for T core rounds is $O(T \cdot 2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H})$. In Section 7.1.4 we discuss the actual running time of our implementation of KMT2 as applied to the AES.

6.6 Summary

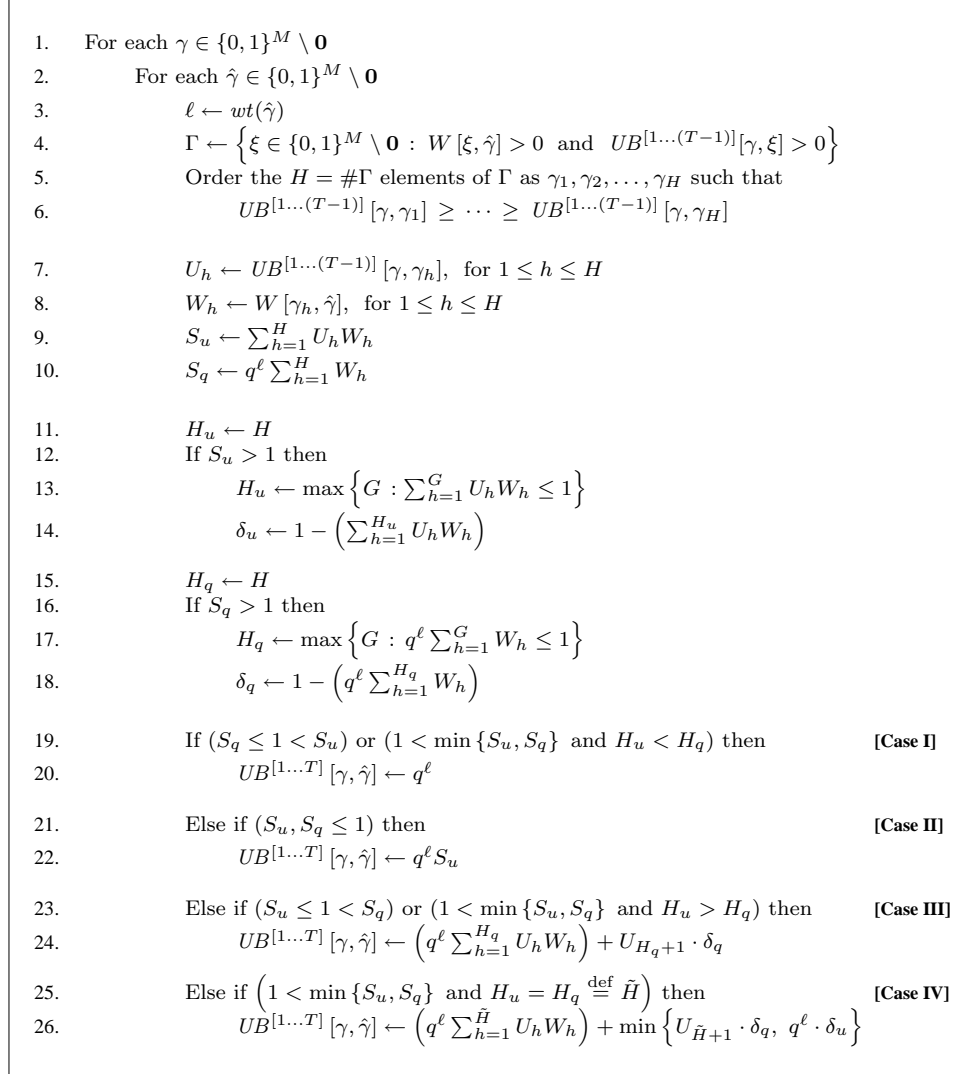
We have presented two new algorithms for computing an upper bound on the maximum average linear hull probability (MALHP) for SPNs. These algorithms are denoted KMT1 and KMT2, and they are, to our knowledge, the first completely general algorithms for this purpose. KMT2 is a refinement of KMT1; in general, KMT2 provides a tighter upper bound but is more computationally intensive. We developed the structure that is common to these two algorithms, and proved the correctness of our approach. We then specified the parameters particular to each of KMT1 and KMT2.

1.	For each $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$
2.	For each $\hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$
3.	$W \leftarrow W[\gamma, \hat{\gamma}]$
4.	$\lambda \leftarrow 0, \text{ Sum} \leftarrow 0$
5.	$h \leftarrow 1$
6.	While $(h \leq \overline{D}^\gamma)$ and $(\overline{\rho}_h^\gamma > 0)$ and $(\overline{\Phi}_h^\gamma \leq W)$ and $(\overline{\Lambda}_h^\gamma \leq 1)$ and $(\lambda < 1)$
7.	$\text{Sum} \leftarrow \text{Sum} + \text{IncSum2}(\overline{\Phi}_h^\gamma)$
8.	$h \leftarrow h + 1$
9.	If $(h \leq \overline{D}^\gamma)$ and $(\overline{\rho}_h^\gamma > 0)$ and $[(\overline{\Phi}_h^\gamma > W) \text{ or } (\overline{\Lambda}_h^\gamma > 1)]$ and $(\lambda < 1)$
10.	$W' \leftarrow \overline{\Phi}_{h-1}^\gamma + (1 - \overline{\Lambda}_{h-1}^\gamma) / \overline{\rho}_h^\gamma$
11.	$W_{\text{end}} \leftarrow \min \{W, W'\}$
12.	$\text{Sum} \leftarrow \text{Sum} + \text{IncSum2}(W_{\text{end}})$
13.	$UB^{[1 \dots 2]}[\gamma, \hat{\gamma}] \leftarrow \text{Sum}$
14.	Function $\text{IncSum2}(Z)$
15.	$J \leftarrow \min \{j : 1 \leq j \leq \underline{D}^{\hat{\gamma}}, \underline{\Phi}_j^{\hat{\gamma}} \geq Z\}$
16.	$\Delta\lambda \leftarrow (\underline{\Delta}_J^{\hat{\gamma}} - \lambda) - [(\underline{\Phi}_J^{\hat{\gamma}} - Z) * \underline{\rho}_J^{\hat{\gamma}}]$
17.	If $(\lambda + \Delta\lambda) > 1$
18.	$\Delta\lambda \leftarrow 1 - \lambda$
19.	$\lambda \leftarrow \lambda + \Delta\lambda$
20.	return $(\overline{\rho}_h^\gamma * \Delta\lambda)$

Figure 6.4: General algorithm for upper bounding the MALHP ($T = 2$ case)

<ol style="list-style-type: none"> 1. For each $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$ 2. For each $\hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}$ 3. $\Gamma \leftarrow \left\{ \xi \in \{0, 1\}^M \setminus \mathbf{0} : W[\xi, \hat{\gamma}] > 0 \text{ and } UB^{[1 \dots (T-1)]}[\gamma, \xi] > 0 \right\}$ 4. Order the $H = \#\Gamma$ elements of Γ as $\gamma_1, \gamma_2, \dots, \gamma_H$ such that 5. $UB^{[1 \dots (T-1)]}[\gamma, \gamma_1] \geq \dots \geq UB^{[1 \dots (T-1)]}[\gamma, \gamma_H]$ 6. $U_h \leftarrow UB^{[1 \dots (T-1)]}[\gamma, \gamma_h]$, for $1 \leq h \leq H$ 7. $W_h \leftarrow W[\gamma_h, \hat{\gamma}]$, for $1 \leq h \leq H$ 8. $\psi \leftarrow 0$, $\lambda \leftarrow 0$, $W_{\text{total}} \leftarrow 0$, $\text{Sum} \leftarrow 0$ 9. $h \leftarrow 1$ 10. While $(h \leq H)$ and $(\psi + (U_h * W_h) \leq 1)$ and $(\lambda < 1)$ 11. $W_{\text{total}} \leftarrow W_{\text{total}} + W_h$ 12. $\text{Sum} \leftarrow \text{Sum} + \text{IncSumT}(W_{\text{total}})$ 13. $h \leftarrow h + 1$ 14. If $(h \leq H)$ and $(\psi + (U_h * W_h) > 1)$ and $(\lambda < 1)$ 15. $W_{\text{total}} \leftarrow W_{\text{total}} + (1 - \psi)/U_h$ 16. $\text{Sum} \leftarrow \text{Sum} + \text{IncSumT}(W_{\text{total}})$ 17. $UB^{[1 \dots T]}[\gamma, \hat{\gamma}] \leftarrow \text{Sum}$ 	<ol style="list-style-type: none"> 18. Function $\text{IncSumT}(Z)$ 19. $J \leftarrow \min \left\{ j : 1 \leq j \leq D^{\hat{\gamma}}, \Phi_j^{\hat{\gamma}} \geq Z \right\}$ 20. $\Delta\lambda \leftarrow \left(\Delta_J^{\hat{\gamma}} - \lambda \right) - \left[\left(\Phi_J^{\hat{\gamma}} - Z \right) * \rho_J^{\hat{\gamma}} \right]$ 21. $\psi \leftarrow \psi + (U_h * W_h)$ 22. If $(\lambda + \Delta\lambda) > 1$ 23. $\Delta\lambda \leftarrow 1 - \lambda$ 24. $\lambda \leftarrow \lambda + \Delta\lambda$ 25. return $(U_h * \Delta\lambda)$
--	---

Figure 6.5: General algorithm for upper bounding the MALHP ($T \geq 3$ case)

Figure 6.6: KMT1 Algorithm ($T \geq 3$ case)

Chapter 7

Analysis of Specific SPN Ciphers

In this chapter we discuss the analysis of specific SPN ciphers with respect to linear cryptanalysis. In Section 7.1 we describe the application of the KMT1 and KMT2 algorithms from Chapter 6 to the AES. In Section 7.2 we explain our use of linear hulls to break the Q cipher.

7.1 Application of KMT1/KMT2 to the AES

In Section 6.1 we summarized the results of applying the KMT1 and KMT2 algorithms to the AES. In this section we give details concerning the actual execution of these algorithms, focusing on important data structures and on the computational effort required. We start with considerations common to both KMT1 and KMT2. Note that much of the material in this section applies to any AES-like SPN (Section 2.5.2).

7.1.1 Computation of $W[\]$ Entries

An important element of the KMT1 and KMT2 algorithms is the $W[\]$ table (Definition 3.3.10). For the AES, $W[\]$ has dimensions $2^{16} \times 2^{16}$. Direct computation of $W[\]$ from the definition is infeasible, since it requires in the order of 2^{128} operations. In addition, storage for the 2^{32} entries in $W[\]$ would be 32GB, assuming 8 bytes per entry (not a problem for disk space, but still a challenge at the time of this writing if we want to store the entire table in RAM). We can significantly reduce the space requirement for $W[\]$ by storing only the nonzero entries (more on this below). And we can significantly reduce the time requirement for computing $W[\]$ by making use of the corresponding table for the 32-bit maximally diffusive linear transformation component of the AES linear transformation (see Figure 2.6); let $W^{32}[\]$ denote this $2^4 \times 2^4$ table. Direct computation of $W^{32}[\]$ requires in the order of 2^{32} operations (feasible), and storage is negligible.

For $T = 2$ core rounds, both KMT1 and KMT2 loop through all $\gamma, \hat{\gamma} \in \{0, 1\}^{16} \setminus \mathbf{0}$ (a total of approximately 2^{32} iterations), making use of each value $W[\gamma, \hat{\gamma}]$. For a given pair $\gamma, \hat{\gamma}$, permute the 16 bits of γ in exactly the same fashion as the first step in the AES linear transformation permutes the 16 bytes of its input (Figure 2.6). Partition the resulting 16-bit vector from left to right into $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \{0, 1\}^4$, and similarly partition $\hat{\gamma}$ into $\hat{\gamma}_1, \hat{\gamma}_2, \hat{\gamma}_3, \hat{\gamma}_4 \in \{0, 1\}^4$. It follows that

$$W[\gamma, \hat{\gamma}] = \prod_{i=1}^4 W^{32}[\gamma_i, \hat{\gamma}_i]. \quad (7.1)$$

By using (7.1), we can compute the entire $W[\]$ table in the order of 2^{32} operations (as opposed to 2^{128} operations).

For $T \geq 3$, KMT1 and KMT2 access the entries of $W[\]$ in the following manner: for each nonzero output mask $\hat{\gamma}$, it is necessary to find all the nonzero input masks γ

such that $W[\gamma, \hat{\gamma}] > 0$. To avoid wasting time accessing zero entries in $W[\]$, we precompute and store the list of all such γ for each $\hat{\gamma}$. The average length of these lists is 1191 [58], for a total of approximately 78,000,000 entries (so roughly 2% of the entries in $W[\]$ are nonzero). Since each entry can be stored using a 16-bit value, we require approximately 150MB of storage. For this reason, our implementations of KMT1 and KMT2 perform best on computers with at least this amount of free RAM (on machines with less RAM, disk swapping seriously impedes performance).

7.1.2 Parallel/Distributed Processing

Both KMT1 and KMT2 are amenable to parallel or distributed computing, independent of the SPN to which they are applied. To see this, consider Figure 6.4 and Figure 6.5, which give the algorithmic structure common to KMT1 and KMT2. For fixed $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$, the values $UB^{[1\dots 2]}[\gamma, \cdot]$ do not depend on any other values $UB^{[1\dots 2]}[\cdot, \cdot]$; and for $T \geq 3$, the values $UB^{[1\dots T]}[\gamma, \cdot]$ depend only on previously computed values of the form $UB^{[1\dots (T-1)]}[\gamma, \cdot]$. Therefore, we can parallelize/distribute the outermost For loop in Figure 6.4 and Figure 6.5, with linear speedup (i.e., using ω processors increases performance by a factor of ω).

7.1.3 Considerations Specific to KMT1

Consider the complexity analysis of KMT1 in Section 6.4.1. It follows from the discussion of $W[\]$ in Section 7.1.1 that the *average* value of H is upper bounded by 1191. As a first approximation to the number of operations involved in applying KMT1 to the AES, we can substitute this value for \mathcal{H} in the expression $(T \cdot 2^{2M} \cdot \mathcal{H} \log_2 \mathcal{H})$,

i		1	2	3	4	5	6	7	8	9
ρ_i		$(\frac{8}{64})^2$	$(\frac{7}{64})^2$	$(\frac{6}{64})^2$	$(\frac{5}{64})^2$	$(\frac{4}{64})^2$	$(\frac{3}{64})^2$	$(\frac{2}{64})^2$	$(\frac{1}{64})^2$	0
ϕ_i		5	16	36	24	34	40	36	48	17

Table 7.1: Distribution of LP values for the AES s-box

together with $M = 16$ and $T = 16$. This gives

$$16 \cdot 2^{32} \cdot 1191 \cdot \log_2 1191 \approx 10^{15} \approx 2^{50}.$$

This is a large but feasible number of operations. On a 2.8 GHz Pentium 4 (Dell Optiplex GX260, Red Hat Linux 9, Intel C++ Compiler 7.1), our implementation of KMT1 as applied to the AES requires roughly 2400 hours of computation time. For the results published in [58], our benchmark machine was a Sun Ultra 5, for which the estimated total running time was 44,000 hours. We completed the computation by distributing it to approximately 60 CPUs for several weeks.

7.1.4 Considerations Specific to KMT2

In order to apply KMT2 to the AES, it is necessary to derive the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ as specified in Section 6.5. The general situation is greatly simplified by the fact that the AES uses a single repeated s-box, and by the fact that the table of LP values for this s-box is highly structured. In fact, all the nontrivial rows and columns of this LP table have the same distribution of entries, given in Table 7.1 [60] (ρ_j denotes a distinct LP value, and ϕ_j denotes the frequency of occurrence of ρ_j). It follows that the sequences $\{u_i^S\}$ and $\{v_i^S\}$ as given in (6.25) and (6.26), respectively, are identical, and are exactly specified by Table 7.1 ($\overline{D}^S = \underline{D}^S = 9$).

To compute the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ as given in (6.28) and (6.30), respectively, we are essentially making use of the following. Let $1 \leq m \leq M$, and let $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_m$ be independent random variables distributed according to Table 7.1 (convert ϕ_j to the probability $\frac{\phi_j}{256}$). Define $\mathbf{Z}^m = \mathbf{Z}_1 \cdot \mathbf{Z}_2 \cdots \mathbf{Z}_m$. Then $\{u_i^\gamma\}$ ($\{v_i^{\hat{\gamma}}\}$) has the same distribution as $\mathbf{Z}^{wt(\gamma)}$ ($\mathbf{Z}^{wt(\hat{\gamma})}$); the only difference is that for $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ we are interested in the frequency with which distinct terms occur, namely the values $\overline{\phi}_j^\gamma$ and $\underline{\phi}_j^{\hat{\gamma}}$, respectively, not in the corresponding probabilities, namely $\left(\frac{\overline{\phi}_j^\gamma}{256^{wt(\gamma)}}\right)$ and $\left(\frac{\underline{\phi}_j^{\hat{\gamma}}}{256^{wt(\hat{\gamma})}}\right)$, respectively. It is important to note that $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ depend only on $wt(\gamma)$ and $wt(\hat{\gamma})$, not on the specific values of γ and $\hat{\gamma}$, and that $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ are identical when $wt(\gamma) = wt(\hat{\gamma})$. Therefore, we compute one sequence for each value of $wt(\gamma)$ ($1 \leq wt(\gamma) \leq 16$). Actual computation is straightforward.

Because of the complexity of the sequences used in KMT2 relative to KMT1, there is a significant increase in execution time (the average number of distinct terms in $\{u_i^\gamma\}$ over all $\gamma \in \{0, 1\}^M \setminus \mathbf{0}$ is 1873; the corresponding sequences for KMT1 consist of a single repeated term). On our benchmark 2.8 GHz Pentium 4 (Section 7.1.3), our implementation of KMT2 as applied to the AES requires approximately 12,000 hours of computation time ($5 \times$ the running time of KMT1). For the results published in [60], our benchmark machine was a Sun Ultra 5, for which the estimated total running time was 200,000 hours. We completed this computation by distributing it to approximately 50 CPUs for several months.

7.2 Linear Cryptanalysis of Q

Q is a block cipher submitted to the NESSIE project by Leslie McBride [79]. Q is a straightforward SPN, only deviating from the SPN architecture as given in Section 2.4.2 in the fact that s-boxes of different sizes are used. We show that the structure of the s-boxes and linear transformations in Q allows the construction of linear characteristics with relatively large ELCP values. By combining many such characteristics into a linear hull, we demonstrate the existence of large ELP values for Q. The best ELP value determined by our method is $2^{-90.1}$; this constitutes a theoretical break of the Q cipher (Section 2.7.1). To our knowledge, this is the first use of linear hulls to break a proposed cipher. This work was published in [61].

7.2.1 Basic Components of Q

Q has a block size of $N = 128$ bits. Q uses three different s-boxes, one 8×8 s-box named S (this is the AES s-box [25]), and two 4×4 s-boxes named A and B (B is used in Serpent[4], and A is “Serpent-like”). (Note that A and B can be implemented with an efficient bit-slicing technique [4]; for clarity, we will use the equivalent representation that involves bitwise permutations before and after application of these s-boxes.) Each substitution stage applies multiple copies of a single s-box to the 128-bit input (16 copies of S , or 32 copies of A or B).

Before continuing, we need to clarify the convention used in Q for numbering consecutive bytes and words, namely that numbering begins at 0 with the object in the lowest memory location—this is also the least significant object, since Q uses “little-endian” ordering. This convention extends to numbering the bits of bytes/nibbles, i.e., the least significant bit is numbered 0. Pictorially, numbering always increases

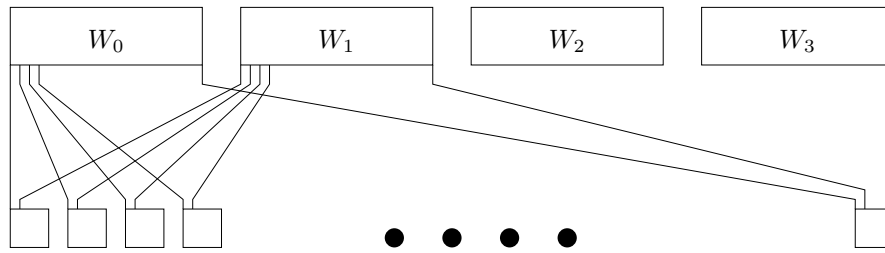


Figure 7.1: PreSerpent() bitwise permutation

from left to right (it follows that the bits in a 128-bit block are numbered $0 \dots 127$, left to right).

Three linear transformations are used in Q . The permutation P operates on a 128-bit block represented as four 32-bit words, W_0, W_1, W_2, W_3 , as follows: W_0 is unchanged; W_1, W_2 , and W_3 are rotated to the right by one byte, two bytes, and three bytes, respectively. The other two linear transformations are bitwise permutations that we term PreSerpent() and PostSerpent(), since they are positioned immediately before and after each application of s-boxes A and B . If we again view a 128-bit block as consisting of words W_0, W_1, W_2, W_3 , PreSerpent() sends the bits of W_0 to the first (leftmost) input bits of the 32 repeated 4×4 s-boxes, the bits of W_1 to the second input bits of these s-boxes, and so on. This is represented in Figure 7.1. PostSerpent() is simply the inverse of PreSerpent().

7.2.2 High-Level Structure of Q

Q accepts keys of any length, although keys longer than 256 bits are shortened to 256 bits. We will consider the version of Q that consists of 8 “full rounds” and uses a key of at most 128 bits (our attack works for any key length, but keys longer than 128

bits make the analysis slightly more complicated). McBride also proposed a 9-round version of Q for “high security applications” [79]. The Q key-scheduling algorithm generates twelve 128-bit subkeys:

$$KW1, KA, KB, K0, K1, \dots, K7, KW2.$$

Only two details of the key-scheduling algorithm are important for our attack. First, although KA and KB are 128 bits in length, each contains only 32 bits of information, since each is the concatenation of four 32-bit words, any two of which are rotations of each other. Second, the key-scheduling algorithm is reversible—given any subkey other than KA or KB , it is easy to determine the remaining subkeys [79].

For $0 \leq r \leq 7$, full round r has the structure in Figure 7.2.¹ Note that a full round actually contains three substitution stages (S , A , and B), and therefore would be considered three rounds according to Section 2.4.2. The entire cipher is described by:

$$\oplus KW1, \text{ Round0}, \dots, \text{ Round7}, [\oplus KA, \text{ Substitution}(S), \oplus KB], \oplus KW2 .$$

It follows that the 8-round version of Q contains a total of 25 substitution stages. The application of subkeys KA and KB before and after substitution with S can be viewed as making S key-dependent (from this perspective, $KW1$ and $KW2$ are used for whitening (Section 2.4.1)). However, Q also conforms to the standard SPN structure in which a single subkey is XOR’d before each fixed substitution stage; this is easily seen by combining pairs of subkeys that are applied in immediate succession, e.g., $KW1$ and KA .

¹This figure is taken from [79]; however, it disagrees slightly with the test code included in McBride’s NESSIE submission, which specifies that permutation P should be applied *before* application of subkey Kr [43].

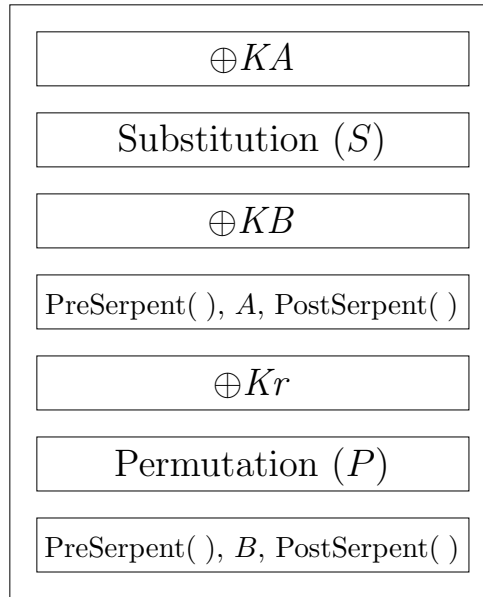


Figure 7.2: Structure of a full round of Q

7.2.3 Select LP Values for the Q S-Boxes

In what follows, we will focus on LP values for the s-boxes of Q corresponding to masks containing a single 1. We give these values in Tables 7.2, 7.3, and 7.4. Entry $[i, j]$ is the LP value for input (output) mask with 1 in position i (j) and all other bits equal to 0. (Recall that we number bits from left to right starting at 0, with 0 indicating least significance.) For S , we denote this entry $LP_S[i, j]$ (entries for A and B are subscripted accordingly).

7.2.4 High Probability Linear Hulls in Q

McBride performs preliminary linear cryptanalysis of Q by considering consistent characteristics in which each 128-bit mask contains a single 1—we call these *restricted characteristics*. A restricted characteristic is guaranteed to activate exactly

	0	1	2	3	4	5	6	7
0	$(\frac{6}{64})^2$	0	$(\frac{7}{64})^2$	$(\frac{6}{64})^2$	$(\frac{4}{64})^2$	$(\frac{2}{64})^2$	$(\frac{2}{64})^2$	$(\frac{6}{64})^2$
1	$(\frac{1}{64})^2$	$(\frac{4}{64})^2$	$(\frac{1}{64})^2$	$(\frac{3}{64})^2$	$(\frac{1}{64})^2$	$(\frac{4}{64})^2$	$(\frac{8}{64})^2$	$(\frac{1}{64})^2$
2	$(\frac{4}{64})^2$	$(\frac{1}{64})^2$	$(\frac{3}{64})^2$	$(\frac{3}{64})^2$	$(\frac{6}{64})^2$	$(\frac{8}{64})^2$	$(\frac{1}{64})^2$	$(\frac{1}{64})^2$
3	$(\frac{1}{64})^2$	$(\frac{1}{64})^2$	$(\frac{2}{64})^2$	0	$(\frac{6}{64})^2$	$(\frac{3}{64})^2$	$(\frac{1}{64})^2$	$(\frac{2}{64})^2$
4	$(\frac{6}{64})^2$	$(\frac{1}{64})^2$	$(\frac{3}{64})^2$	$(\frac{1}{64})^2$	$(\frac{4}{64})^2$	$(\frac{5}{64})^2$	0	$(\frac{4}{64})^2$
5	$(\frac{3}{64})^2$	$(\frac{5}{64})^2$	$(\frac{1}{64})^2$	$(\frac{6}{64})^2$	$(\frac{1}{64})^2$	0	$(\frac{4}{64})^2$	$(\frac{6}{64})^2$
6	$(\frac{2}{64})^2$	$(\frac{2}{64})^2$	$(\frac{6}{64})^2$	$(\frac{8}{64})^2$	$(\frac{3}{64})^2$	$(\frac{4}{64})^2$	$(\frac{6}{64})^2$	$(\frac{2}{64})^2$
7	$(\frac{6}{64})^2$	$(\frac{6}{64})^2$	$(\frac{8}{64})^2$	$(\frac{7}{64})^2$	$(\frac{4}{64})^2$	$(\frac{6}{64})^2$	$(\frac{2}{64})^2$	$(\frac{2}{64})^2$

Table 7.2: LP values for s-box S

	0	1	2	3
0	0	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
1	0	0	$\frac{1}{16}$	0
2	0	$\frac{1}{16}$	0	$\frac{1}{16}$
3	0	0	0	0

Table 7.3: LP values for s-box A

	0	1	2	3
0	$\frac{1}{16}$	0	0	0
1	$\frac{1}{16}$	$\frac{1}{16}$	0	$\frac{1}{16}$
2	0	$\frac{1}{16}$	$\frac{1}{16}$	0
3	0	0	$\frac{1}{16}$	$\frac{1}{16}$

Table 7.4: LP values for s-box B

one s-box in each substitution stage, and therefore, in general, restricted characteristics will yield the largest ELCP values among all consistent characteristics (this follows from (3.7) and (3.11)). McBride gives a partial argument that the best restricted characteristic has an ELCP value in the range of 2^{-118} [79]; this corresponds to a data complexity of 2^{123} for a 96.7% success rate (Table 3.1).²

However, as we show below, it is straightforward to combine all restricted characteristics with the same first and last masks into a linear hull for which the resulting ELP value is significantly higher than 2^{-118} (in the range of 2^{-90}), i.e., *the linear hull effect is significant for Q*.

Remark 7.2.1. Technically, we are building *sub-linear hulls*, since we are including only a subset of all the characteristics belonging to a particular linear hull. We will continue to use the term *linear hull*, since this will not be a source of confusion.

In order to form linear hulls over T core substitution stages, our algorithm uses a 3-dimensional data structure $\text{DS}[\]$ of size $128 \times (T + 1) \times 128$, in which each entry is a record of two values: an integer *Count*, and a floating-point value *ELP*. After running the algorithm, $\text{DS}[i, t, j]$ contains information about the linear hull over substitution stages $1 \dots t$ whose first mask contains a single 1 in position i , and whose last mask contains a single 1 in position j —specifically, the *Count* field is the number of restricted characteristics in the linear hull with *nonzero* ELCP values,³ and the *ELP* field is the sum of the ELCP values of those restricted characteristics.

For our attack on Q, we strip off the first and last S -substitution stages, so $T = 23$ (our presentation is easily generalized to any number of core substitution stages).

²McBride estimates a *bias* value of 2^{-60} , which is equivalent to an ELCP value of 2^{-118} .

³A characteristic with a *zero* ELCP value makes no contribution to the ELP value of the corresponding linear hull, so we omit all such characteristics.

Thus the values that are important to us will be stored in the entries $DS[i, 23, j]$. Our algorithm is given in Figure 7.3 and Figure 7.4. (The pseudocode for subroutine $\text{ApplyB}()$ is omitted, as it is symmetric to that for $\text{ApplyA}()$ —simply replace LP_A with LP_B . Also, the pseudocode for subroutine $\text{PostSerpent}()$ is omitted, as it is simply the inverse of $\text{PreSerpent}()$. Note that we use the shorthand $x += y$ to mean $x \leftarrow x + y$.)

Theorem 7.2.2. *If $DS[\]$ is filled using the algorithm in Figures 7.3 and 7.4, then for $0 \leq i, j \leq 127$ and $0 \leq t \leq 23$, $DS[i, t, j].\text{Count}$ is the number of restricted characteristics over the first t of the 23 substitution stages whose first (last) mask contains a single 1 in position i (j) and for which $ELCP > 0$, and $DS[i, t, j].\text{ELP}$ is the sum of the $ELCP$ values of these characteristics.*

Proof. Let $0 \leq i \leq 127$ be fixed. The theorem is easily proven using induction on t . Trivially, the base case ($t = 0$) is made true for all j by the initialization statement and the first For loop in the main program (Figure 7.3). We assume that the statement of the theorem holds for some $t \geq 0$, and we demonstrate its truth for $(t + 1)$. Note that the truth of the statement is not affected by the linear transformations (bitwise permutations) in Q ; we simply perform the “bookkeeping” of permuting the elements of $DS[i, t, \cdot]$ accordingly.⁴ Therefore, we need only consider the effect of the $(t + 1)^{\text{st}}$ substitution stage on $DS[i, t, \cdot]$. Without loss of generality we will limit our consideration to substitution using S ; further, without loss of generality we will consider only the effect of the first (leftmost) copy of S , denoted S_0 . The

⁴This works because given a mask *before* a linear transformation in Q , the corresponding mask *after* the linear transformation is obtained by applying the linear transformation to the mask. And this is true for Q because all linear transformations are bitwise permutations. However, this does not hold in general—for an arbitrary linear transformation represented as a binary matrix, *output* masks are transformed to *input* masks via multiplication by the *transpose* of the linear transformation (Lemma 3.3.4).

inputs/outputs for S_0 are bits $0 \dots 7$ of the relevant 128-bit blocks.

Let $\tilde{j} \in \{0, \dots, 7\}$. Consider all restricted characteristics over the first $(t + 1)$ substitution stages whose first mask has a single 1 in position i , and whose last mask has a single 1 in position \tilde{j} . Clearly all these characteristics make S_0 active. Therefore, in any of these characteristics, the mask *preceding* the $(t + 1)^{\text{st}}$ substitution stage must have the position of its single 1 in the range $0 \dots 7$. It follows that

$$\text{DS}[i, t + 1, \tilde{j}].\text{Count} = \sum_{j=0}^7 \text{DS}[i, t, j].\text{Count}, \quad (7.2)$$

with one proviso: if $\text{LP}_S[j, \tilde{j}] = 0$, then extending any t -stage restricted characteristic enumerated by $\text{DS}[i, t, j].\text{Count}$ (for $0 \leq j \leq 7$) to $(t + 1)$ stages will produce a *zero* ELCP value ((3.7) and (3.11)). Therefore, we omit all such j by modifying (7.2) as follows:

$$\text{DS}[i, t + 1, \tilde{j}].\text{Count} = \sum_{\substack{0 \leq j \leq 7 \\ \text{LP}_S[j, \tilde{j}] \neq 0}} \text{DS}[i, t, j].\text{Count} \quad (7.3)$$

(this is done via the If statement in subroutine `ApplyS()`). It is easily seen that $\text{DS}[i, t + 1, \tilde{j}].\text{ELP}$ is correctly assigned the sum of the ELCP values of all the characteristics enumerated by (7.3). □

7.2.5 Computational Results

We ran the algorithm described above for various combinations of full and partial rounds of Q (the main program in Figure 7.3 was modified appropriately). The best ELP values found are listed in Table 7.5. As a point of contrast, we include Table 7.6, which contains the corresponding best expected differential probability (EDP) values (Section 2.6.2) for Q, due to Biham et al. [15]. For a given number of rounds, ELP and

EDP values are roughly comparable in their significance. The ELP values represent a minimum improvement in the exponent of approximately 17 relative to the EDP values; the improvement in the exponent is 20.4 for the case that we will use to break the cipher, namely 7 full rounds with $A + B$ prepended, hereafter denoted $A + B + 7$. Also note the significant improvement relative to McBride's estimate of 2^{-118} for the best ELCP value of a restricted characteristic.

Number of rounds	Full rounds only	With additional S appended	With additional $A + B$ prepended
6	$2^{-72.3}$	$2^{-77.2}$	$2^{-78.8}$
7	$2^{-83.7}$	$2^{-88.6}$	$2^{-90.1}$
8	$2^{-95.1}$	$2^{-100.0}$	$2^{-101.5}$
9	$2^{-106.4}$	$2^{-111.3}$	-

Table 7.5: Best ELP values

Number of rounds	Full rounds only	With additional S appended	With additional $A + B$ prepended
6	$2^{-92.9}$	$2^{-105.35}$	$2^{-95.5}$
7	$2^{-107.9}$	$2^{-120.35}$	$2^{-110.5}$
8	$2^{-122.9}$	$2^{-135.35}$	$2^{-125.5}$
9	$2^{-137.9}$	$2^{-150.35}$	-

Table 7.6: Corresponding best EDP values from Biham et al. [15]

7.2.6 Recovering the Full Key

A linear hull over $A + B + 7$ with input/output masks that each contain a single 1 can be used to derive two bytes of keying information: the byte XOR'd before the active copy of S in the first substitution stage, and the byte XOR'd after the active copy of S in the last substitution stage. Therefore we need 2^{16} counters to carry out linear cryptanalysis using such a linear hull. Note that the bytes we obtain are in fact pieces of the 128-bit vectors $(KW1 \oplus KA)$ and $(KB \oplus KW2)$, respectively, i.e., they do not give us subkey bytes directly.

By carrying out linear cryptanalysis with 16 different linear hulls, each of which activates a different copy of S in the last substitution stage, we can systematically recover the bytes of $(KB \oplus KW2)$. Using our algorithm, we found the best linear hull for attacking each of these bytes; these are given in Table 7.7. The smallest of the 16 ELP values is approximately 2^{-91} , so opting for a 99.9% success rate for each linear hull requires a data complexity of $\frac{64}{2^{-91}} = 2^{97}$ (Table 3.1). Assuming that the success rates of the 16 individual attacks are independent, the overall success rate is $(0.999)^{16} \approx 98.4\%$.

Once we have determined $(KB \oplus KW2)$, we can exhaustively search all 2^{32} bits of information in KB (see Section 7.2.2). For each guess of KB we obtain a guess of $KW2$, and this can be used to extract the remaining subkeys by running the key-scheduling algorithm backwards. A simple trial encryption can be used to discard wrong guesses of KB , so a total of 2^{32} trial encryptions are required.

The above attack constitutes a theoretical break of Q.

Byte of ($KB \oplus KW2$)	Position of 1 in mask (input,output)	ELP	Number of characteristics in linear hull
0	(31, 3)	$2^{-91.1}$	94,726,326
1	(7, 11)	$2^{-91.1}$	94,726,326
2	(15, 19)	$2^{-91.1}$	94,726,326
3	(23, 27)	$2^{-91.1}$	94,726,326
4	(31, 35)	$2^{-90.1}$	191,795,706
5	(7, 43)	$2^{-90.1}$	191,795,706
6	(15, 51)	$2^{-90.1}$	191,795,706
7	(23, 59)	$2^{-90.1}$	191,795,706
8	(23, 67)	$2^{-90.2}$	188,281,125
9	(31, 75)	$2^{-90.2}$	188,281,125
10	(7, 83)	$2^{-90.2}$	188,281,125
11	(15, 91)	$2^{-90.2}$	188,281,125
12	(7, 99)	$2^{-90.2}$	183,092,934
13	(15, 107)	$2^{-90.2}$	183,092,934
14	(7, 115)	$2^{-90.2}$	183,092,934
15	(15, 123)	$2^{-90.2}$	183,092,934

Table 7.7: Best linear hulls for attacking the bytes of ($KB \oplus KW2$)

7.2.7 Reasons for the Success of Our Attack

There are several reasons that we are able to find linear hulls with large ELP values in Q . First, each of the three s-boxes in Q has multiple nonzero LP values corresponding to input/output masks containing a single 1. In contrast, one of the design criteria for A and B was the absence of nonzero DP values corresponding to input/output differences containing a single 1 [79]. This is the primary reason that our ELP values are superior to the EDP values in [15]. Second, the linear transformations in Q have low

diffusion (branch number $\mathcal{B}_l = 2$); in fact, since they are bitwise permutations, they cause a mask containing a single 1 to be transformed into a mask that also contains a single 1. Together, these two facts mean that there exist restricted characteristics with nonzero ELCP values; in a well-designed cipher, no such characteristics would exist.

Further, it is easy to combine a large number of restricted characteristics into a linear hull, enabling the attack we have presented above. Finally, the cryptanalyst's job is made easier by the reversibility of the key-scheduling algorithm, which means that determining any 128-bit subkey other than KA or KB allows the remaining subkeys to be recovered. This could be avoided by building a one-way property into the key-scheduling algorithm.

```

Initialize all Count and LP entries in DS[ ] to 0
For  $i = 0$  to 127
  DS[ $i, 0, i$ ].Count  $\leftarrow 1$ 
  DS[ $i, 0, i$ ].ELP  $\leftarrow 1$ 

For  $i = 0$  to 127
   $t \leftarrow 0$ 
  ApplyA ( $i, t$ );  $t += 1$ 
  Permute ( $i, t$ )
  ApplyB ( $i, t$ );  $t += 1$ 

  For Round = 1 to 7
    ApplyS ( $i, t$ );  $t += 1$ 
    ApplyA ( $i, t$ );  $t += 1$ 
    Permute ( $i, t$ )
    ApplyB ( $i, t$ );  $t += 1$ 

Subroutine ApplyS ( $i, t$ )
   $\mathcal{J} \leftarrow \{j : DS[i, t, j].Count > 0\}$ 
  For  $j \in \mathcal{J}$ 
    BoxIndex  $\leftarrow j \text{ div } 8$ 
    InBit  $\leftarrow j \text{ mod } 8$ 
    For OutBit = 0 to 7
      If LPS[InBit, OutBit]  $\neq 0$ 
         $\tilde{j} \leftarrow 8 \times \text{BoxIndex} + \text{OutBit}$ 
        DS[ $i, t + 1, \tilde{j}$ ].Count  $+= DS[i, t, j].Count$ 
        DS[ $i, t + 1, \tilde{j}$ ].ELP  $+= DS[i, t, j].ELP \times LP_S[\text{InBit}, \text{OutBit}]$ 

```

Figure 7.3: Pseudocode for computation of linear hulls over 23 core stages

<pre> Subroutine ApplyA (<i>i</i>, <i>t</i>) PreSerpent (<i>i</i>, <i>t</i>) $\mathcal{J} \leftarrow \{j : \text{DS}[i, t, j].\text{Count} > 0\}$ For $j \in \mathcal{J}$ BoxIndex $\leftarrow j \text{ div } 4$ InBit $\leftarrow j \text{ mod } 4$ For OutBit = 0 to 3 If $\text{LP}_A[\text{InBit}, \text{OutBit}] \neq 0$ $\tilde{j} \leftarrow 4 \times \text{BoxIndex} + \text{OutBit}$ $\text{DS}[i, t + 1, \tilde{j}].\text{Count} += \text{DS}[i, t, j].\text{Count}$ $\text{DS}[i, t + 1, \tilde{j}].\text{ELP} += \text{DS}[i, t, j].\text{ELP} \times \text{LP}_A[\text{InBit}, \text{OutBit}]$ End For PostSerpent (<i>i</i>, <i>t</i> + 1) </pre>
<pre> Subroutine PreSerpent (<i>i</i>, <i>t</i>) For $j = 0$ to 127 Temp[<i>j</i>] $\leftarrow \text{DS}[i, t, j]$ End For For $j = 0$ to 127 $\tilde{j} \leftarrow 4 \times (j \text{ mod } 32) + (j \text{ div } 32)$ $\text{DS}[i, t, \tilde{j}] \leftarrow \text{Temp}[j]$ End For </pre>
<pre> Subroutine Permute (<i>i</i>, <i>t</i>) Partition $\text{DS}[i, t, \cdot]$ into 4 “words” of size 32 (W_0, W_1, W_2, W_3): $W_s \leftarrow \langle \text{DS}[i, t, 32s], \dots, \text{DS}[i, t, 32s + 31] \rangle$, for $s = 0 \dots 3$ Leave W_0 unchanged Right rotate W_1 by 8, W_2 by 16, W_3 by 24 </pre>

Figure 7.4: Pseudocode for other subroutines

Chapter 8

Conclusions

8.1 Summary of Thesis

In this thesis we have dealt with linear cryptanalysis of substitution-permutation networks. In particular, we have focused on the linear hull concept, an approach that has proven especially fruitful.

Our consideration of SPNs with randomly selected s-boxes has produced new expressions for important values related to linear cryptanalysis. First, we have derived an exact expression (feasible to compute) for expected linear probability values for such SPNs. Experimental evidence indicates that this expression converges to the corresponding value for the true random cipher with an increasing number of rounds. This contributes to the quantitative evidence that the SPN structure is a good approximation to the true random cipher. Second, we have obtained a lower bound on the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis after a given number of rounds. For common block sizes, this bound appears to converge rapidly to 1.

In our analysis of SPNs with fixed s-boxes, we have presented two new algorithms, KMT1 and KMT2, for upper bounding the maximum average linear hull probability for SPNs. At the time of this writing, these are the only such algorithms that are completely general, in that they can be applied to any SPN, and they compute an upper bound that is a function of the number of encryption rounds. In contrast, other approaches to this problem either require that the SPN linear transformation have a specific structure, or compute a single value independent of the number of rounds. KMT2 produces a tighter upper bound than KMT1 by taking into account detailed information about the distribution of linear probability values for the SPN s-boxes. (As a consequence, KMT2 is also more computationally expensive.) By applying KMT1 and KMT2 to the AES, we established the provable security of the AES against linear cryptanalysis.

The concept of linear hulls played an instrumental role in the design of KMT1 and KMT2. Our work with linear hulls also gave us insight into certain fundamental weaknesses in the Q cipher, a NESSIE candidate. This enabled us to break Q using linear hulls, even though Q has good security against linear cryptanalysis when only characteristics are used. To our knowledge, this is the first use of linear hulls to break a proposed cipher.

In conclusion, the SPN structure possesses a simplicity that facilitates detailed analysis. When we began this research, linear cryptanalysis of SPNs represented an area that was largely untapped. By focusing particularly on linear hulls, we have made several important contributions to the rigorous application of linear cryptanalysis to this important block cipher architecture.

8.2 Ideas for Future Research

The work presented in this thesis gives rise to many ideas for future research.

1. All of the results of this thesis are tailored to the SPN structure. It may be possible to modify many of these results to apply to Feistel networks.
2. To this point, we have only applied KMT1 and KMT2 to the AES (due largely to the computational complexity involved). There are many other SPNs for which provable security against linear cryptanalysis has not yet been established—applying these two algorithms would yield new results in this direction. For AES-like SPNs, running times should be comparable to those for the AES.
3. Conjecture 4.4.1 and Conjecture 4.4.4 remain unproven.
4. As noted in Section 3.4.2, existing literature dealing with multiple linear approximations is based on the use of linear characteristics. Adapting multiple linear approximations to an approach based on linear hulls would be of theoretical and practical interest.
5. We believe that the attack on the Q cipher can be significantly improved by expanding the set of linear characteristics over which we sum ELCP values. It should be straightforward to include characteristics that activate a small number of s-boxes in each round (instead of just restricted characteristics). This may result in an attack that can be carried out in practice. The use of multiple linear approximations (see previous item) might yield an additional improvement. Although Q is officially “broken,” improving our current attack would further emphasize the importance of the linear hull effect.

6. In Chapter 5 we lower bound the probability that an SPN with randomly selected s-boxes is practically secure against linear cryptanalysis. The author is not aware of any results concerning the provable security of this SPN model.
7. Although linear hulls remove the need for the approximation in (3.8), we are still making use of the approximation in (3.4) (the Hypothesis of Fixed-Key Equivalence). One approach to removing this assumption is to compute or upper bound the following value:

$$\max_{\tilde{\mathbf{k}}} \max_{\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}} LP^{[1\dots T]}(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}}).$$

Another approach is to compute/bound the fraction of keys, $\tilde{\mathbf{k}}$, for which

$$\max_{\mathbf{a}, \mathbf{b} \in \{0,1\}^N \setminus \mathbf{0}} LP^{[1\dots T]}(\mathbf{a}, \mathbf{b}; \tilde{\mathbf{k}}).$$

is small/large. This matter of examining a “non-averaged” version of linear cryptanalysis is almost entirely unexplored.

Bibliography

- [1] C. Adams, *A formal and practical design procedure for substitution-permutation network cryptosystems*, Ph.D. Thesis, Queen's University, Kingston, Canada, 1990.
- [2] C. Adams, *Constructing symmetric ciphers using the CAST design procedure*, *Designs, Codes, and Cryptography*, Vol. 12, No. 3, pp. 283–316, November 1997.
- [3] C. Adams, *The CAST-256 encryption algorithm*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.
- [4] R. Anderson, E. Biham, and L. Knudsen, *Serpent: A flexible block cipher with maximum assurance*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.
- [5] K. Aoki and K. Ohta, *Strict evaluation of the maximum average of differential probability and the maximum average of linear probability*, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E80-A, No. 1, 1997.
- [6] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, *Camellia: A 128-bit block cipher suitable for multiple platforms—design and analysis*, Seventh Annual International Workshop on Selected Areas in Cryptography (SAC 2000), LNCS 2012, pp. 39–56, Springer-Verlag, 2001.
- [7] F. Ayoub, *The design of complete encryption networks using cryptographically equivalent permutations*, *Computers and Security*, Vol. 2, 261–267, 1983.
- [8] F. Ayoub, *Probabilistic completeness of substitution-permutation encryption networks*, *IEE Proceedings*, Vol. 129, Part E, No. 5, 195–199, September 1982.

- [9] E. Biham, *New types of cryptanalytic attacks using related keys*, Advances in Cryptology—EUROCRYPT'93, LNCS 765, pp. 398–409, Springer-Verlag, 1994.
- [10] E. Biham, *On Matsui's linear cryptanalysis*, Advances in Cryptology—EUROCRYPT'94, LNCS 950, pp. 341–355, Springer-Verlag, 1995.
- [11] E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Advances in Cryptology—CRYPTO'90, LNCS 537, pp. 2–21, Springer-Verlag, 1991.
- [12] E. Biham and A. Shamir, *Differential cryptanalysis of DES-like cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, pp. 3–72, 1991.
- [13] E. Biham and A. Shamir, *Differential cryptanalysis of the full 16-round DES*, Advances in Cryptology—CRYPTO'92, LNCS 740, pp. 487–496, Springer-Verlag, 1993.
- [14] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
- [15] E. Biham, V. Furman, M. Mızstal, and V. Rijmen, *Differential cryptanalysis of Q*, Fast Software Encryption (FSE 2001), LNCS 2355, pp. 174–186, Springer-Verlag, 2002.
- [16] A. Biryukov and A. Shamir, *Structural cryptanalysis of SASAS*, Advances in Cryptology—EUROCRYPT 2001, LNCS 2045, pp. 394–405, Springer-Verlag, 2001.
- [17] D. Biss, *A lower bound on the number of functions satisfying the strict avalanche criterion*, Discrete Mathematics, Vol. 185, pp. 29–39, 1998.
- [18] L. Brown, J. Pieprzyk, and J. Seberry, *LOKI: A cryptographic primitive for authentication and secrecy applications*, Advances in Cryptology—AUSCRYPT'90, LNCS 453, pp. 229–236, Springer-Verlag, 1990.
- [19] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. Matyas, Jr., L. O'Connor, M. Peyravian, D. Safford, N. Zunic, *MARS—a candidate cipher for AES*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.
- [20] C. Carlet and A. Gouget, *An upper bound on the number of m -resilient boolean functions*, Advances in Cryptology—ASIACRYPT 2002, LNCS 2501, pp. 484–496, Springer-Verlag, 2002.

- [21] Z.G. Chen and S.E. Tavares, *Towards provable security of substitution-permutation encryption networks*, Fifth Annual International Workshop on Selected Areas in Cryptography (SAC'98), LNCS 1556, pp. 43–56, Springer-Verlag, 1999.
- [22] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*, The MIT Press, 2001.
- [23] J. Daemen, R. Govaerts, and J. Vandewalle, *Correlation matrices*, Fast Software Encryption: Second International Workshop, LNCS 1008, pp. 275–285, Springer-Verlag, 1995.
- [24] J. Daemen, L. Knudsen, and V. Rijmen, *The block cipher SQUARE*, Fast Software Encryption (FSE'97), LNCS 1267, pp. 149–165, Springer-Verlag, 1997.
- [25] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer-Verlag, 2002.
- [26] W. Diffie and M. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory, Vol. IT-22, No. 6, November 1976.
- [27] W. Diffie and M. Hellman, *Exhaustive cryptanalysis of the NBS Data Encryption Standard*, Computer, Vol. 10, No. 6, pp. 74–84, June 1977.
- [28] H. Feistel, *Cryptography and computer privacy*, Scientific American, Vol. 228, No. 5, pp. 15–23, May 1973.
- [29] H. Feistel, W.A. Notz, and J.L. Smith, *Some cryptographic techniques for machine to machine data communications*, Proceedings of the IEEE, Vol. 63, No. 11, pp. 1545–1554, November 1975.
- [30] N. Ferguson, R. Schroepel, and D. Whiting, *A simple algebraic representation of Rijndael*, Eighth Annual International Workshop on Selected Areas in Cryptography (SAC 2001), LNCS 2259, pp. 103–111, Springer-Verlag, 2001.
- [31] N. Ferguson and B. Schneier, *Practical Cryptography*, John Wiley and Sons, 2003.
- [32] FIPS 46, *Data Encryption Standard*, Federal Information Processing Standards Publication 46, U.S. Department of Commerce, National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1977.

- [33] FIPS 197, *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197, U.S. Department of Commerce, National Institute of Standards and Technology, National Technical Information Service, Springfield, Virginia, November 26, 2001.
- [34] FIPS 180-1, *Secure hash standard*, Federal Information Processing Standards Publication 180-1, U.S. Department of Commerce, National Institute of Standards and Technology, National Technical Information Service, Springfield, Virginia, April 17, 1995.
- [35] H. Handschuh and D. Naccache, *SHACAL*, First Open NESSIE Workshop, Proceedings, Leuven, Belgium, November 2000.
- [36] C. Harpes, G. Kramer, and J. Massey, *A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma*, Advances in Cryptology—EUROCRYPT'95, LNCS 921, pp. 24–38, Springer-Verlag, 1995.
- [37] H.M. Heys, *The design of substitution-permutation network ciphers resistant to cryptanalysis*, Ph.D. Thesis, Queen's University, Kingston, Canada, 1994.
- [38] H.M. Heys and S.E. Tavares, *Avalanche characteristics of substitution-permutation encryption networks*, IEEE Transactions on Computers, Vol. 44, No. 9, pp. 1131–1139, September 1995.
- [39] H.M. Heys and S.E. Tavares, *Substitution-permutation networks resistant to differential and linear cryptanalysis*, Journal of Cryptology, Vol. 9, No. 1, pp. 1–19, 1996.
- [40] H.M. Heys and S.E. Tavares, *Cryptanalysis of substitution-permutation networks using key-dependent degeneracy*, Cryptologia, Vol. 20, No. 3, pp. 258–274, 1996.
- [41] S. Hong, S. Lee, J. Lim, J. Sung, and D. Cheon, *Provable security against differential and linear cryptanalysis for the SPN structure*, Fast Software Encryption (FSE 2000), LNCS 1978, pp. 273–283, Springer-Verlag, 2001.
- [42] T. Hungerford, *Algebra*, Graduate Texts in Mathematics 73, Springer-Verlag, 1974.
- [43] K. Itakura, *Personal communication*, 2003.
- [44] T. Jakobsen and L. Knudsen, *The interpolation attack on block ciphers*, Fast Software Encryption (FSE'97), LNCS 1267, pp. 28–40, Springer-Verlag, 1997.

- [45] T. Jakobsen and L. Knudsen, *Attacks on block ciphers of low algebraic degree*, Journal of Cryptology, Vol. 14, No. 3, pp. 197–210, 2001.
- [46] P. Junod, *On the complexity of Matsui's attack*, Eighth Annual International Workshop on Selected Areas in Cryptography (SAC 2001), LNCS 2259, pp. 199–211, Springer-Verlag, 2001.
- [47] P. Junod and S. Vaudenay, *Optimal key ranking procedures in a statistical cryptanalysis*, Fast Software Encryption (FSE 2003), to appear in Lecture Notes in Computer Science, Springer-Verlag.
- [48] B. Kaliski and M. Robshaw, *Linear cryptanalysis using multiple approximations*, Advances in Cryptology—CRYPTO'94, LNCS 839, pp. 26–39, Springer-Verlag, 1994.
- [49] B. Kaliski and M. Robshaw, *Linear cryptanalysis using multiple approximations and FEAL*, Fast Software Encryption: Second International Workshop, LNCS 1008, pp. 249–264, Springer-Verlag, 1995.
- [50] J.B. Kam and G.I. Davida, *Structured design of substitution-permutation encryption networks*, IEEE Transactions on Computers, Vol. C-28, No. 10, pp. 747–753, October 1979.
- [51] M. Kanda, *Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function*, Seventh Annual International Workshop on Selected Areas in Cryptography (SAC 2000), LNCS 2012, pp. 324–338, Springer-Verlag, 2001.
- [52] M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki, and K. Ohta, *A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis*, Fifth Annual International Workshop on Selected Areas in Cryptography (SAC'98), LNCS 1556, pp. 264–279, Springer-Verlag, 1999.
- [53] M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta, and T. Matsumoto, *E2—A candidate cipher for AES*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.
- [54] J.-S. Kang, C. Park, S. Lee, and J. Lim, *On the optimal diffusion layers with practical security against differential and linear cryptanalysis*, Information Security and Cryptology—ICISC'99, LNCS 1787, pp. 38–52, Springer-Verlag, 2000.

- [55] J.-S. Kang, S. Hong, S. Lee, O. Yi, C. Park, and J. Lim, *Practical and provable security against differential and linear cryptanalysis for substitution-permutation networks*, ETRI Journal, Vol. 23, No. 4, December 2001.
- [56] L. Keliher and H. Meijer, *A new substitution-permutation network cryptosystem using key-dependent s-boxes*, Proceedings of Fourth Annual Workshop on Selected Areas in Cryptography (SAC'97), Carleton University, Ottawa, Canada, pp. 13–26, August 1997.
- [57] L. Keliher, H. Meijer, and S. Tavares, *Modeling linear characteristics of substitution-permutation networks*, Sixth Annual International Workshop on Selected Areas in Cryptography (SAC'99), LNCS 1758, pp. 78–91, Springer-Verlag, 2000.
- [58] L. Keliher, H. Meijer, and S. Tavares, *New method for upper bounding the maximum average linear hull probability for SPNs*, Advances in Cryptology—EUROCRYPT 2001, LNCS 2045, pp. 420–436, Springer-Verlag, 2001.
- [59] L. Keliher, H. Meijer, and S. Tavares, *Dual of new method for upper bounding the maximum average linear hull probability for SPNs*, Technical Report, IACR ePrint Archive (<http://eprint.iacr.org>, Paper # 2001/033), 2001.
- [60] L. Keliher, H. Meijer, and S. Tavares, *Improving the upper bound on the maximum average linear hull probability for Rijndael*, Eighth Annual International Workshop on Selected Areas in Cryptography (SAC 2001), LNCS 2259, pp. 112–128, Springer-Verlag, 2001.
- [61] L. Keliher, H. Meijer, and S. Tavares, *High probability linear hulls in Q* , Proceedings of Second Open NESSIE Workshop, Royal Holloway College, University of London, Egham, U.K, 2001, selected to appear in LNCS volume for the NESSIE project, Springer-Verlag.
- [62] L. Keliher, H. Meijer, and S. Tavares, *Toward the true random cipher: On expected linear probability values for SPNs with randomly selected s-boxes*, chapter in Communications, Information and Network Security, V. Bhargava, H. Poor, V. Tarokh, and S. Yoon (Eds.), pp. 123–146, Kluwer Academic Publishers, 2003.
- [63] M.G. Kendall, *The Advanced Theory of Statistics, Volume I*, Charles Griffin & Company Limited, 1943.
- [64] J. Kilian and P. Rogaway, *How to protect DES against exhaustive key search*, Advances in Cryptology—CRYPTO'96, LNCS 1109, pp. 252–267, Springer-Verlag, 1996.

- [65] L.R. Knudsen, *Practically secure Feistel ciphers*, Fast Software Encryption, LNCS 809, pp. 211–221, Springer-Verlag, 1994.
- [66] L.R. Knudsen, *Truncated and higher order differentials*, Fast Software Encryption: Second International Workshop, LNCS 1008, pp. 196–211, Springer-Verlag, 1995.
- [67] L. Knudsen, *Block Ciphers—Analysis, Design and Applications*, Ph.D. Thesis, University of Aarhus, Aarhus, Denmark, 1994.
- [68] X. Lai, *Higher order derivatives and differential cryptanalysis*, in Communications and Cryptography, Two Sides of One Tapestry, Blahut et al. (Eds.), pp. 227–223, Kluwer Academic Publishers, 1994.
- [69] X. Lai and J. Massey, *A proposal for a new block encryption standard*, Advances in Cryptology—EUROCRYPT’90, LNCS 473, pp. 389–404, Springer-Verlag, 1991.
- [70] X. Lai, J. Massey, and S. Murphy, *Markov ciphers and differential cryptanalysis*, Advances in Cryptology—EUROCRYPT’91, LNCS 547, pp. 17–38, Springer-Verlag, 1991.
- [71] S.K. Langford and M.E. Hellman, *Differential-linear cryptanalysis*, Advances in Cryptology—CRYPTO’94, LNCS 839, pp. 17–25, Springer-Verlag, 1994.
- [72] C.H. Lim, *CRYPTON: A new 128-bit block cipher*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.
- [73] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, Elsevier/North-Holland, 1977.
- [74] M. Matsui, *Linear cryptanalysis method for DES cipher*, Advances in Cryptology—EUROCRYPT’93, LNCS 765, pp. 386–397, Springer-Verlag, 1994.
- [75] M. Matsui, *On correlation between the order of s-boxes and the strength of DES*, Advances in Cryptology—EUROCRYPT’94, LNCS 950, pp. 366–375, Springer-Verlag, 1995.
- [76] M. Matsui, *The first experimental cryptanalysis of the Data Encryption Standard*, Advances in Cryptology—CRYPTO’94, LNCS 839, pp. 1–11, Springer-Verlag, 1994.

- [77] M. Matsui, *New Block Encryption Algorithm MISTY*, Fast Software Encryption (FSE'97), LNCS 1267, pp. 54–68, Springer-Verlag, 1997.
- [78] M. Matsui and A. Yamagishi, *A new method for known plaintext attack of FEAL cipher*, Advances in Cryptology—EUROCRYPT'92, LNCS 658, pp. 81–91, Springer-Verlag, 1993.
- [79] L. McBride, *Q: A Proposal for NESSIE v2.00*, First NESSIE Workshop, Leuven, Belgium, November 2000.
- [80] W. Meier and O. Staffelbach, *Nonlinearity criteria for cryptographic functions*, Advances in Cryptology—EUROCRYPT'89, LNCS 434, pp. 549–562, Springer-Verlag, 1990.
- [81] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [82] R. Merkle, *Fast software encryption functions*, Advances in Cryptology—CRYPTO'90, LNCS 537, pp. 476–501, Springer-Verlag, 1991.
- [83] S. Mister and C. Adams, *Practical s-box design*, Proceedings of Third Annual Workshop on Selected Areas in Cryptography (SAC'96), Queen's University, Kingston, Canada, pp. 61–76, August 1996.
- [84] S. Murphy, *The cryptanalysis of FEAL-4 with 20 chosen plaintexts*, Journal of Cryptology, Vol. 2, No. 3, pp. 145–154, 1990.
- [85] S. Murphy and M. Robshaw, *Essential algebraic structure within the AES*, Advances in Cryptology—CRYPTO 2002, LNCS 2442, pp. 1–16, Springer-Verlag, 2002.
- [86] National Institute of Standards and Technology, *Announcing request for candidate algorithm nominations for the Advanced Encryption Standard (AES)*, Federal Register, Vol. 62, No. 177, pp. 48051–48058, September 12, 1997.
- [87] National Institute of Standards and Technology, Information Technology Laboratory, *The First Advanced Encryption Standard Candidate Conference*, Proceedings, Ventura, California, August 1998.
- [88] K. Nyberg, *Perfect nonlinear s-boxes*, Advances in Cryptology—EUROCRYPT'91, LNCS 547, pp. 378–386, Springer-Verlag, 1991.
- [89] K. Nyberg, *Differentially uniform mappings for cryptography*, Advances in Cryptology—EUROCRYPT'93, LNCS 765, pp. 55–64, Springer-Verlag, 1994.

- [90] K. Nyberg, *Linear approximation of block ciphers*, Advances in Cryptology—EUROCRYPT'94, LNCS 950, pp. 439–444, Springer-Verlag, 1995.
- [91] K. Nyberg and L. Knudsen, *Provable security against a differential attack*, Journal of Cryptology, Vol. 8, No. 1, pp. 27–37, 1995.
- [92] L. O'Connor, *An analysis of product ciphers based on the properties of Boolean functions*, Ph.D. Thesis, University of Waterloo, Waterloo, Canada, 1992.
- [93] L. O'Connor, *On the distribution of characteristics in bijective mappings*, Advances in Cryptology—EUROCRYPT'93, LNCS 765, pp. 360–370, Springer-Verlag, 1994.
- [94] L. O'Connor, *Properties of linear approximation tables*, Fast Software Encryption: Second International Workshop, LNCS 1008, pp. 131–136, Springer-Verlag, 1995.
- [95] K. Ohkuma, H. Muratani, F. Sano, and S. Kawamura, *The block cipher Hierocrypt*, Seventh Annual International Workshop on Selected Areas in Cryptography (SAC 2000), LNCS 2012, pp. 72–88, Springer-Verlag, 2001.
- [96] S. Park, S.H. Sung, S. Chee, E-J. Yoon, and J. Lim, *On the security of Rijndael-like structures against differential and linear cryptanalysis*, Advances in Cryptology—ASIACRYPT 2002, LNCS 2501, pp. 176–191, Springer-Verlag, 2002.
- [97] S. Park, S.H. Sung, S. Lee, J. Lim, *Improving the upper bound on the maximum differential and the maximum linear hull probability for SPN structures and AES*, Fast Software Encryption (FSE 2003), to appear in Lecture Notes in Computer Science, Springer-Verlag.
- [98] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, *Propagation characteristics of Boolean functions*, Advances in Cryptology—EUROCRYPT'90, LNCS 473, pp. 161–173, Springer-Verlag, 1991.
- [99] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, E. De Win, *The cipher SHARK*, Fast Software Encryption: Third International Workshop, LNCS 1039, pp. 99–112, Springer-Verlag, 1996.
- [100] R. Rivest, M. Robshaw, R. Sidney, and Y. Lin, *The RC6 block cipher*, The First Advanced Encryption Standard Candidate Conference, Proceedings, Ventura, California, August 1998.

- [101] F. Sano, K. Ohkuma, H. Shimizu, and S. Kawamura, *On the security of nested SPN cipher against the differential and linear cryptanalysis*, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E86-A, No. 1, pp. 37–46, 2003.
- [102] B. Schneier, *Description of a new variable-length key, 64-bit block cipher (Blowfish)*, Fast Software Encryption, LNCS 809, pp. 191–204, Springer-Verlag, 1994.
- [103] B. Schneier and J. Kelsey, *Unbalanced Feistel networks and block-cipher design*, Fast Software Encryption: Third International Workshop, LNCS 1039, pp. 121–144, Springer-Verlag, 1996.
- [104] B. Schneier, *Applied Cryptography, Second Ed.*, John Wiley and Sons, 1996.
- [105] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*, John Wiley and Sons, 1999.
- [106] J. Seberry, X. Zhang, and Y. Zheng, *Relationships among nonlinearity criteria*, Advances in Cryptology—EUROCRYPT’94, LNCS 950, pp. 376–388, Springer-Verlag, 1995.
- [107] C.E. Shannon, *Communication theory of secrecy systems*, Bell System Technical Journal, Vol. 28, no. 4, pp. 656–715, 1949.
- [108] A. Shimizu and S. Miyaguchi, *Fast data encipherment algorithm FEAL*, Advances in Cryptology—EUROCRYPT’87, LNCS 304, pp. 267–278, Springer-Verlag, 1988.
- [109] M. Sivabalan, S. Tavares, and L. Peppard, *On the design of SP networks from an information theoretic point of view*, Advances in Cryptology—CRYPTO’92, LNCS 740, pp. 260–279, Springer-Verlag, 1993.
- [110] A. Sorkin, *Lucifer, A cryptographic algorithm*, Cryptologia, Vol. 8, No. 1, pp. 22–41, 1984, with addenda in Vol. 8, No. 3, pp. 260–261, 1984.
- [111] W. Stallings, *Cryptography and Network Security: Principles and Practice, Third Edition*, Prentice Hall, 2003.
- [112] D.R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.
- [113] S. Vaudenay, *An experiment on DES statistical cryptanalysis*, Proceedings of 3rd ACM Conference on Computer and Communications Security, pp. 139–147, ACM Press, 1996.

- [114] S. Vaudenay, *On the security of CS-Cipher*, Fast Software Encryption (FSE'99), LNCS 1636, pp. 260–274, Springer-Verlag, 1999.
- [115] A.F. Webster and S.E. Tavares, *On the design of s-boxes*, Advances in Cryptology—CRYPTO'85, LNCS 218, pp. 523–534, Springer-Verlag, 1986.
- [116] A.M. Youssef, *Analysis and design of block ciphers*, Ph.D. Thesis, Queen's University, Kingston, Canada, 1997.
- [117] A.M. Youssef and S.E. Tavares, *On the avalanche characteristics of substitution-permutation networks*, Proceedings of PRAGOCRYPT'96, pp. 18-29, 1996.
- [118] A.M. Youssef, S.E. Tavares, and H.M. Heys, *A new class of substitution-permutation networks*, Proceedings of Third Annual Workshop on Selected Areas in Cryptography (SAC'96), Queen's University, Kingston, Canada, pp. 132–147, August 1996.

Appendix A

Duality Between Linear and Differential Cryptanalysis

In this appendix we outline the duality between linear cryptanalysis and differential cryptanalysis. This duality enables us to translate certain results for one attack into corresponding results for the other attack. Our main goal is to present the differential cryptanalysis versions of KMT1 and KMT2, and to give the results of the application of these dual algorithms to the AES.

A.1 Elements of the Duality

The duality between linear and differential cryptanalysis is well known [10, 75]. The basis of the duality is a correspondence between pairs of concepts, one for each attack, as given in Table A.1.

Linear Cryptanalysis	Differential Cryptanalysis
input/output mask	input/output XOR difference
linear probability (LP)	differential probability (DP)
linear characteristic	differential characteristic
linear hull	differential
linear characteristic probability (LCP)	differential characteristic probability (DCP)
expected linear characteristic probability (ELCP)	expected differential characteristic probability (EDCP)
linearly active s-box	differentially active s-box
linear branch number (\mathcal{B}_l)	differential branch number (\mathcal{B}_d)
maximum average linear hull probability (MALHP)	maximum expected differential probability (MEDP)

Table A.1: Elements of duality between linear and differential cryptanalysis

A.2 Maximum Expected Differential Probability

In this section we simply highlight the details of differential cryptanalysis that are relevant to our discussion. For a thorough explanation of the attack, see [14]. For a description of *differentials*, which are the counterpart of linear hulls, see [70].

Before carrying out differential cryptanalysis, the attacker precomputes XOR differences $\Delta \mathbf{x}, \Delta \mathbf{y} \in \{0, 1\}^N \setminus \mathbf{0}$ for T core SPN rounds (often $T = R - 1$ or $T = R - 2$, as for linear cryptanalysis) such that the *expected differential probability* over those rounds, $EDP^{[1 \dots T]}(\Delta \mathbf{x}, \Delta \mathbf{y})$, is relatively large (see Definition 2.6.4). The attacker then chooses pairs of plaintexts $\langle \mathbf{p}_i, \mathbf{p}'_i \rangle$ satisfying $\mathbf{p}_i \oplus \mathbf{p}'_i = \Delta \mathbf{p}$, and obtains the corresponding ciphertexts $\langle \mathbf{c}_i, \mathbf{c}'_i \rangle$ (recall that differential cryptanalysis is a chosen-plaintext attack). The number of such plaintext pairs required (the data complexity) [39] is in

the order of

$$\mathcal{N}_D = \frac{1}{EDP^{[1\dots T]}(\Delta\mathbf{x}, \Delta\mathbf{y})}. \quad (\text{A.1})$$

To guarantee *provable security against differential cryptanalysis* [91], it is necessary to demonstrate that the *maximum expected differential probability* (MEDP) is sufficiently small that the corresponding data complexity is prohibitively large, where

$$MEDP = \max_{\Delta\mathbf{x}, \Delta\mathbf{y} \in \{0,1\}^N \setminus \mathbf{0}} EDP^{[1\dots T]}(\Delta\mathbf{x}, \Delta\mathbf{y}).$$

The MEDP is the counterpart of the MALHP in linear cryptanalysis. As for the MALHP, the exact value of the MEDP appears to be difficult to compute exactly, and therefore researchers have focused on upper bounding this value.

A.3 Upper Bounding the MEDP for SPNs

A series of results have been published upper bounding the MEDP for SPNs. In each case, such an upper bound was presented as a dual version of an upper bound on the MALHP. In this section we survey these results, and consider their application to the AES. Because of the duality, this material almost exactly parallels that in Section 6.1. We start with the counterparts of some important definitions from Section 3.3. Note that we continue to use \mathcal{L} to denote the SPN linear transformation represented as an invertible $N \times N$ binary matrix.

Definition A.3.1 (dual version of Definition 3.3.10). *Let $\gamma, \hat{\gamma} \in \{0,1\}^M$. Then*

$$W_d[\gamma, \hat{\gamma}] \stackrel{\text{def}}{=} \# \{ \Delta\mathbf{x} \in \{0,1\}^N : \gamma_{\Delta\mathbf{x}} = \gamma, \gamma_{\Delta\mathbf{y}} = \hat{\gamma}, \text{ where } \Delta\mathbf{y} = \mathcal{L}(\Delta\mathbf{x}) \}.$$

Definition A.3.2 (dual version of Definition 3.3.11). *The differential branch number, \mathcal{B}_d , of an SPN linear transformation is given by*

$$\begin{aligned} \mathcal{B}_d &\stackrel{\text{def}}{=} \min \{ wt(\gamma_{\Delta\mathbf{x}}) + wt(\gamma_{\Delta\mathbf{y}}) : \Delta\mathbf{x} \in \{0, 1\}^N \setminus \mathbf{0}, \Delta\mathbf{y} = \mathcal{L}(\Delta\mathbf{x}) \} \\ &= \min \{ wt(\gamma) + wt(\hat{\gamma}) : \gamma, \hat{\gamma} \in \{0, 1\}^M \setminus \mathbf{0}, W_d[\gamma, \hat{\gamma}] > 0 \}. \end{aligned}$$

Definition A.3.3 (dual version of Definition 3.3.13). *Let p be the maximum nontrivial DP value over all SPN s-boxes. Symbolically,*

$$p \stackrel{\text{def}}{=} \max_{S \in \text{SPN}} \max_{\alpha, \beta \in \{0, 1\}^n \setminus \mathbf{0}} DP^S(\alpha, \beta).$$

Daemen and Rijmen show that $p = 2^{-6}$ and $\mathcal{B}_d = 5$ for the AES [25] (the latter is true because $\mathcal{B}_d = 5$ for the 32-bit linear transformation component of the AES).

Lemma A.3.4 (dual version of Lemma 6.1.1). *Let $0 < \delta \leq 1$, and suppose $MEDP \leq \delta$ for T core SPN rounds. Then $MEDP \leq \delta$ for $T + 1$ core SPN rounds.*

Hong et al. [41] gave the first upper bound on the MEDP for SPNs in the following theorem.

Theorem A.3.5 (dual version of Theorem 6.1.2). *Let $T \geq 2$. If $\mathcal{B}_d = (M + 1)$, then $MEDP \leq p^M$. If $\mathcal{B}_d = M$, then $MEDP \leq p^{M-1}$.*

Since $M = 16$ and $\mathcal{B}_d = 5$ for the AES, Theorem A.3.5 cannot be applied. The first result applicable to the AES was due to Kang et al. [55].

Theorem A.3.6 (dual version of Theorem 6.1.3). *Let $T \geq 2$. Then $MEDP \leq p^{\mathcal{B}_d-1}$.*

Evaluating Theorem A.3.6 for the AES gives an upper bound of $p^4 = 2^{-24}$, for which the corresponding data complexity is not prohibitive (see (A.1)).

We then presented the dual version of KMT1 (hereafter denoted KMT1-DC) in [59], and applied this dual algorithm to the AES. The upper bound from KMT1-DC for the AES is plotted in Figure A.1 (the upper curve). It happens that for the AES, the upper bound from KMT1-DC is identical to the upper bound on the MALHP from KMT1 (see Figure 6.1). (We explain the reasons for these identical results in Section A.4.1.) For $T \geq 7$, then, the upper bound on the MEDP from KMT1-DC is 2^{-75} . This result established the provable security of the AES against differential cryptanalysis.

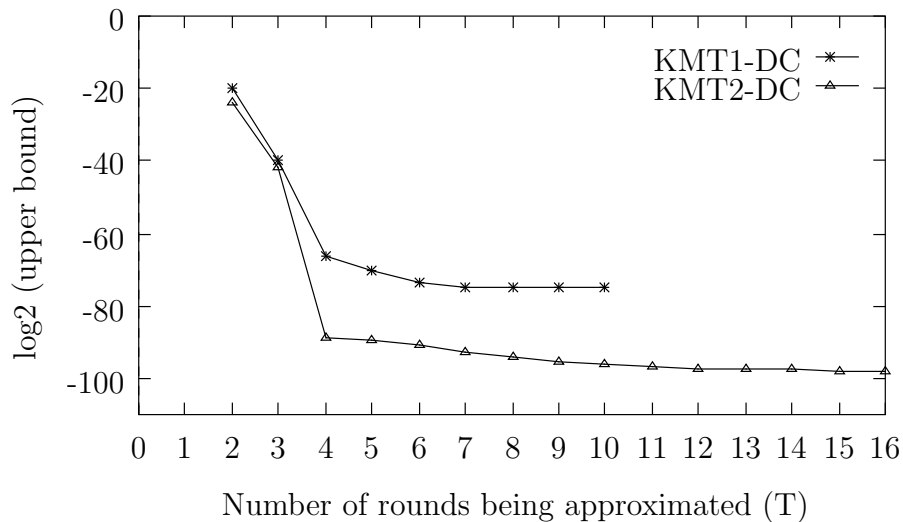


Figure A.1: Upper bounds from KMT1-DC and KMT2-DC for the AES

We did not present the dual version of KMT2 (KMT2-DC) when we introduced KMT2 in [60], due to a delay in obtaining computational results. In fact, the description of KMT2-DC in the current appendix is the first publication of this algorithm.¹ The upper bound from KMT2-DC for the AES is plotted in Figure A.1 (the lower

¹Therefore, in terms of our chronological listing of results, this discussion of KMT2-DC should be placed at the end of this section.

curve). The values from KMT2-DC are also given in Table A.2, for $2 \leq T \leq 16$. It is interesting to note that these values drop more rapidly than those for KMT2, but begin to level out sooner (see Figure 6.1 and Table 6.1).

Number of rounds	Upper bound from KMT2-DC	Number of rounds	Upper bound from KMT2-DC
1	—	9	$2^{-95.1}$
2	$2^{-24.0}$	10	$2^{-96.1}$
3	$2^{-42.0}$	11	$2^{-96.6}$
4	$2^{-88.6}$	12	$2^{-97.1}$
5	$2^{-89.5}$	13	$2^{-97.5}$
6	$2^{-90.7}$	14	$2^{-97.6}$
7	$2^{-92.5}$	15	$2^{-97.7}$
8	$2^{-93.9}$	16	$2^{-97.8}$

Table A.2: Upper bound from KMT2-DC for the AES

After KMT1-DC [59], Sano et al. [101] published the following theorem.

Theorem A.3.7 (dual version of Theorem 6.1.4). *Consider a nested SPN with M_1 high-level s-boxes, each of which is a 2-round SPN containing M_2 low-level s-boxes in each round. Suppose all linear transformations have maximum differential branch numbers, i.e., the high-level differential branch number is $(M_1 + 1)$ and the low-level differential branch number is $(M_2 + 1)$. Then for $T \geq 2$ core high-level rounds, $MEDP \leq p^{M_1 M_2}$.*

For the AES, Theorem A.3.7 yields an upper bound of 2^{-96} on the MEDP, since $M_1 = M_2 = 4$ and $p = 2^{-6}$.

The next result was due to Park et al. [96].

Theorem A.3.8 (dual version of Theorem 6.1.5). *Let $T \geq 4$ for an AES-like SPN. Then the MEDP is upper bounded by*

$$\max \{ 4p^{19} + 6p^{18} + 4p^{17} + p^{16}, 184p^{22} + 912p^{21} + 438p^{20} + 72p^{19} + 4p^{18} + p^{16} \}.$$

Applying Theorem A.3.8 to the AES gives an upper bound on the MEDP of 1.06×2^{-96} for $T \geq 4$.

Park et al. then gave an improved result for the AES in [97], based on the following theorem, which can be applied to any SPN.

Theorem A.3.9. *Let the s -boxes in the SPN substitution stage be enumerated from left to right as S_1, S_2, \dots, S_M . Then for $T \geq 2$, the MEDP is upper bounded by*

$$\max \left\{ \max_{\substack{1 \leq i \leq M \\ \alpha \in \{0,1\}^n \setminus \mathbf{0}}} \left[\sum_{\chi \in \{0,1\}^n \setminus \mathbf{0}} (DP^{S_i}(\alpha, \chi))^{\mathcal{B}_d} \right], \max_{\substack{1 \leq i \leq M \\ \beta \in \{0,1\}^n \setminus \mathbf{0}}} \left[\sum_{\chi \in \{0,1\}^n \setminus \mathbf{0}} (DP^{S_i}(\chi, \beta))^{\mathcal{B}_d} \right] \right\}.$$

When applied to the AES, Theorem A.3.9 gives an upper bound of 1.23×2^{-28} for $T \geq 2$ rounds. Park et al. used this to obtain an upper bound on the MEDP of $(1.23 \times 2^{-28})^4 \approx 1.144 \times 2^{-111}$ for $T \geq 4$ AES rounds.

A.4 The KMT1-DC Algorithm

The KMT1-DC algorithm is identical to the KMT1 algorithm given in Theorem 6.4.1 and Figure 6.6, except that q is replaced by p , and $W[]$ is replaced by $W_d[]$. The complexity analysis from Section 6.4.1 remains unchanged, except that the value \mathcal{H} may be affected by the use of $W_d[]$ in place of $W[]$.

A.4.1 Application of KMT1-DC to the AES

For the AES, $p = q = 2^{-6}$, and further, $W_d[\] = W[\]$ (see [59]). It follows that KMT1 and KMT1-DC are identical when applied to the AES.

A.5 The KMT2-DC Algorithm

The KMT2-DC algorithm is identical to the KMT2 algorithm given by the pseudocode in Figure 6.4 and Figure 6.5, and by the sequences defined Section 6.5, except for two modifications. First, $W[\]$ is replaced by $W_d[\]$. Second, the sequence

$$\langle e_{\alpha,1}^S, e_{\alpha,2}^S, \dots, e_{\alpha,2^n-1}^S \rangle$$

in Section 6.5.1, which is the basis for the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$, is obtained by using values of the form $DP^S(\alpha, \beta)$ instead of values of the form $LP^S(\alpha, \beta)$.

i		1	2	3
ρ_i		$\left(\frac{4}{256}\right)$	$\left(\frac{2}{256}\right)$	0
ϕ_i		1	126	129

Table A.3: Distribution of DP values for the AES s-box

A.5.1 Application of KMT2-DC to the AES

As noted in Section A.4.1, $W_d[\] = W[\]$ for the AES. When DP values are used in place of LP values, as explained above, the sequences $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ are much simpler than for KMT2, since not only does the AES s-box have the property that all nontrivial

rows and columns of its DP table have the same distribution of entries, but this distribution, given in Table A.3, is much simpler than the corresponding distribution for the LP table (compare Table 7.1). Because $\{u_i^\gamma\}$ and $\{v_i^{\hat{\gamma}}\}$ are greatly simplified, our implementation of KMT2-DC runs significantly faster than KMT2 when applied to the AES. On our benchmark 2.8GHz Pentium 4 (Section 7.1.3), the total running time is approximately 7000 hours, roughly 60% of the running time of KMT2 (see Section 7.1.4).

Acknowledgement

The computation required to apply KMT2-DC to the AES was performed on the Mount Allison Cluster for Advanced Research. We are especially grateful to Stacey Wetmore and Aaron Russell for access to this resource.

Vita

Name Liam Timothy Keliher
Place of birth Seattle, Washington, U.S.A.

Education

M.Sc. (Computer Science) Queen's University, 1997
M.Sc. (Mathematics) McGill University, 1996
B.Sc. Honours (Mathematics) St. Francis Xavier University, 1993

Scholarships and Awards

Ontario Graduate Scholarship in Science and Technology (OGSST), 2000–2001
Ontario Graduate Scholarship (OGS), 1998–2000
NSERC Postgraduate Scholarship (PGS-A), 1993–1995
A.A. MacDonald Prize for Mathematics, St. Francis Xavier University, 1993
Canada Scholarship, 1988–1991
Francis J. Ginivan Mathematics Prize, St. Francis Xavier University, 1990

Memberships in Professional Associations

Member of the International Association for Cryptologic Research (IACR)
Member of the Institute of Electrical and Electronics Engineers (IEEE)

Publications

- L. Keliher, H. Meijer, and S. Tavares, *High probability linear hulls in Q* , Proceedings of Second Open NESSIE Workshop, Royal Holloway College, University of London, Egham, U.K, 2001, selected to appear in LNCS volume for the NESSIE project, Springer-Verlag.
- L. Keliher, H. Meijer, and S. Tavares, *Toward the true random cipher: On expected linear probability values for SPNs with randomly selected s -boxes*, chapter in Communications, Information and Network Security, V. Bhargava, H. Poor, V. Tarokh, and S. Yoon (Eds.), pp. 123-146, Kluwer Academic Publishers, 2003.
- L. Keliher, H. Meijer, and S. Tavares, *Improving the upper bound on the maximum average linear hull probability for Rijndael*, Eighth Annual International Workshop on Selected Areas in Cryptography (SAC 2001), LNCS 2259, pp. 112–128, Springer-Verlag, 2001.
- L. Keliher, H. Meijer, and S. Tavares, *New method for upper bounding the maximum average linear hull probability for SPNs*, Advances in Cryptology—EUROCRYPT 2001, LNCS 2045, pp. 420–436, Springer-Verlag, 2001.
- L. Keliher, H. Meijer, and S. Tavares, *Dual of new method for upper bounding the maximum average linear hull probability for SPNs*, Technical Report, IACR ePrint Archive (<http://eprint.iacr.org>, Paper # 2001/033), 2001.
- L. Keliher, H. Meijer, and S. Tavares, *Modeling linear characteristics of substitution-permutation networks*, Sixth Annual International Workshop on Selected Areas in Cryptography (SAC'99), LNCS 1758, pp. 78–91, Springer-Verlag, 2000.

- L. Keliher and H. Meijer, *A new substitution-permutation network cryptosystem using key-dependent s-boxes*, Proceedings of Fourth Annual Workshop on Selected Areas in Cryptography (SAC'97), Carleton University, Ottawa, Canada, pp. 13–26, August 1997.