TKK Dissertations in Information and Computer Science

Espoo 2010 TKK-ICS-D16

MULTIDIMENSIONAL LINEAR CRYPTANALYSIS

Miia Hermelin

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

TKK Dissertations in Information and Computer Science

Espoo 2010 TKK-ICS-D16

MULTIDIMENSIONAL LINEAR CRYPTANALYSIS

Miia Hermelin

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T2 at the Aalto University School of Science and Technology (Espoo, Finland) on the 11th of June, 2010, at 12 noon.

Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Aalto yliopiston teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Aalto University School of Science and Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O.Box 15400
00076 Aalto
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 470 01
Fax +358 9 470 23369
E-mail: series@ics.tkk.fi

ISBN 978-952-60-3189-7 (Print)
ISBN 978-952-60-3190-3 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)
URL: http://lib.tkk.fi/Diss/2010/isbn9789526031903/

Multiprint
Espoo 2010

ABSTRACT: Linear cryptanalysis is an important tool for studying the security of symmetric ciphers. In 1993 Matsui proposed two algorithms, called Algorithm 1 and Algorithm 2, for recovering information about the secret key of a block cipher. The algorithms exploit a biased probabilistic relation between the input and output of the cipher. This relation is called the (one-dimensional) linear approximation of the cipher. Mathematically, the problem of key recovery is a binary hypothesis testing problem that can be solved with appropriate statistical tools.

The same mathematical tools can be used for realising a distinguishing attack against a stream cipher. The distinguisher outputs whether the given sequence of keystream bits is derived from a cipher or a random source. Sometimes, it is even possible to recover a part of the initial state of the LFSR used in a key stream generator.

Several authors considered using many one-dimensional linear approximations simultaneously in a key recovery attack and various solutions have been proposed. In this thesis a unified methodology for using multiple linear approximations in distinguishing and key recovery attacks is presented. This methodology, which we call multidimensional linear cryptanalysis, allows removing unnecessary and restrictive assumptions. We model the key recovery problems mathematically as hypothesis testing problems and show how to use standard statistical tools for solving them. We also show how the data complexity of linear cryptanalysis on stream ciphers and block ciphers can be reduced by using multiple approximations.

We use well-known mathematical theory for comparing different statistical methods for solving the key recovery problems. We also test the theory in practice with reduced round Serpent. Based on our results, we give recommendations on how multidimensional linear cryptanalysis should be used.

KEYWORDS: multidimensional cryptanalysis, Matsui's algorithm, linear cryptanalysis, block cipher, stream cipher

TIIVISTELMÄ: Lineaarinen kryptoanalyysi on tärkeä työkalu symmetristen salainten turvallisuuden tutkimisessa. Matsui ehdotti 1993 kahta algoritmia, Algoritmit 1 ja 2, tiedon saamiseen lohkosalaimessa käytetystä salausavaimesta. Menetelmässä käytetään hyväksi salaimen selväkielen ja salakielen välistä tilastollista riippuvuutta, jota kutsutaan (yksiulotteiseksi) lineaarikseksi approksimaatioksi. Matemaattisesti avaimen paljastaminen tällä tavoin on mallinnettavissa binaariseksi hypoteesin testausongelmaksi, joka voidaan ratkaista sopivilla tilastollisilla menetelmillä.

Samaa menetelmää voidaan käyttää myös jonosalainta vastaan tehtävään erotteluhyökkäykseen. Tilastollinen erottelija kertoo onko annettu bittijono saatu salaimesta vai satunnaisesta lähteestä. Joissain tapauksissa on myös mahdollista selvittää osa avaingeneraattorissa käytettävän lineaarisen siirtorekisterin alkutilasta.

Monissa aiemmissa tutkimuksissa on pohdittu miten useampaa yksiulotteista lineaarista approksimaatiota voitaisiin käyttää samanaikaisesti ja on ehdotettu useita eri lähestymistapoja. Tässä työssä esitetään moniulotteiseksi lineaariseksi kryptoanalyysiksi kutsuttu yhtenäinen metodologia usean approksimaation samanaikaiseen käyttämiseen. Tämä metodologia mahdollistaa tarpeettomien ja rajoittavien oletuksien poistamisen. Työssä mallinnetaan avaimen paljastuksen ongelma matemaattisesti hypoteesin testausongelmana ja selvitetään oikeat tilastolliset menetelmät näiden ongelmien ratkaisemiseen. Lisäksi näytetään, että lineaariseen kryptoanalyysiin liittyvä datavaativuus pienenee, kun käytetään useampaa approksimaatioita yhtäaikaa.

Ongelman ratkaisemisessa käytettyjen tilastollisten menetelmien vertailussa sovelletaan tunnettua matemaattista teoriaa. Työssä kokeillaan myös teoriaa käytännössä supistetulla Serpent-salaimella. Saatujen tulosten perusteella voidaan suositella tehokkainta tapaa moniulotteisen lineaarisen kryptoanalyysin käyttämiseen.

AVAINSANAT: moniulotteinen kryptoanalyysi, Matsuin algoritmi, lineaarinen kryptoanalyysi, lohkosalain, jonosalain

CONTENTS

PREFACE

This thesis is a result from research I have conducted in the Cryptology group of Department of Information and Computer Science at the Aalto University School of Science and Technology. The research has been funded in 2006-2008 by the Finnish Scientific Advisory Board for Defence (MATINE) under project Mdd702 and in 2009-2010 by the Academy of Finland under project 122736. In 2009-2010 the research was partially funded by the Department of Information and Computer Science through funding for postgraduate studies. The work has also been supported by the Finnish Foundation for Technology Promotion TES through a personal grant for doctoral studies.

I thank my pre-examiners, Professor Thomas Johansson (Lund University) and Dr. Henri Gilbert (Orange Labs, France) for taking the time to read my thesis and for their comments and suggestions. I am grateful to Professor Anne Canteaut (INRIA, France) for the honour of having her as the opponent.

I am indebted to my supervisor Kaisa Nyberg for suggesting interesting and fruitful problems for studying as well as for the advice she has given me over the years. I also thank Joo Yeon Cho, the co-author of many of my publications. His practical experiments and knowledge of cryptology were crucial to this work.

The working environment in the Department of Information and Computer Science has been positive and encouraging. I thank all my colleagues especially in the Cryptology group for their help in cryptology, LaTeX and choosing the optimal time for lunch.

Last but not least I thank my family and friends for their support and encouragement.

Espoo, May 2010

Miia Hermelin

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning	See page
AES	Advanced Encryption Standard	36
DES	Data Encryption Standard	36
FFT	Fast Fourier Transform	61
HTP	Hypothesis testing problem	42
LFSR	Linear feedback shift register	39
LLR	Log-likelihood ratio	45
SPN	Substitution permutation network	37
c.d.f.	Cumulative distribution function	27
k.s.g.	Key stream generator	38
p.d.	Probability distribution	27
p.d.f.	Probability distribution function	27
s.i.	Statistically independent	28

LIST OF SYMBOLS AND NOTATIONS

MATHEMATICAL NOTATION

Notation	Meaning	See page
$*$	Convolution	27
$\oplus$	Component-wise addition modulo 2 (XOR)	26
$\bigoplus_{i=1}^{m}$	XOR of m binary vectors	26
$a = (a^1, \ldots, a^n)$	Binary vector in $\mathbb{F}_2^n$ or an n-bit integer	26
$a \cdot b$	Inner product in $\mathbb{F}_2^n$	26
ab	Multiplication of $a, b \in \mathbb{F}_2^n$ in $\mathbb{F}_2^n$	26
f	Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$	34
$f = (f_1, \ldots, f_m)$	m-dimensional vector Boolean function	34
$\mathbb{F}_2^n$	Vector space over the field $\mathbb{F}_2$	26
u_i	ith row of $U = (u_1, \ldots, u_m)^T$	65
U	Binary matrix	26
U^T	Transpose of U	26
$\hat{\varphi}$	Walsh-Hadamard transform of $\varphi : \mathbb{F}_2^n \mapsto \mathbb{R}$	27

STATISTICAL NOTATION

Notation	Meaning	See page
$\sim$	$\mathbf{X} \sim \mathcal{D} : \mathbf{X}$ follows $\mathcal{D}$	27
$\nsim$	$\mathbf{X} \nsim \mathcal{D} : \mathbf{X}$ does not follow $\mathcal{D}$	49
$\mathrm{Bernoulli}(p)$	Bernoulli distribution	29
$\mathrm{Bin}(N, p)$	Binomial distribution	29
$c(\mathbf{X}), c$	Correlation of $\mathbf{X}$	29
$C(p, q)$	Capacity between p.d.'s p and q	31
$C(p) = C(p, \theta)$	Capacity of p.d. p	31
$D(p\|q)$	Kullback-Leibler distance between p.d.'s p and q	30
$D^*(p, q)$	Chernoff information of p.d.'s p and q	31
$\mathcal{D},\ p$	Probability distribution	27
$f_{\mathbf{X}}(x),\ f(x)$	P.d.f.	27
$f_{\mathbf{X}}(x; \omega),\ f(x; \omega), p^{\omega}$	P.d.f. with parameter ω	28
$f_{\mathbf{X}_1, \ldots, \mathbf{X}_N}(x_1, \ldots, x_N)$	Joint p.d.f.	28
$F_{\mathbf{X}}(x),\ F(x)$	C.d.f	27
$\mathcal{FN}(\mu, \sigma^2)$	Folded normal distribution	29
H_i	Hypothesis	42
$I^{\lambda}(p : q)$	Divergence between p.d.'s p and q with parameter λ	31
LR	Likelihood ratio	44
LLR	Log-likelihood ratio	45
$\mathrm{LLR}(q; p^0, p^1)$	LLR-test to distinguish p^0 and p^1	45
$\mathcal{L}$	Likelihood function	43
$\mathrm{multi}(N, p)$	Multinomial distribution	30
M_{ω}	Mark for parameter value ω	52
$\mathcal{N}(0, 1)$	Normed normal p.d.	28
$\mathcal{N}(\mu, \sigma^2)$	Normal p.d. with mean μ and variance σ^2	28

$p = (p_0, \dots, p_M)$	P.d.f. (p.d.) if $\mathcal{X} = \{0, \dots, M\}$	27
P_e	Error probability	44
$\mathcal{P}$	Set of (alternative) p.d.'s	48 and 50
P_S	Success probability	59
q	Empirical p.d., realisation of $\mathbf{Q}$	30
$\mathbf{Q}$	Random vector of relative frequencies	30
$R^\lambda(\mathbf{Q} : p)$	Divergence test for null p.d. p	49
x	Realisation of $\mathbf{X}$	27
$\mathbf{X}$	Random variable	27
$\mathbf{X}_{(r)}$	rth order statistic	33
$\mathbf{X}_{i,n}$	nth term in the ith random sample	52
$\mathbf{X}_1, \dots, \mathbf{X}_N$	Random sample	28
$\mathcal{X}$	Sample space of $\mathbf{X}$	27
$\mathcal{X}^N$	Sample space of random sample	28
α	Type I Error, level of test	44
β	Type II Error, $1-$power of test	44
δ	Distinguisher	43
θ	Uniform p.d.	28
ρ	Empirical correlation	29
ϕ	P.d.f. of normed normal distribution	28
ϕ_{μ,σ^2}	P.d.f. of $\mathcal{N}(\mu, \sigma^2)$	28
Φ	C.d.f. of normed normal distribution	28
Φ_{μ,σ^2}	C.d.f of $\mathcal{N}(\mu, \sigma^2)$	28
χ^2_M	χ^2-distribution with M degrees of freedom	29
$\chi^2_M(\lambda)$	Non-central χ^2-distribution	29
$\chi^2(\mathbf{Q})$, $\chi^2(\mathbf{Q}; p)$	χ^2-test for null p.d. p	50

CIPHERS AND LINEAR CRYPTANALYSIS

Notation	Meaning	See page
$b(z)$	Test of Biryukov method for Alg. 1	68
$b_F(z)$	Test of full Biryukov method for Alg. 1	69
b_k	Mark of full Biryukov method for Alg. 2	77
$C_{\min}(\cdot)$	Minimum capacity in a set of p.d.'s	67
$c(a)$	Theoretical correlation of $a \cdot f$, $a \in \mathbb{F}_2^m$	69
$c(f, 0)$	Correlation of Boolean function f	34
$c_f(u_1, u_2; w)$	Correlation of $u_1 \cdot x_1 \oplus u_2 \cdot x_2 \oplus w \cdot f$	53
$\mathcal{C}$	Ciphertext space	22
$\mathcal{D}_R$, $\mathcal{D}_W$	Right/wrong key distribution	59
E_K	Encryption algorithm using K	22
$f(\cdot, k)$	Last round function	60
$G(z)$	Test of convolution method for Alg. 1	71
G_k	Mark of convolution method for Alg. 2	80
$h(z)$	Test of log-likelihood method (G-test) for Alg. 1	69
k	Last $((R+1)$th) round key candidate	60
k_0	Right last round key candidate	60
k_i	Round key of a block cipher, $i = 1, 2, \dots,$	36
K	Key of cipher	22

K_R	Key data over R rounds	56
$\mathcal{K}$	Key space	22
$l(z)$	Test of LLR-method for Alg. 1	71
L_k	Mark of LLR-method for Alg. 2	79
m	Dimension of the approximation	65
$\mathcal{M}$	Message space	22
N	Data complexity, size of random sample	24
p	P.d. of the multidimensional approximation	65
p^z	Permutation of p, determined by z	67
q^k	Empirical p.d. for key candidate k	75
$s(z)$	Test of χ^2-method for Alg. 1	70
S_k	Mark of χ^2-method for Alg. 2	77
x	Plaintext	22
y	Ciphertext (after R rounds)	22
y'	Ciphertext (after $R+1$ rounds)	22
$Y_0 = Y$	Initial state of an LFSR	39
$Y_t = (s_t, \ldots, s_{t+L-1})$	State of an LFSR at time t	39
z_t	Output of a k.s.g at time t	38
z	Inner key class	67
z_0	Right inner key class	67
Δ_M	Set of binary vectors, M first bits non-zero	63
π_S, π_P	Substitution and permutation functions	37
$\rho(a)$	Empirical correlation of $a \cdot f$, $a \in \mathbb{F}_2^m$	69

LIST OF PUBLICATIONS

This thesis consists of the summary and the following articles, which are referred to in the text with Roman numerals:

I. **Multidimensional Walsh Transform and a Characterization of Bent Functions.** Kaisa Nyberg and Miia Hermelin. In P. Vijay Kumar Tor Helleseth and Oyvind Ytrehus, editors, Proceedings of the 2007 IEEE Information Theory Workshop on Information Theory for Wireless Networks, IEEE, pages 83–86, 2007.

II. **Multidimensional Linear Distinguishing Attacks and Boolean Functions.** Miia Hermelin and Kaisa Nyberg. In Preproceedings of Fourth International Workshop on Boolean Functions: Cryptography and Applications BFCA'08, 2008. Publications des Universités de Rouen et du Havre. Proceedings available on-line: http://www.liafa.jussieu.fr/bfca/

III. **Multidimensional Linear Cryptanalysis of Reduced Round Serpent.** Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, Information Security and Privacy, 13th Australasian Conference, ACISP 2008 Wollongong, Australia, July 7-9, 2008, Proceedings, volume 5107 of LNCS, pages 203–215. Springer, 2008.

IV. **Multidimensional Extension of Matsui's Algorithm 2.** Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. In Orr Dunkelman, editor, 16th International Workshop, FSE 2009 Leuven, Belgium, February 22-25, 2009 Revised Selected Papers, volume 5665 of Lecture Notes in Computer Science, pages 209–227. Springer, 2009.

V. **Statistical Tests for Key Recovery Using Multidimensional Extension of Matsui's Algorithm 1.** Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. EUROCRYPT'09 POSTERSESSION, 2009. Also appeared in Helena Handschuh, Stefan Lucks, Bart Preneel, and Phillip Rogaway, editors, Symmetric Cryptography, number 09031 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Available at http://drops.dagstuhl.de/opus/portals/index.php?semnr=09031

VI. **Improved linear cryptanalysis of SOSEMANUK.** Joo Yeon Cho and Miia Hermelin. In The 12th International Conference on Information Security and Cryptology 2009, ICISC'09, Seoul, Korea, Lecture Notes in Computer Science, to appear.

VII. **Dependent Linear Approximations - The Algorithm of Biryukov and Others Revisited.** Miia Hermelin and Kaisa Nyberg. In J. Pieprzyk, editor, CT-RSA - RSA Cryptographers' Track 2010 (CT-RSA 2010), San Francisco, USA, volume 5985 of Lecture Notes in Computer Science, pages 318–333. Springer, 2010.

1 INTRODUCTION

1.1 BACKGROUND AND PREVIOUS WORK

Linear cryptanalysis exploits the statistical dependence between the input and output of a cipher. A one-dimensional linear approximation of a cipher is a linear combination of its input and output bits. The approximation can be interpreted as a binary random variable. The goal is to find a linear approximation with a large correlation in absolute value. It is then possible to find information about the cipher using statistical methods. The goal of this work is to create a theoretically sound framework for using multiple one-dimensional approximations simultaneously. We call this new methodology multidimensional linear cryptanalysis.

Matsui invented linear cryptanalysis in 1993 [31]. He presented two algorithms, Algorithm 1 (Alg. 1) and Algorithm 2 (Alg. 2) that can be used for finding one bit of information about the secret key of a block cipher, provided that the attacker has enough plaintext-ciphertext pairs. Alg. 2 can also be used for determining several bits of the last round key of a block cipher. The amount of data needed for successfully realising the attack is called the *data complexity* of the attack. Matsui showed that the data complexity is determined by the correlation of the approximation.

Matsui's algorithms were designed for block ciphers. It is sometimes possible to realise a key or initial state recovery attack against a stream cipher, but in most cases only distinguishing attacks are possible to realise. The output of a keystream generator, used in a stream cipher, should look random. In distinguishing attacks the goal is to determine whether a given sequence is produced by a cipher or a random source. While distinguishing attacks are not as strong as key recovery attacks, finding a good distinguisher implies a weakness in the cipher. At present, one of the most important security criteria for a symmetric encryption algorithm is its resistance against linear cryptanalysis.

Matsui suggested using two approximations simultaneously in 1994 [30]. In the same year, Kaliski and Robshaw used several approximations in an attempt to reduce the data complexities of Matsui's algorithms [8]. Biryukov, et al., [6] used multiple approximations for finding several bits of the secret key with reduced data complexity in 2004. However, the theoretical foundations of the methods by Kaliski and Robshaw and Biryukov, et al., both depend on assumptions about the statistical properties of the one-dimensional linear approximations. In particular, they assumed that the one-dimensional linear approximations are statistically independent. Murphy pointed out that the assumption may not hold in a general case [33].

Baignères, et al., presented in 2004 a linear distinguisher that does not suffer from this limitation [1]. The distinguisher has also another advantage over the previous approaches: it is based on a well established statistical theory of log-likelihood ratio (LLR). Unfortunately, the authors did not provide an efficient way for determining the probability distribution that is needed in their method. Englund and Maximov presented computational methods for determining the distribution directly [21], but they are in general not feasible for handling distributions of larger than 32-bit values.

In the articles **I–VII** we consider the following problems and attempt to solve them:

- How can we realise the multidimensional distinguisher in practice?
- How is key recovery realised using multidimensional methods?
- How can we measure the efficiency of different methods?
- Is there an optimal way for exploiting multiple approximations and what is it?
- What kind of theoretical restrictions we get for ciphers?
- Other than distinguishing attack, can we use multiple approximations for attacking a stream cipher?
- The method by Biryukov, et al., which we call the Biryukov method for simplicity, relies on the assumption of statistical independence. If the assumption is true, can we verify it?
- Biryukov, et al., also proposed an enhancement to their method [6]. What is the mathematical justification for this enhancement and how is it related to the assumption of statistical independence?
- What is the difference between the multidimensional method and the previous methods, especially, is the multidimensional method better than the one-dimensional or the Biryukov method?

To answer these questions, we have to find a proper statistical model for the key recovery problem. While the mathematical tools used in the papers **I–VII** are well-known in statistics, many of them have not been applied to linear cryptanalysis before.

The Alg.1 type of key recovery is related to coding theory and pattern recognition theory. In coding theory we send a message, a codeword, over a noisy channel. The noise can distort the message to some other codeword with some probability. The same message is sent several times in order to recover the original message with high probability of success. The number of required repetitions is the data complexity.

In coding theory and pattern recognition, the error induced by the noise is small and the data complexity is relatively small. Therefore, the methods can be tested on real systems. The main interest is in efficiently recovering the original message or classifying the observed data.

In cryptanalysis, the error probability is large and the data complexity is the main criterion for measuring the success of an attack. The data complexities of ciphers currently considered safe are so large that true attacks against them are not feasible. Consequently, only parts of the ciphers can be analysed in practice. However, attacks that are unfeasible now can be feasible later and the cipher designers must consider the future and the development of computing power. The emphasis in linear cryptanalysis is in determining the data complexity and finding methods for which the data complexity is minimised. Therefore, while linear cryptanalysis, coding theory and pattern recognition use similar statistical tools, their goals and interests are different.

1.2 RESULTS OF THE THESIS AND AUTHOR'S CONTRIBUTION

In this section we briefly describe the main results of the articles **I– VII** and the contributions of the author of this thesis.

Publication I: We present a new concept of multi-bent functions and show that they are equivalent to vector bent functions defined in the classical way. Using this new definition of bent functions we can show that these functions are optimal against multidimensional linear cryptanalysis.

The multi-Walsh transform was developed as a theoretical tool for handling multidimensional probability distributions. The author helped in developing the theory. The co-author was the main author of this paper.

Publication II: We show how one-dimensional approximations can be used for determining the multidimensional probability distribution. Baignères, et al., showed that the efficiency of a multidimensional distinguisher is determined by its distance from the uniform distribution. We call this distance the capacity, due to Biryukov, et al., [6] and we show how the one-dimensional correlations determine the capacity.

We calculate the capacities for some multidimensional approximations of Boolean functions and keystream generators. We show concrete examples where the approximations are statistically dependent while being linearly independent. Moreover, we show how using multiple approximations increases the efficiency of linear cryptanalysis. We also consider the problem of chaining the multidimensional linear approximations.

The author did practical experiments on the filter generator example presented in Section 5.1 and made the interesting observation that all the one-dimensional correlations are equal in absolute value. This remark prompted the theoretical results in the paper. The paper was written together with the co-author.

Publication III: We consider Matsui's Alg. 1 and propose a truly multidimensional approach where we measure the distance between the empirical and theoretical distributions. Collard, et al., studied the Biryukov method and made practical experiments on reduced round block cipher Serpent [11]. We use this setting to compare the multidimensional method to the Biryukov method. The experiments show that the multidimensional method is more certain to yield the correct key with given amount of data.

The author was responsible for writing most of the article and developing the statistical theory and method proposed in the paper. Cho did the practical experiments with Serpent.

Publication IV: Next we consider extending Matsui's Alg. 2 to multiple dimensions. We study two statistical settings, one based on goodness-of-fit tests and one based on parametric hypothesis testing problems, which can be solved with the log-likelihood ratio (LLR). We show that the enhancement of the Biryukov method can be regarded as a goodness-of-fit test.

Selçuk presented the concept of advantage for measuring the efficiency of one-dimensional Alg. 2 [40]. We extend the theory to multiple dimensions to compare the different methods. We derive the advantage as a function of the data complexity in theory and in practice using the reduced round Serpent. The results show that the LLR-based method is more efficient than the goodness-of-fit test.

The author developed the statistical framework, did the theoretical calculations and had the main responsibility in the writing of the article. The experiments were designed and implemented by Cho.

Publication V: Similarly as for Alg. 2, we consider two different statistical settings for Alg. 1: the goodness-of-fit problem and a parametric hypothesis problem, solved with LLR. We show that the method in **III** is equivalent to the goodness-of-fit solution.

We propose extending the concept of advantage to multidimensional Alg. 1 for comparing the methods. However, the use of advantage with Alg. 1 requires an artificially strong assumption. Due to this assumption, the theoretical predictions about the LLR, which in theory gives the optimal method, seem to be slightly pessimistic when compared to the empirical results. For the goodness-of-fit setting, the difference between the empirical results and theoretical predictions is notably larger. We clarify this disagreement in Section 7.4 of the summary part of this thesis.

The author is responsible for the statistical derivations and had the main responsibility in the writing of the article. Cho made the practical experiments.

Publication VI: Berbain, et al., presented a method where one one-dimensional linear approximation of the stream cipher Grain could be used for the initial state recovery of the linear feedback shift register used in the cipher [4]. We extend the idea in this paper by showing how multiple one-dimensional approximations can be used for making a similar attack against the stream cipher SOSEMANUK more efficient.

The author's contribution is in the theoretical part of the paper, in developing the attack and refining Section 3.3. The main author is Cho, who also did the experiments.

Publication VII: We propose yet another Alg. 1. method, called the convolution method. We show that the Biryukov method can be enhanced to what we call a full Biryukov method and that this enhancement is equivalent to the convolution method. Therefore, the assumption about statistical independence is not required for the full Biryukov method. On the other hand, we show how it is possible to verify the assumption of statistical independence when necessary.

We also give a proper statistical framework for Alg. 1 and show how different methods can be compared. We show that under certain conditions, which hold for practical ciphers, the convolution method, the full Biryukov method and all the other Alg. 1 methods we consider in **III** and **V** have the same data complexities. The empirical tests done on Serpent verified the theoretical results. Since the convolution method has the smallest time complexity, we conclude that it is the most efficient of these methods in practice.

The relationship between the full Biryukov method and convolution method and also all the statistical derivations are due to the author. The author was responsible for writing the article.

1.3 THESIS OUTLINE

This thesis consists of a summary and seven publications, which are appended to this thesis. The summary gives a unified view to the methodology of multidimensional linear cryptanalysis. The main new results of this thesis are presented in Chapters 7 and 8. The remainder of this summary is structured as follows:

Chapter 2 presents an overview of cryptography and cryptanalysis.

Chapter 3 introduces the basic notations, definitions and theory about finite fields, probability theory and Boolean functions.

Chapter 4 studies stream and block ciphers.

Chapter 5 presents the statistical tools used in the thesis. Although the theory is well-known, many of the tools have not been used in linear cryptanalysis before.

Chapter 6 recalls the different methods used in one-dimensional linear cryptanalysis, including distinguishing attacks and Matsui's algorithms .

Chapter 7 discusses the multidimensional linear cryptanalysis attacks. The chapter concludes the results of the articles **III- VII**. In addition, it considers some practical aspects of implementing the Matsui's algorithms.

Chapter 8 considers some applications of multidimensional linear cryptanalysis and its theoretical bounds. It is based on articles **I- II**.

Chapter 9 draws conclusions and suggests future work.

2 CRYPTOGRAPHY

This chapter introduces the basic concepts used in cryptography. In Section 2.1, we define *cryptosystems* and *symmetric cryptography*. *Cryptanalysis* is the theory of security analysis of cryptographic systems [44]. We consider different methods for cryptanalysis, the outcomes, assumptions and measuring the efficiency of the methods in Section 2.2.

2.1 CRYPTOSYSTEMS

Cryptography enables two parties, called the sender and receiver, to transmit messages over an insecure channel without a third party, called the attacker, being able to understand the messages. The sender encrypts the message using an encryption algorithm and some predetermined data called the encryption key. The original and encrypted messages are called the plaintext and ciphertext, respectively. The receiver deciphers the ciphertext using a decryption algorithm and a secret decryption key. A cryptographic system, or a cryptosystem, can be defined as follows:

Definition 2.1. A cryptosystem consists of the following:

- The message space $\mathcal{M}$: a set of strings over some alphabet. An element of $\mathcal{M}$ is called a plaintext message.
- The ciphertext space $\mathcal{C}$: a set of strings over some alphabet that maybe different from the message space alphabet. An element of $\mathcal{C}$ is called a ciphertext.
- Sets $\mathcal{K}$ and $\mathcal{K}'$: the encryption and decryption key space consisting of possible encryption and decryption keys, respectively.
- A set $\{E_K : K \in \mathcal{K}\}$ of encryption algorithms or encryption functions: For each key $K \in \mathcal{K}$ there is a unique bijection E_K from $\mathcal{M}$ to $\mathcal{C}$.
- A set $\{D_{K'} : K' \in \mathcal{K}'\}$ of decryption algorithms or decryption functions: For each key $K' \in \mathcal{K}'$ the function $D_{K'}$ is a bijection from $\mathcal{C}$ to $\mathcal{M}$.

For each encryption key $K \in \mathcal{K}$ there is a unique decryption key $K' \in \mathcal{K}'$ such that $D_{K'}(E_K(x)) = x$ for all plaintexts $x \in \mathcal{M}$.

The *Kerchoff's principle* states that the security of the system should reside only in the secret key. Hence, the attacker knows the message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, the encryption key space $\mathcal{K}$, the decryption key space $\mathcal{K}'$ and the sets $\{E_K : K \in \mathcal{K}\}$ and $\{D_{K'} : K' \in \mathcal{K}'\}$. If the encryption key K is also public, the cryptosystem is called asymmetric or public. The security of the system depends only on the decryption key K', known only to the receiver.

In a symmetric cryptosystem the encryption and decryption keys are equal or can be easily derived from each other. Hence, only the sender and the receiver should know the key. There are two types of symmetric encryption schemes: block ciphers and stream ciphers. They will be studied more closely in Chapter 4. We will next study different concepts about cryptanalysis.

2.2 CRYPTANALYSIS

2.2.1 Attack Scenarios

Different attack scenarios can be realised depending on the information available to the attacker. Some scenarios are listed below. The goal is to recover the plaintext or to find information about the secret key.

Ciphertext-only attack: the attacker has only access to the ciphertext. If a cipher is vulnerable to this type of attack, it is considered insecure.

Known-plaintext attack: the attacker has a quantity of plaintext and corresponding ciphertext.

Chosen-plaintext attack: the attacker can choose the plaintext and is given the corresponding ciphertext.

Chosen-ciphertext attack: the attacker selects the ciphertext and is then given the corresponding plaintext. The objective of this scenario is to deduce the plaintext from different ciphertext or to find information about the secret key.

2.2.2 Outcomes of an Attack

In a ciphertext-only attack, the attacker tries to determine the plaintext corresponding to the ciphertext. However, once the plaintext is known, there is other information available for the attack. In the worst case, the attacker can recover the secret key of the cipher and find all the information sent using the corrupted key. Even if it is not possible to find the whole key, some information may still be revealed to the attacker. Different levels of breaking a cipher are listed below:

Total break: attacker finds the secret key

Instance deduction: attacker gets a clone of $D_{K'}$. Hence, while K' remains unknown, the attacker can decrypt any message.

Key information deduction: attacker gets partial information about the key.

Distinguishing: attacker can distinguish the cipher from a purely random function.

The list is hierarchical: total break means that the attacker can also realise any of the other attack levels etc.

2.2.3 Attack Methods

There are several different attack methods that can be used against symmetric ciphers. The *exhaustive search* assumes no knowledge of the inner structure of the cipher. The attacker tries each key exhaustively, until the right key is found. All stream and block ciphers except the *one-time pad*, see Section 4.2.1, can in theory be attacked by exhaustive search but in practice, the key space is too large.

In algebraic attacks the whole cipher is expressed as a large system of multivariate algebraic equations, which have to be solved in order to recover the secret key [13]. Obviously, solving such large systems is difficult.

A good cipher should imitate the behaviour of a perfectly random function. If the cipher has detectable non-random behaviour, the attacker can use *statistical cryptanalysis* in realising a distinguishing attack. Sometimes key information deduction is also possible. Examples of different attack types used in statistical cryptanalysis include for example differential, linear, integral and correlation attacks.

2.2.4 Attack Parameters

When realising the attack, the attacker must also consider the *cost* of the attack. If the cost of breaking the cipher on some level is too high, the attack is not considered successful. The cost consists of the amount of memory, time and data that are needed for successfully realising the attack. The different parameters, some of which may depend on the other parameters, are given below:

Time complexity: the amount of computation time (in given units) required to perform the attack successfully. Often the time complexity of an attack is compared to that of the exhaustive search.

Data complexity: the amount of data (ciphertext, keystream, plaintext-ciphertext pairs etc.) needed for attack

Memory complexity: the amount of memory units needed to store for the attack

Success probability: the probability of successfully breaking the cipher. This parameter is needed in statistical cryptanalysis.

The total complexity of the attack is not easy to define. It can be the sum of the time, memory and data complexities (with a given success probability) or the largest of them. There is usually some *trade-off* between the parameters. For example, an attempt to decrease the time complexity may increase the data complexity or decrease the success probability. A formula describing the trade-off makes it easier to compare the complexities of different attacking methods.

There is no general definition for when a cipher is broken. The average time needed for finding an l-bit key with exhaustive search is $\mathcal{O}(2^l)$. Therefore, the desired strength of the cipher gives a lower bound to the size of the key space. For contemporary ciphers l is usually at least 100. A general criterion for ciphers is that there should not be an attack that has lower complexity than the exhaustive search or other known generic attacks, that is, an attack is *successful* if it has lower complexity than the exhaustive search. This applies to all levels of break, including the distinguishing attack, though for example Rose and Hawkes criticised that distinguishing attacks are not a practical threat [38].

In this thesis, we consider different ways for applying linear cryptanalysis to symmetric ciphers. We assume that we are given a large number of plaintext-ciphertext pairs. Using proper statistical tools, we aim at distinguishing or key information deduction.

3 MATHEMATICAL PRELIMINARIES

In this chapter the basic notation and definitions needed in the rest of the thesis are given. The reader is assumed to be familiar with basic mathematical theory about statistics and finite fields. First, we recall facts about finite fields in Section 3.1. Then, in Section 3.2 we study briefly the Walsh-Hadamard transform. Section 3.3 is devoted to probability theory and statistics. The necessary information about Boolean functions is given in Section 3.4.

3.1 SOME PROPERTIES OF FINITE FIELDS

General theory of finite fields $GF(p^n)$ with p^n elements, where p is prime, is covered in [29]. The finite field $GF(2^n)$ can be identified the with space of n-dimensional binary vectors $\mathbb{F}_2^n$. If $a = (a^1, \ldots, a^n) \in \mathbb{F}_2^n$ and $b = (b^1, \ldots, b^n) \in \mathbb{F}_2^n$, the operation $\oplus$ is the component-wise modulo 2 sum (XOR): $a \oplus b = (a^1 \oplus b^1, \ldots, a^n \oplus b^n)$. We denote $\bigoplus_{i=1}^m a_i = a_1 \oplus \cdots \oplus a_m$.

The *inner product* for $a = (a^1, \ldots a^n), b = (b^1, \ldots, b^n) \in \mathbb{F}_2^n$ is defined as $a \cdot b = a^1 b^1 \oplus \cdots \oplus a^n b^n$. Then the vector a is called the (linear) *mask* of b.

Let L be a linear mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$. Then for all linear masks $b \in \mathbb{F}_2^n$ it holds that

$$b \cdot Lx = L^T b \cdot x, \text{ for all } x \in \mathbb{F}_2^n,$$

where L^T is the transpose of the linear mapping L. Hence, we obtain for each L and b a unique mask $b_L \in \mathbb{F}_2^n$ that satisfies

$$b \cdot Lx = b_L \cdot x, \text{ for all } x \in \mathbb{F}_2^n. \tag{3.1}$$

This linear transformation property is needed for example in Section 6.2.3.

The multiplication by a fixed element in $\mathbb{F}_2^n$ is a linear operation. For all $a \in \mathbb{F}_2^n$ there exists a unique $n \times n$ binary matrix U_a such that

$$ax = U_a x, \text{ for all } x \in \mathbb{F}_2^n.$$

Hence, by (3.1), for all $a, b \in \mathbb{F}_2^n$ there is a unique mask b_a such that

$$b \cdot ax = b \cdot U_a x = U_a^T b \cdot x = b_a \cdot x, \text{ for all } x \in \mathbb{F}_2^n. \tag{3.2}$$

The binary vector $a = (a^1, \ldots, a^n) \in \mathbb{F}_2^n$ can be identified with a unique integer $b \in \mathbb{N}$ using the formula

$$b = \sum_{i=1}^{n} a^i 2^{i-1}.$$

Hence, a is used interchangeably to notate both a binary vector and the corresponding integer.

3.2 WALSH TRANSFORMS

We recall the following facts about Walsh transforms. Let $\varphi : \mathbb{F}_2^m \mapsto \mathbb{R}$ be a real-valued function. The *Walsh-Hadamard transform* $\hat{\varphi}$ of φ is defined as

$$\hat{\varphi}(a) = \sum_{\eta \in \mathbb{F}_2^m} \varphi(\eta)(-1)^{\eta \cdot a}, \; a \in \mathbb{F}_2^m. \tag{3.3}$$

Then $\varphi(\eta) = 2^{-m}\hat{\hat{\varphi}}(\eta), \eta \in \mathbb{F}_2^m$, using the inverse of Walsh-Hadamard transform. The *convolution* of two functions $\varphi : \mathbb{F}_2^m \mapsto \mathbb{R}$ and $\psi : \mathbb{F}_2^m \mapsto \mathbb{R}$ is defined as

$$(\varphi * \psi)(\eta) = \sum_{\zeta \in \mathbb{F}_2^m} \varphi(\zeta)\psi(\eta \oplus \zeta), \; \eta \in \mathbb{F}_2^m. \tag{3.4}$$

It is straightforward to verify that then

$$\widehat{(\varphi * \psi)}(a) = \hat{\varphi}(a)\hat{\psi}(a), \; a \in \mathbb{F}_2^m. \tag{3.5}$$

Parseval's theorem states that

$$2^m \sum_{\eta \in \mathbb{F}_2^m} \varphi(\eta)^2 = \sum_{a \in \mathbb{F}_2^m} \hat{\varphi}(a)^2. \tag{3.6}$$

3.3 STATISTICS

This section introduces the notation and main concepts used in statistics.

3.3.1 Probability Theory

We denote *random variables* $\mathbf{X}, \mathbf{Y}, \ldots$ by capital boldface letters and their *sample spaces* by $\mathcal{X}, \mathcal{Y}, \ldots$. The *realisations* $x \in \mathcal{X}, y \in \mathcal{Y}, \ldots$ of random variables $\mathbf{X}, \mathbf{Y}, \ldots$ are denoted by small letters.

Let $\mathbf{X}$ be a random variable with sample space $\mathcal{X}$. If $\mathbf{X}$ follows *probability distribution* (p.d.) $\mathcal{D}$, we denote $\mathbf{X} \sim \mathcal{D}$. The *cumulative distribution function* (c.d.f.) of $\mathbf{X}$, denoted by $F_{\mathbf{X}}(x)$, is given by

$$F_{\mathbf{X}} = \Pr_{\mathcal{D}}(\mathbf{X} \leq x), \text{ for all } x \in \mathcal{X}.$$

We omit the subscripts $\mathbf{X}$ and $\mathcal{D}$, if they are clear from the context.

The *probability density function* (p.d.f.) of a discrete random variable $\mathbf{X} \sim \mathcal{D}$ is given by

$$f_{\mathbf{X}}(x) = \Pr_{\mathcal{D}}(\mathbf{X} = x), \text{ for all } x \in \mathcal{X}.$$

We denote $f(x)$ and $\Pr(x)$, if $\mathbf{X}$ and $\mathcal{D}$ are clear from the context

Let $\mathbf{X}$ be a discrete random variable with sample space $\mathcal{X} = \{0, 1, \ldots, M\}$ for some integer $M \geq 0$. In this thesis, we denote the p.d.f. of $\mathbf{X}$ by the vector $p = (p_0, \ldots, p_M)$, whose components satisfy $p_x = \Pr(x)$ for all $x \in \mathcal{X}$. Moreover, we identify the p.d. and p.d.f. of $\mathbf{X}$ and call p the p.d. of $\mathbf{X}$.

If $\mathbf{X}$ is a continuous random variable with a continuous c.d.f. $F_{\mathbf{X}}(x)$, the p.d.f. of $\mathbf{X}$ is the function denoted by $f_{\mathbf{X}}(x)$ that satisfies

$$F_{\mathbf{X}}(x) = \int_{-\infty}^{x} f_{\mathbf{X}}(t)\,\mathrm{dt}, \text{ for all } x \in \mathcal{X}.$$

We use notation $F(x)$ and $f(x)$, if $\mathbf{X}$ is clear from the context.

If the p.d.f. of $\mathbf{X}$ depends on the parameter ω, we denote $f_{\mathbf{X}}(x;\omega)$, $f(x;\omega)$ or p^{ω}. The *uniform distribution* is denoted by θ.

3.3.2 Statistical Independence

Let $\mathbf{X}_1,\ldots,\mathbf{X}_N$, be random variables and $x_1,\ldots,x_N$, be the corresponding observations, where each $x_t \in \mathcal{X}$. Let each $\mathbf{X}_t$ have p.d.f. $f_{\mathbf{X}_t}(x_t)$. The joint sample space and joint p.d.f. of the random vector $(\mathbf{X}_1,\ldots,\mathbf{X}_N)$ is denoted by $\mathcal{X}^N$ and $f_{\mathbf{X}_1,\ldots,\mathbf{X}_N}(x_1,\ldots,x_N)$, respectively. We define the *statistical independence* of random variables as usual:

Definition 3.1. Let $\mathbf{X}_1,\ldots,\mathbf{X}_N$ be random variables with joint p.d.f. $f_{\mathbf{X}_1,\ldots,\mathbf{X}_N}(x_1,\ldots,x_N)$. For each $t = 1,\ldots,N$, let $f_{\mathbf{X}_t}(x_t)$ be the (marginal) p.d.f. of $\mathbf{X}_t$. Then the random variables are (mutually) statistically independent (s.i.), if for every realisation $(x_1,\ldots,x_N) \in \mathcal{X}^N$,

$$f_{\mathbf{X}_1,\ldots,\mathbf{X}_N}(x_1,\ldots,x_N) = \prod_{t=1}^{n} f_{\mathbf{X}_t}(x_t).$$

If random variables $\mathbf{X}_1,\ldots,\mathbf{X}_N$ are s.i., then each pair $(\mathbf{X}_i,\mathbf{X}_j)$, where $i,j \in \{1,\ldots,N\}$ and $i \neq j$, are s.i. However, the converse does not hold: pairwise statistical independence does not imply that the random variables $\mathbf{X}_1,\ldots,\mathbf{X}_N$ are s.i.

If random variables $\mathbf{X}_1,\ldots,\mathbf{X}_N$ are independent and identically distributed (i.i.d.) with the same p.d.f. f, they form a *random sample* from the population f. Their joint p.d.f. is then

$$f_{\mathbf{X}_1,\ldots,\mathbf{X}_N}(x_1,\ldots,x_N) = \prod_{t=1}^{N} f(x_t). \tag{3.7}$$

A *statistic* is a function of a random sample $\mathbf{X}_1,\ldots,\mathbf{X}_N$:

Definition 3.2. Let $\mathbf{X}_1,\ldots,\mathbf{X}_N$, be i.i.d. random variables and $g(x_1,\ldots,x_N)$, be a real- or vector-valued function defined on their joint sample space. Then the random variable or random vector $g(\mathbf{X}_1,\ldots,\mathbf{X}_N)$ is called a statistic.

3.3.3 Some Continuous Distributions

The normed *normal distribution* with mean 0 and variance 1 is denoted by $\mathcal{N}(0,1)$. Its p.d.f. is

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

The c.d.f. of a normally distributed random variable is denoted by $\Phi(x)$. The normal distribution with mean μ and variance σ^2 is denoted by $\mathcal{N}(\mu,\sigma^2)$ and its p.d.f. and c.d.f. are ϕ_{μ,σ^2} and Φ_{μ,σ^2}, respectively.

If $\mathbf{X} \sim \mathcal{N}(\nu, \sigma^2)$, then the absolute value $|\mathbf{X}|$ has the *folded normal distribution*, denoted by $\mathcal{FN}(\mu, \sigma)$. Its p.d.f. $f_{|\mathbf{X}|}$ is given by

$$f_{|\mathbf{X}|}(x) = \frac{1}{\sigma\sqrt{2\pi}}\left(e^{-\frac{(x-\mu)^2}{2\sigma^2}} + e^{-\frac{(x+\mu)^2}{2\sigma^2}}\right),\ x \geq 0.$$

The mean and variance of $|\mathbf{X}|$ are

$$\begin{aligned} E(|\mathbf{X}|) &= \mu(1 - 2\Phi(-\mu/\sigma)) + 2\sigma\phi(\mu/\sigma) \\ \mathrm{Var}(|\mathbf{X}|) &= \mu^2 + \sigma^2 - E(|X|)^2. \end{aligned}$$

The χ^2_M*-distribution* with M degrees of freedom has mean M and variance $2M$. The non-central $\chi^2_M(\lambda)$-distribution with M degrees of freedom has mean $\lambda + M$ and variance $2(M + 2\lambda)$. If $M > 30$, we may approximate $\chi^2_M(\lambda) \approx \mathcal{N}(\lambda + M, 2(M + 2\lambda))$ [16].

3.3.4 Discrete Random Variables and Their Probability Distributions

Binary Random Variables

A binary random variable $\mathbf{X}$ taking on values in $\{0, 1\}$ is Bernoulli(p)-distributed, if $\Pr(\mathbf{X} = 0) = p$. The correlation of $\mathbf{X}$ is

$$c(\mathbf{X}) = 2\Pr(\mathbf{X} = 0) - 1 = 2p - 1. \tag{3.8}$$

We denote c if $\mathbf{X}$ is clear from context. The bias of $\mathbf{X}$ is $\varepsilon = \frac{1}{2}c$.

Let $\mathbf{X}_1, \ldots, \mathbf{X}_N$ be a binary random sample from population Bernoulli(p). Then the statistic

$$\mathbf{Y} = \sum_{t=1}^{N} \mathbf{X}_t, \tag{3.9}$$

follows the *binomial distribution* $\mathrm{Bin}(N, p)$ with mean Np and variance $Np(1 - p)$. If N is large, then $\mathrm{Bin}(N, p) \approx \mathcal{N}(Np, Np(1 - p))$. The realisation of the statistic $N^{-1}(\mathbf{Y}, 1 - \mathbf{Y})$ is the vector $q = (q_0, q_1)$ defined by

$$q_0 = N^{-1}\#\{t = 1, \ldots, N : \mathbf{X}_t = 0\} \quad \text{and} \quad q_1 = 1 - q_0. \tag{3.10}$$

We say that q is the *empirical p.d.* of the random sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$. Respectively, we call the realisation $\rho = 2q_0 - 1$ of the statistic $\mathbf{T} = 2N^{-1}\mathbf{Y} - 1$ the *empirical correlation* of the random sample. If $|c|$ is small and N is large, then $\mathbf{T}$ is asymptotically normal with mean c and variance $(1-c^2)/N \approx 1/N$. Moreover, $|\mathbf{T}| \sim \mathcal{FN}(c, 1/\sqrt{N})$.

Consider now m binary random variables $\mathbf{X}_1, \ldots, \mathbf{X}_m$. The Piling Up lemma [31] has traditionally been used in calculating correlations of linear combinations of statistically independent random variables. The Piling Up lemma has also a converse that offers a natural criterion for verifying statistical independence of binary random variables. We give the Piling Up lemma and its converse as the following theorem:

Theorem 3.3. *Let $m \geq 2$ be an integer. The binary random variables $\mathbf{X}_1, \ldots, \mathbf{X}_m$, with correlations $c_i = c(\mathbf{X}_i)$, $i = 1, \ldots, m$ are s.i., if and only* *if for all index sets $I \subset \{1, 2, \ldots, m\}$, the correlation of the sum over indexes in I satisfies*

$$c(\bigoplus_{i \in I} \mathbf{X}_i) = \prod_{i \in I} c_i. \tag{3.11}$$

The proof is given in Appendix A in **VII**.

Multinomial Distribution

Consider a discrete random sample $\mathbf{X}_1, \ldots, \mathbf{X}_N$ taken from a sample space $\mathcal{X} = \{0, 1, \ldots, M\}$ using p.d. $p = (p_0, \ldots, p_M)$, and let $x_1, \ldots, x_N$, be the corresponding observations. Denote the joint sample space of $(\mathbf{X}_1, \ldots, \mathbf{X}_N)$ by $\mathcal{X}^N$.

Let $\mathbf{Q} = (\mathbf{Q}_0, \ldots, \mathbf{Q}_M)$ be a random vector, whose components

$$\mathbf{Q}_\eta = N^{-1}\#\{t = 1, \ldots, N : \mathbf{X}_t = \eta\}, \text{ for all } \eta \in \mathcal{X}, \tag{3.12}$$

are the relative frequencies of η's in the sequence $\mathbf{X}_1, \ldots, \mathbf{X}_N$. The vector $q = (q_0, \ldots, q_M)$, where

$$q_\eta = N^{-1}\#\{t = 1, \ldots, N : x_t = \eta\}, \text{ for all } \eta \in \mathcal{X}, \tag{3.13}$$

is a realisation of $\mathbf{Q}$. Similarly as for binary random variables, we call q the *empirical p.d.* of the random sample. All possible empirical p.d.'s, whose components satisfy $Nq_\eta \in \mathcal{X}$, for each $\eta \in \mathcal{X}$ and $\sum_{\eta \in \mathcal{X}} q_\eta = 1$, form the sample space $\mathcal{Q}$ of $\mathbf{Q}$.

The random vector $\mathbf{Q}$ has *multinomial distribution*, denoted by $\text{multi}(N, p)$, with probabilities

$$\Pr(\mathbf{Q} = q) = \frac{N!}{\prod_{\eta \in \mathcal{X}} (q_\eta N)!} \prod_{\eta \in \mathcal{X}} p_\eta^{N q_\eta}, \text{ for all } q \in \mathcal{Q}. \tag{3.14}$$

The following lemma gives the asymptotic distribution of any linear combination of the components of $\mathbf{Q}$ [43].

Lemma 3.4. *Let $\mathbf{Q} \sim \text{multi}(N, p)$. Let Λ_η, $\eta = 0, \ldots, M$ be any real numbers. Then the linear combination $N \sum_{\eta=0} \Lambda_\eta \mathbf{Q}_\eta$ is asymptotically normal with mean and variance*

$$\mu = N \sum_{\eta=0}^{M} \Lambda_\eta p_\eta \quad \textit{and} \quad \sigma^2 = N \sum_{\eta=0}^{M} \Lambda_\eta^2 p_\eta - \mu^2.$$

The proof is given in Appendix B in **VII**.

Some Definitions about Discrete Probability Distributions

This section gives some definitions and properties of discrete p.d.'s. Let $p = (p_0, \ldots, p_M)$ and $q = (q_0, \ldots, q_M)$ be p.d.'s of random variables with sample space $\mathcal{X} = \{0, 1, \ldots, M\}$. The Kullback-Leibler distance between p and q is defined as follows:

Definition 3.5. The *relative entropy* or *Kullback-Leibler distance* between p and q is

$$D(p \| q) = \sum_{\eta \in \mathcal{X}} p_\eta \log \frac{p_\eta}{q_\eta},$$

with the conventions $0 \log 0/b = 0$, $b \neq 0$ and $b \log b/0 = \infty$.

Note that the Kullback-Leibler distance is not a norm. However, in many situations (see for example [14] and Chapter 7) it is descriptive to interpret it as a norm. A similar concept is the Chernoff-information between p and q:

Definition 3.6. The *Chernoff-information* between p and q is

$$D^*(p,q) = -\min_{0\le\lambda\le 1} \log\left(\sum_{\eta\in\mathcal{X}} (p_\eta)^\lambda (q_\eta)^{1-\lambda}\right).$$

We say that p is *is close to* q, if there exists $\varepsilon, 0 < \varepsilon < 1/2$, such that

$$|p_\eta - q_\eta| \le \varepsilon q_\eta, \text{ for all } \eta \in \mathcal{X}. \qquad (3.15)$$

This definition implies that all the components of p and q satisfy the following condition: If $p_\eta q_\eta = 0$, then $p_\eta = q_\eta = 0$. We show later that this condition is not restrictive to our analysis.

If p is close to q, their Kullback-Leibler distance can be approximated using Taylor series [1] such that

$$D(p\|q) = C(p,q)/2 + \mathcal{O}(\varepsilon^3),$$

where ε is the parameter in (3.15) and the capacity $C(p,q)$ of p and q is defined as follows:

Definition 3.7. The *capacity* between two p.d.'s p and q is

$$C(p,q) = \sum_{\eta\in\mathcal{X}} (p_\eta - q_\eta)^2 q_\eta^{-1}.$$

If q is the uniform distribution, then $C(p,q)$ will be denoted by $C(p)$ and called the capacity of p.

It follows that if p is close to q, then $C(p,q) < \varepsilon^2 < 1$, where ε is the parameter in the definition (3.15). In practical cryptanalysis it is possible to assume $\varepsilon < 0.01$. This assumption can be verified in practice for each cipher. The capacity given by Definition 3.7 can be considered as a generalisation of the capacity introduced in [6].

Similarly, if p is close to q the Chernoff-information between p and q can also be approximated using the capacity [2]:

$$D^*(p,q) \approx (8\ln 2)^{-1} C(p,q). \qquad (3.16)$$

The *divergence* of p and q is defined as

$$I^\lambda(p:q) = \frac{1}{\lambda(1+\lambda)} \sum_{\eta=0}^{M} p_\eta \left((p_\eta/q_\eta)^\lambda - 1\right), \qquad (3.17)$$

for real $\lambda \ne 0, -1$ and by continuity in λ when $\lambda = 0, -1$. For $\lambda = 1$ the divergence is

$$I^1(p:q) = \frac{1}{2} C(p,q). \qquad (3.18)$$

For $\lambda = 0$ it is the Kullback-Leibler distance between p and q [17]:

$$I^0(p:q) = D(p\|q). \qquad (3.19)$$

Walsh Transforms and Discrete Probability Distributions

Let $m \geq 1$ be an integer. Let p be the p.d. of an m-bit random variable $\mathbf{X}$ and let $a \in \mathbb{F}_2^m$. By the definition of correlation (3.8), we have $c(a \cdot \mathbf{X}) = \Pr(a \cdot \mathbf{X} = 0) - \Pr(a \cdot \mathbf{X} = 1)$. The first probability can be written as

$$\begin{aligned}\Pr(a \cdot \mathbf{X} = 0) &= \sum_{\eta \in \mathbb{F}_2^m} \Pr(a \cdot \mathbf{X} = 0 \mid \mathbf{X} = \eta) p_\eta \\ &= \sum_{\eta, a \cdot \eta = 0} p_\eta = \sum_{\eta, a \cdot \eta = 0} (-1)^{a \cdot \eta} p_\eta .\end{aligned}$$

Similarly $\Pr(a \cdot \mathbf{X} = 1) = \sum_{\eta, a \cdot \eta = 1} -(-1)^{a \cdot \eta} p_\eta$. Hence, by the definition of Walsh transform (3.3) we get that

$$c(a \cdot \mathbf{X}) = \sum_{\eta \in \mathbb{F}_2^m} (-1)^{a \cdot \eta} p_\eta = \hat{p}(a). \qquad (3.20)$$

The following lemma presented in **III** then follows by using the inverse Walsh-Hadamard transform :

Lemma 3.8. *Let* $\mathbf{X} \sim p$ *be a discrete* m*-bit random variable taking on values in* $\mathbb{F}_2^m$. *Then*

$$p_\eta = 2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot \eta} c(a \cdot \mathbf{X}), \text{ for all } \eta \in \mathbb{F}_2^m .$$

In other words, the lemma says that the p.d. p of random variable $\mathbf{X}$ taking on values in $\{0, 1, \ldots, 2^m - 1\}$ is determined by the correlations of the projections $a \cdot \mathbf{X}$, $a \in \mathbb{F}_2^m$, in the subspace $\mathbb{F}_2$. This result is also known as the Cramér-Wold theorem [15]. The c.d.f. of a random variable is uniquely determined by its Fourier-Stieltjes transforms known as the characteristic functions. For discrete random variables, the transformation is the Walsh-transform and the characteristic function at $a \in \mathbb{F}_2^m$ is the correlation $c(a \cdot \mathbf{X})$.

Applying Parseval's theorem (3.6) to Lemma 3.8 we have the following result about the capacity of a p.d.

Lemma 3.9. *Let* p *be a p.d. of an* m*-bit random variable* $\mathbf{X}$ *taking on values in* $\mathbb{F}_2^m$. *Then the capacity of* p *is*

$$C(p) = \sum_{a \in \mathbb{F}_2^m \setminus \{0\}} c(a \cdot \mathbf{X})^2 .$$

Let $\mathbf{X}$ and $\mathbf{Y}$ be discrete m-bit random variables with sample space $\mathbb{F}_2^m$. Let p and q be the p.d.'s of $\mathbf{X}$ and $\mathbf{Y}$, respectively. By (3.5) and (3.20) the zth component of the convolution of p and q is

$$(q * p)_z = \sum_{\eta \in \mathbb{F}_2^m} q_\eta p_{z \oplus \eta} = 2^{-m} \sum_{a \in \mathbb{F}_2^m} (-1)^{a \cdot z} c(a \cdot \mathbf{X}) c(a \cdot \mathbf{Y}). \qquad (3.21)$$

We have the following result that can be seen as a generalisation of the Piling Up lemma (*if*-part of Theorem 3.3):

Lemma 3.10. *Let $\mathbf{X}$ and $\mathbf{Y}$ be s.i. m-bit random variables with sample space $\mathbb{F}_2^m$ and p.d.'s p and q, respectively. Then the sum $\mathbf{X} \oplus \mathbf{Y}$ is distributed as $p * q$.*

Proof. Let s be the p.d. of $\mathbf{X} \oplus \mathbf{Y}$, such that $s_z = \Pr(\mathbf{X} \oplus \mathbf{Y} = z)$. Then

$$s_\eta = \sum_{\eta \in \mathbb{F}_2^m} \Pr(\mathbf{X} \oplus \mathbf{Y} = z \mid \mathbf{X} = \eta) p_\eta = \sum_{\eta \in \mathbb{F}_2^m} \Pr(\mathbf{Y} = z \oplus \eta \mid \mathbf{X} = \eta) p_\eta.$$

Since $\mathbf{X}$ and $\mathbf{Y}$ are s.i. the conditional probability $\Pr(\mathbf{Y} = z \oplus \eta \mid \mathbf{X} = \eta) = q_{z \oplus \eta}$ for all $\eta \in \mathbb{F}_2^m$. □

We prove in **II** that the capacity of the p.d. $p * q$ capacity satisfies the inequality

$$C(p * q) \leq C(p)C(q). \tag{3.22}$$

Compare this to the one-dimensional case where $m = 1$. By Piling Up lemma, the correlation of the sum of two independent random variables is the product of their correlation. However, for $m \geq 2$ we have only inequality for the capacities. One open question is whether there is a non-trivial inequality of the form $C(p * q) \geq AC(p)C(q)$ for some constant $A > 0$. In Chapter 8 we see why it would be advantageous to find a lower bound for the combined capacity.

3.3.5 Order Statistics

Let $\mathbf{X}_1, \ldots, \mathbf{X}_d$ be continuous random variables and arrange them in decreasing order such that the ordered values are $\mathbf{X}_{(1)}, \mathbf{X}_{(2)}, \ldots, \mathbf{X}_{(d)}$, where $\mathbf{X}_{(1)} \geq \mathbf{X}_{(2)} \geq \cdots \geq \mathbf{X}_{(d)}$. The vector $(\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(d)})$ is the *order statistics* of the sample and $\mathbf{X}_{(r)}$ is called the *rth order statistic*. In literature, the ordering is usually done in increasing order [23], [18], but in this thesis the opposite ordering is more convenient.

If $\mathbf{X}_1, \ldots, \mathbf{X}_d$ are i.i.d., the following asymptotic result holds for the rth order statistic [23], [18].

Theorem 3.11. *Let $\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(d)}$ be the order statistics of i.i.d. random variables $\mathbf{X}_1, \ldots, \mathbf{X}_d$. Let the c.d.f. and p.d.f. of each $\mathbf{X}_t$ be F and f, respectively. When d approaches infinity and r/d remains fixed, the rth order statistic is asymptotically normally distributed with mean and variance*

$$\mu = F^{-1}(1 - r/d) \quad \text{and} \quad \sigma^2 = \frac{(1 - r/d)r/d}{df(\mu)^2}.$$

If the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_d$ are not statistically independent, we do not have a general result similar to Theorem 3.11. However, for the maximum order statistic $\mathbf{X}_{(1)}$ of dependent random variables $\mathbf{X}_1, \ldots, \mathbf{X}_d$ the following holds:

Lemma 3.12. *Let the continuous random variables $\mathbf{X}_1, \ldots, \mathbf{X}_d$ be identically distributed with c.d.f. $F(x)$ and p.d.f. $f(x)$ such that*

1. *$F(x) < 1$ for all $x \in \mathbb{R}$*

2. *There exists $x_0 \in \mathbb{R}$ such that $F(x)$ is twice differentiable at least for all $x > x_0$*

3. $$\lim_{x\to\infty} \frac{d}{dx}\left(\frac{1-F(x)}{f(x)}\right) = 0.$$

Then the following holds uniformly for all $y \in \mathbb{R}$:

$$\lim_{d\to\infty} \Pr\left((\mathbf{X}_{(1)} - l_d)Nf(l_d) \leq y\right) = \Lambda_3(y),$$

where $\Lambda_3(y)$ is

$$\Lambda_3(y) = \exp\left(-e^{-y}\right),$$

and l_d is given by

$$F(l_d) = \frac{d-1}{d}.$$

The proof is given in [18] with the following corollary:

Corollary 3.13. *If $\mathbf{X}_1, \dots, \mathbf{X}_d$ are normed normally distributed with p.d.f. ϕ and c.d.f. Φ, then their maximum is asymptotically distributed as*

$$\lim_{d\to\infty} \Pr\left((2\log d)^{1/2}\left(\mathbf{X}_{(1)} - (2\log d)^{1/2}\right) \leq y\right) = \Lambda_3(y),\ y \in \mathbb{R}.$$

3.4 BOOLEAN FUNCTIONS

A function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is called a *Boolean function*. A linear Boolean function is a mapping $x \mapsto u \cdot x$. A function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ with $f = (f_1, \dots, f_m)$, where f_i are Boolean functions is called a *vector Boolean function* of dimension m. A linear Boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is represented by an $m \times n$ binary matrix U. Hence, $x \mapsto Ux$ is a linear mapping and U is called a (multidimensional) linear mask of x. The m rows of U are denoted by $u_1, \dots, u_m$, where each u_i is a linear mask.

The correlation between a Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ and zero is

$$c(f, 0) = 2^{-n}\left(\#\{\xi \in \mathbb{F}_2^n \mid f(\xi) = 0\} - \#\{\xi \in \mathbb{F}_2^n \mid f(\xi) \neq 0\}\right),$$

and it is also called the correlation of f. Similarly, the p.d. of a vector Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is defined to be the p.d. of the random variable $f(\mathbf{X})$, where $\mathbf{X} \sim \theta$. Hence, we can identify any m-dimensional Boolean function f with a random variable $\mathbf{Y} = f(\mathbf{X})$, $\mathbf{X} \sim \theta$. We say that $\mathbf{Y}$ is associated with the Boolean function f. Therefore, we can define many concepts and derive many results about m-dimensional vector Boolean functions using discrete p.d.'s. We give some results below.

We have the following natural definition of statistical independence of Boolean functions: We say that the Boolean functions $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ and $g : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ are statistically independent, if the associated binary random variables are s.i. Hence, two Boolean functions are s.i. if they have no common input bits. Similarly, m Boolean functions $f_1, \dots, f_m : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ are s.i. if the associated random variables are s.i. Moreover, the test provided by the converse of Piling Up lemma 3.3 can be applied to Boolean functions to determine if they are s.i.

We define the capacity of a vector Boolean function f to be the capacity of its p.d. and we denote $C(f)$. Lemma 3.8 and Lemma 3.9 also hold for the correlations $c(a \cdot f, 0)$, $a \in \mathbb{F}_2^m$ and the p.d. of f.

The Walsh-transform $\hat{f}$ of a Boolean function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is defined as follows: We replace $\varphi(\eta)$ by $(-1)^{f(\eta)}$ in (3.3). Then $\hat{\varphi}(a)$ is replaced by $\hat{f}(a)$ and

$$\hat{f}(a) = \sum_{\eta \in \mathbb{F}_2^n} (-1)^{a \cdot \eta \oplus f(\eta)}.$$

The set $\{\hat{f}(a) \mid a \in \mathbb{F}_2^n\}$ is called the Walsh spectrum of f. It follows that the correlation and the Walsh transform of a Boolean function f are related by $c(f, 0) = 2^{-n} \hat{f}(0)$.

4 SYMMETRIC CRYPTOGRAPHY

In this section we present the two basic primitives used in symmetric cryptography: the block ciphers and the stream ciphers. We study block ciphers in Section 4.1 and stream ciphers in Section 4.2.

4.1 BLOCK CIPHERS

Some examples of block ciphers are the Data Encryption Standard (DES, FIPS 46-3), the Advanced Encryption Standard (AES, FIPS PUB 197) and Serpent [5]. DES was adopted as the encryption standard for US government in 1976. It was used worldwide for two decades. However, the development in technology and cryptology made it necessary to find a new and more secure block cipher standard, called AES. There was an open competition to choose the algorithm for AES. Among the candidates were Rijndael, which won the competition, and Serpent, which is the testbed in most of the practical experiments of this thesis.

A block cipher cuts the plaintext messages into strings or blocks of n bits $x_1, \ldots, x_N \in \mathbb{F}_2^n$. It then uses a key $K \in \mathcal{K}$ for encrypting one block at a time to obtain an n-bit ciphertext. For example, for DES $n = 64$ and for AES and Serpent $n = 128$.

In an *iterated block cipher* the same non-linear function $f(\cdot, k) : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$, dependent on the parameter $k \in \mathbb{F}_2^l$, is applied repeatedly. Each mapping with f is called a round and f is called the round function. Let $R \geq 1$ be the number of rounds and let K be the encryption key. Then the encryption function E_K is given by

$$E_K(x) = f(\ldots f(f(x, k_1), k_2), \ldots k_R).$$

For each $i = 1, \ldots, R$, the parameter k_i is called the ith round key and it is obtained from the original secret key K by using a key-schedule algorithm. See also Fig. 4.1. The last round key k_R is called the *outer key* and the keys $k_1, \ldots, k_{R-1}$ are called *inner keys* [6]. Iterated block ciphers are efficient and small to implement, since the same round function f is used in each round. If f is properly chosen, the security of the cipher increases with the number of rounds R.

DES, AES and Serpent are all iterated block ciphers. The initial key K of DES has only 56 bits whereas for AES and Serpent the key has 128, 192 or 256 bits. Each round key has $l = 48$ bits for DES and $l = n = 128$ bits for AES and Serpent. The number of rounds R is 16 for DES, 10, 12 or 14 for AES and 32 for Serpent.

Shannon stated that a good cipher should have the *diffusion* and *confusion* properties [41]. Diffusion ensures that the statistical redundancy or non-randomness should be spread out over the whole ciphertext. Hence, a large amount of ciphertext is required for finding non-random behaviour for the cipher. Diffusion can be obtained by permuting the symbols of each block.

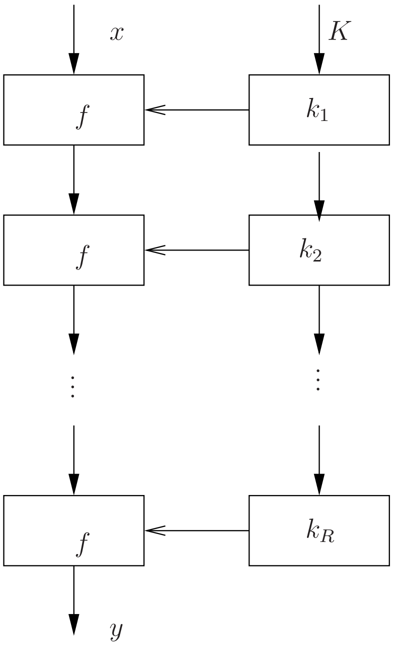


Figure 4.1: An R-round iterated block cipher

Confusion makes the relationship between the key and ciphertext as complex as possible such that the attacker should not be able to determine the key from a given ciphertext. Confusion is often achieved by using S-boxes that are non-linear multidimensional Boolean functions.

Confusion and diffusion can then be obtained by combining consecutive permutations and substitutions. Such a cipher is called a *substitution-permutation network* (SPN). An iterated cipher is an SPN, if the round function $f(\cdot, k)$ is defined by

$$f(x, k) = \pi_P(\pi_S(x)) \oplus k, \text{ for all } x \in \mathbb{F}_2^n,$$

where $\pi_S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ is a non-linear substitution function consisting of one or more S-boxes and $\pi_P : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ is a permutation of the bits.

The permutation in AES is not a simple bit permutation but a slightly more general mapping. It uses the S-box defined by $Ax^{-1} + b$ calculated in the finite field $\mathbb{F}_2^8$. Here A is a constant matrix and b is a constant vector defined in $\mathbb{F}_2^8$. The 128 bit block is divided to 16 blocks of 8 bits and then each smaller block is mapped using the S-box. Since the S-box is the only non-linear part in the cipher, it is crucial to the security of AES.

As an example, we give a short description of the block cipher Serpent we use in our experiments.

Example 4.1 (Serpent). Serpent has block size 128 and it supports key lengths 128, 192 and 256 bits. It consists of 32 similar rounds. We use the notation of [5]. Each intermediate value of round i is denoted by $\hat{B}_i$ (a 128-bit value). Each $\hat{B}_i$ is treated as four 32-bit words X_0, X_1, X_2, X_3 where the jth bit of X_i is the $4 * i + j$th bit of $\hat{B}_i$. Serpent has a set of eight 4-bit to 4-bit S-boxes $S_0, \ldots, S_7$ and a 128-bit to 128-bit linear transformation LT. Each round function R_i uses a single S-box 32 times in parallel.

Let the plaintext be x and ciphertext y. Then $\hat{B}_0 = \mathrm{IP}(x), \hat{B}_{i+1} = R_i(\hat{B}_i)$ for $i = 0, \ldots, 31$ and $y = \mathrm{IP}^{-1}(\hat{B}_{32})$. Here IP is a permutation and the round function is

$$\begin{aligned} R_i(X) &= \mathrm{LT}(S_i(X \oplus \hat{K}_i)), \quad i = 0, \ldots, 30 \\ R_i(X) &= S_i(X \oplus \hat{K}_i) \oplus \hat{K}_{32}, \quad i = 31, \end{aligned}$$

A more detailed description of can be found in [5].

Block ciphers can be analysed with Matsui's linear cryptanalysis by finding a biased linear relation between the plaintext and ciphertext words. We study the attacks in Chapters 6 and 7.

4.2 STREAM CIPHERS

Stream ciphers are fast and suit well light-weight applications such as mobile telephones. Hence, the eSTREAM-project was launched in 2004 in hope to finding some good stream cipher candidates that could be regarded as a standard. The final portfolio contains for example SOSEMANUK [3], Rabbit [7] and Grain [24]. Another commonly used stream cipher is SNOW [20].

In a stream cipher each plaintext message symbol $x_t \in \mathbb{F}_2^n$, where $t = 1, \ldots, N$, is encrypted with a different n'-bit key z_t such that the ciphertext is $y_t = E_{z_t}(x_t)$. The sequence $z_1, \ldots, z_N$, is called the *keystream*. Originally, the message symbols had just one bit. Nowadays, n is usually the word-size of the computer system, for example 32 or 64.

4.2.1 Additive Stream Ciphers and Key Stream Generators

In additive stream ciphers the encryption function E_z is simply the XOR: for each $t = 1, \ldots, N$,

$$y_t = x_t \oplus z_t,$$

and $n' = n$. If the words $z_1, \ldots, z_N$, correspond to i.i.d. and uniformly distributed random variables, the additive cipher is called the one-time pad, which Shannon showed to provide *perfect secrecy* [41]. However, if we want to cipher N plaintexts $x_1, \ldots, x_N$, we must also create a keystream $z_1, \ldots, z_N$, of Nn bits that are i.i.d. and uniformly distributed. In ordinary applications N is so large that this is not feasible. Instead, we use a *keystream generator* (k.s.g.) to extend a smaller secret random number $K \in \mathbb{F}_2^l$ to the keystream $z_1 \ldots z_N$. Typically, l is 128 or 256.

The security of the stream cipher lies then on the properties of the k.s.g. It should be difficult to determine the original secret key from a given key sequence and the key sequence should be indistinguishable from a random sequence. The output sequence $z_0, z_1, \ldots$ should also have a long period. The sequence $z_t, t = 0, 1, \ldots$ has period p if p is the smallest non-zero number such that for all $t \geq 0$,

$$z_t = z_{t+p}.$$

A common way to produce sequences of long periods is to use linear feedback shift registers. They are studied in the next section.

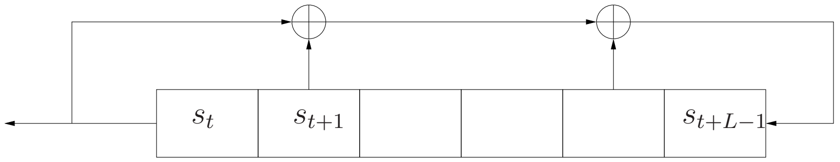


Figure 4.2: An LFSR with 3 taps

4.2.2 Linear Feedback Shift Registers

Figure 4.2 shows an example of a linear feedback shift register (LFSR). Its output at time $t = 0, 1, 2, \ldots$ is $s_t \in \mathbb{F}_2^n$, where n is called the block size of the LFSR. The vector $(s_t, s_{t+1}, \ldots, s_{t+L-1})$ is the state of the LFSR at time t and L is called the length of the LFSR. The LFSR is updated using the linear recursion

$$s_{t+L} = \bigoplus_{i=0}^{L-1} b_i s_{t+i}, \text{ for all } t = 0, 1, \ldots, \tag{4.1}$$

where the multipliers $b_0, \ldots, b_{L-1} \in \mathbb{F}_2^n$ and $b_0 \neq 0$. The number of non-zero coefficients $b_0, \ldots, b_{L-1}$ is called the number of taps.

Denote the whole internal state of the LFSR at time $t \geq 0$ by $Y_t = (s_t, \ldots, s_{t+L-1})$. The update recursion (4.1) can then be written as $Y_{t+1} = AY_t$, for all $t \geq 0$. Here A is a matrix of the form

$$A = \begin{bmatrix} \mathbf{0} & I \\ b_0 & b_1 \ldots b_{L-1} \end{bmatrix}, \tag{4.2}$$

where $\mathbf{0}$ is an $(L-1) \times 1$ zero-vector and I is an $(L-1) \times (L-1)$ identity matrix. Then, for given initial state Y_0, any state Y_t is given by $Y_t = AY_{t-1} = A^t Y_0$.

4.2.3 Some K.S.G. Constructions for Additive Synchronous Stream Ciphers

LFSRs have several nice properties as they are fast to implement and produce sequences with long periods. However, since they are linear, they can be analysed algebraicly. The solution is to add some non-linearity to the k.s.g. This section studies some such constructions.

Non-Linear Combiner Generator

Figure 4.3 shows a classical k.s.g. construction called a non-linear combination generator. It consists of m LFSRs, whose outputs are used as an input to the non-linear combination function f. The output of f is the key sequence z_t, $t \geq 0$. The non-linear part may also have an internal state. It is called memory or carry. The E0 cipher used in Bluetooth devices is a combination generator with four LFSRs and four bits of memory. Siegenthaler showed how this type of k.s.g. can be attacked using a correlation attack [42].

Non-Linear Filter Generator

A different approach is to use one LFSR and take several of its state blocks, denoted by S_t, as an input to a non-linear vector Boolean function f called

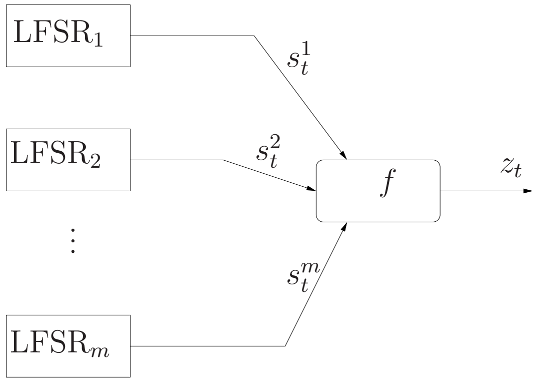


Figure 4.3: Non-linear combination generator with m LFSRs and non-linear output function f. The output of the ith LFSR at time t is s_t^i.

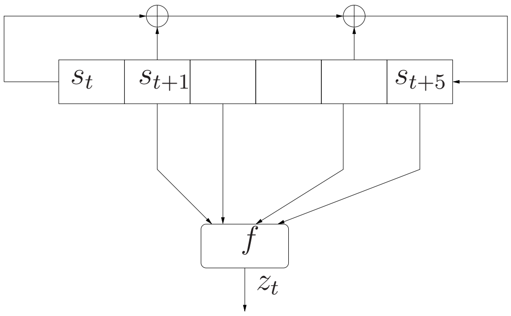


Figure 4.4: A non-linear filter generator with filtering function f.

a *filtering function*. This construction is called a non-linear filter generator and it is depicted in Figure 4.4. The filtering function can have an internal state called memory. The output z_t, $t \geq 0$ of the k.s.g. is then

$$z_t = f(S_t), \text{ for all } t \geq 0. \tag{4.3}$$

The SNOW2.0 -cipher is an example of a non-linear filter generator with memory. The SOSEMANUK-stream cipher we analyse in **VI** inherits the design structure of SNOW2.0.

Example 4.2 (SNOW2.0)**.** The linear part is an LFSR with $L = 16$ words of $n = 32$ bits. The LFSR is updated independently of the non-linear part using the update function

$$s_{t+16} = \alpha^{-1} s_{t+11} \oplus s_{t+2} \oplus \alpha s_t,$$

where $\alpha \in \mathbb{F}_2^{32}$ is a root of a primitive polynomial of degree 4 in $\mathbb{F}_2^8$. The non-linear part is called a finite state machine consisting of two registers $R1_t$ and $R2_t$ updated by

$$R1_{t+1} = s_{t+5} \boxplus R2_t \quad \text{and} \quad R2_{t+1} = \pi_S(R1_t), \text{ for } t \geq 0, \tag{4.4}$$

where $\boxplus$ is addition modulo 32 and π_S is a non-linear permutation in $\mathbb{F}_2^{32}$ based on the AES S-box. The registers $R1_t$ and $R2_t$ are the memory of the cipher.

The output is generated using the registers $R1_t$ and $R2_t$ and LFSR words s_t and s_{t+15} as follows:

$$z_t = s_t \oplus (s_{t+15} \boxplus R1_t) \oplus R2_t, \ t \geq 1.$$

Non-linear filter generators can be attacked using linear cryptanalysis. The attacker must find a biased linear relation between the certain keystream words. We study the attack more closely in Chapters 6 and 7.

5 STATISTICAL INFERENCE

Statistical inference is used in finding information about a population. In parametric hypothesis testing problems (HTP) there are two or more distinct *hypotheses* about a population parameter, one of which is true. The task is to distinguish the right hypotheses from the other hypotheses, i.e, to determine the right value for the parameter. In Section 5.1 we study some parametric hypothesis testing problems.

In a goodness-of-fit problem we have to decide whether a given random sample is drawn from a given p.d. or not. This problem and its solution under some special conditions is studied in Section 5.2.

In a general d-sample problem we have d random samples. The task is to determine whether all samples follow the same population or not. However, it is also possible to find more information about the underlying populations, provided we have some additional information. Specifically, if exactly one population is different from the other populations, order statistics can be used for determining the distinct population. This problem and its solution is studied in Section 5.3.

5.1 PARAMETRIC HYPOTHESIS TESTING

In parametric hypothesis testing problems there are two possible approaches: the *classical* or Neyman-Pearson tests and the *Bayesian* tests. In a classical HTP, one hypothesis is a *null hypothesis* that is either accepted or rejected. If it is rejected, an *alternative hypothesis* is concluded.

In Bayesian statistic, all the hypotheses are equal: One hypothesis is accepted and the others are rejected. Each hypothesis has an *a priori* probability to be true. There has been some dispute which test type should be preferred [10]. In statistical cryptanalysis there is no practical difference between the two approaches when solving a binary HTP. On the other hand, when more than two hypotheses are to be tested, Bayesian statistic is the natural choice.

A distinguisher decides, which hypothesis should be accepted with the given data. The next section gives an accurate description for a parametric HTP and studies distinguishers in general and the cost related to a distinguisher.

5.1.1 Distinguisher and Cost

Let the population parameter and parameter space be ω and Ω, respectively. Let Ω_i, $i = 1, \ldots, d$, be $d \geq 2$ distinct, non-empty subsets of Ω. In a d-ary hypotheses testing problem each hypotheses H_i, $i = 1, \ldots, d$ states that $\omega \in \Omega_i$. If the hypothesis specifies the population distribution completely, i.e., if Ω_i consists of just one point in the parameter space, the hypothesis is called *simple*. Otherwise, it is called *composite*.

The random sample drawn from the population is $\mathbf{X}_1, \ldots, \mathbf{X}_N$, its realisation is $x_1, \ldots, x_N$, and joint sample space is $\mathcal{X}^N$. A *distinguisher* δ is a function that using $(x_1, \ldots, x_N) \in \mathcal{X}^N$ outputs which hypothesis is true:

$$\delta(x_1, \ldots, x_N) = i, \text{ if } H_i \text{ is accepted}, i = 1, \ldots, d.$$

The distinguisher is defined using a suitable *test statistic* $g(\mathbf{X}_1, \ldots, \mathbf{X}_N; \omega)$ that depends on the parameter ω. Using the statistic we define function $g'(\omega; \mathbf{x}_1, \ldots, \mathbf{x}_N) = g(x_1, \ldots, x_N; \omega)$, where ω is the variable and the observations $x_1, \ldots, x_N$ are the parameter. Given the data $x_1, \ldots, x_N$, the distinguisher outputs i, if $g'(\omega; x_1, \ldots, x_N)$ is maximised in the set Ω_i.

The choice of a proper statistic is the main problem in statistical hypothesis testing. It should be both efficient to evaluate and accurate when making the decision. There is no unique measure for the accuracy of a distinguisher. One possibility is to use the concept of *lost* [45] or *cost* [32]: If the distinguisher outputs j when hypothesis H_i is true, the cost (or lost) is C_{ij}. The total cost is then

$$C = \sum_{i,j \in \mathcal{X}} C_{ij} \Pr(H_i) \Pr(\delta(\mathbf{X}_1, \ldots, \mathbf{X}_N) = j \mid H_i), \tag{5.1}$$

where $\Pr(H_i)$, $i = 1, \ldots, d$, are the *a priori* probabilities of the hypotheses [45] [32]. A distinguisher should have as small cost as possible, for given amount N of observed data. An *optimal distinguisher* minimises the cost. All HTP's do not have an optimal distinguisher. However, we consider now a simple d-ary HTP, for which an optimal solution exists. Since the hypotheses are simple, each hypothesis H_i states that the population has p.d.f. $f(x; \omega_i)$. The joint p.d.f. $f_{\mathbf{X}_1, \ldots, \mathbf{X}_N}$ of the sample is then given by (3.7).

Consider the case where choosing the right hypothesis has zero cost, that is, $C_{ii} = 0$ for all $i = 1, \ldots, d$ and the cost of choosing a wrong hypothesis is always one, such that $C_{ij} = 1$, for all $i \neq j$. To minimise the cost (5.1), we choose the hypothesis H_i that has the maximum *a posteriori* probability $\Pr(H_i \mid x_1, \ldots, x_N)$ [32]. Using Bayes's formula, we have

$$\Pr(H_i \mid x_1, \ldots, x_N) = \Pr(H_i) f_{\mathbf{X}_1, \ldots, \mathbf{X}_N}(x_1, \ldots, x_N; \omega_i) / \Pr(x_1, \ldots, x_N).$$

Hence, the optimal distinguisher outputs i if i gives the maximum of

$$\Pr(H_i) \mathcal{L}(i; x_1, \ldots, x_N),$$

where

$$\begin{aligned} \mathcal{L}(i; x_1, \ldots, x_N) &= \mathcal{L}(\omega_i; x_1, \ldots, x_N) \\ &= f_{\mathbf{X}_1, \ldots, \mathbf{X}_N}(x_1, \ldots, x_N; \omega_i) \\ &= \prod_{t=1}^{N} f(x_t; \omega_i) \end{aligned} \tag{5.2}$$

is the *likelihood function* of the p.d.f. $f(x; \omega_i)$. Hence, the joint p.d.f. $\prod_{t=1}^{N} f(x_t; \omega_i)$ weighted with the *a priori* probability $\Pr(H_i)$ is the optimal test statistic.

The logarithm of the likelihood function, called the *log-likelihood function*, gives an equivalent distinguisher. It is often more convenient to use in practice than the likelihood function.

The cost of the optimal distinguisher is equal to the *total error probability* given by

$$P_e = \sum_{i=1}^{d} \Pr(H_i) \sum_{j \neq i}^{d} \Pr(\delta(\mathbf{X}_1, \ldots, \mathbf{X}_N) = j \mid H_i). \tag{5.3}$$

In the next sections the simple binary and *d*-ary hypothesis testing problems are studied in more detail. We show that the optimal distinguisher for those tests is based on a test statistic called the log-likelihood ratio (LLR).

5.1.2 Simple Binary Hypothesis Testing Problem and LLR

A simple binary HTPconsists of hypotheses

$$\begin{aligned} H_0 &: \omega = \omega_0 \\ H_1 &: \omega = \omega_1 \neq \omega_0. \end{aligned}$$

We denote the p.d.f.'s corresponding to H_0 and H_1 by $f(x_1, \ldots, x_N; \omega_0)$ and $f(x_1, \ldots, x_N; \omega_1)$, respectively. In classical statistics the null hypothesis H_0 is either accepted or rejected. If H_0 is accepted, then H_1 is concurred. In the Bayesian approach one hypothesis is accepted and the other is rejected - neither hypothesis has a "special" status.

In a binary HTP there are two error probabilities $\alpha = \Pr(\delta(\mathbf{X}_1, \ldots, \mathbf{X}_N) = 1 \mid H_0)$ and $\beta = \Pr(\delta(\mathbf{X}_1, \ldots, \mathbf{X}_N) = 0 \mid H_1)$, called the Type I and Type II Error of test, respectively. In classical statistics α is also referred to as the *level* or size of the test and $1 - \beta$ interpreted as a function of the parameter ω is called the *power* function of the test. The total error used in Bayesian statistics is $P_e = \Pr(H_0)\alpha + \Pr(H_1)\beta$.

Consider the optimal distinguisher for the binary HTP that outputs 0 if $\Pr(H_0)\mathcal{L}(x_1, \ldots, x_N; 0) > \Pr(H_1)\mathcal{L}(x_1, \ldots, x_N; 1)$. An equivalent distinguisher is given by the *likelihood ratio,* defined by

$$\mathrm{LR}(x_1, \ldots, x_N) = \frac{\mathcal{L}(x_1, \ldots, x_N; 0)}{\mathcal{L}(x_1, \ldots, x_N; 1)} = \prod_{t=1}^{N} \frac{f(x_t; \omega_0)}{f(x_t; \omega_1)}. \tag{5.4}$$

According to the Neyman-Pearson lemma in classical statistics [14], the optimal distinguisher outputs H_0, if $\mathrm{LR}(x_1, \ldots, x_N) \geq \tau$, where τ is a threshold that depends on α and β. Taking logarithm of $\mathrm{LR}(x_1, \ldots, x_N)$ gives another equivalent test statistic, the *log-likelihood ratio* (LLR):

$$\mathrm{LLR}(x_1, \ldots, x_N) = \sum_{t=1}^{N} \log \frac{f(x_t; \omega_0)}{f(x_t; \omega_1)}.$$

The optimal distinguisher outputs 0 (1) if $\mathrm{LLR}(x_1, \ldots, x_N) \geq \tau$ ($< \tau$) for a threshold τ. If $\alpha = \beta$ then $\tau \approx 0$ and we set $\tau = 0$ for the inference.

Assume now that the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$ are discrete such that $\mathcal{X} = \{0, 1, \ldots, M\}$. Denote the p.d.'s determined by p.d.f.'s $f(x_1, \ldots, x_N; \omega_0)$ and $f(x_1, \ldots, x_N; \omega_1)$ by p^0 and p^1, respectively. For distinguishing between H_0 and H_1, or equivalently between p^0 and p^1, it is not necessary to know the order in which the sample values appear in the sequence $x_1, \ldots, x_N$. Rather, it is enough to consider their relative frequencies given by the random vector $\mathbf{Q}$ defined in (3.12). Then H_0 states that $\mathbf{Q} \sim \text{multi}(N, p^0)$ and H_1 states that $\mathbf{Q} \sim \text{multi}(N, p^1)$.

Let q be the empirical p.d. defined in (3.13). The LLR can be written as

$$\text{LLR}(x_1, \ldots, x_N) = \text{LLR}(q) = \sum_{\eta \in \mathcal{X}} N q_\eta \log \frac{p^0_\eta}{p^1_\eta}. \tag{5.5}$$

We use notation $\text{LLR}(q; p^0, p^1)$ if it is necessary to emphasise the dependence of the LLR on p^0 and p^1.

We consider now the special cases where p^0 or p^1 or both have zero components. This problem was also considered in [2]. If for some $\eta \in \mathcal{X}$ the components $p^0_\eta = p^1_\eta = 0$, we simply omit the ηth component from the LLR. Assume next that $p^0_\eta = 0$ and $p^1_\eta \neq 0$. If $q_\eta = 0$, we define similarly as for the Kullback-Leibler distance that $0 \log(0/p^1_\eta) = 0$. On the other hand, if $q_\eta \neq 0$, the data cannot be drawn from p^0 and hence, we define $q_\eta \log(0/p^1_\eta) = -\infty$ to ensure that H_1 is accepted. Similar deduction goes for $p^0_\eta \neq 0$ and $p^1_\eta = 0$. Hence, we can restrict to situations where p^0 and p^1 have only positive components. Then the capacity and the condition (3.15) for closeness of two distributions are well-defined.

Our main goal is to determine the data complexity of the optimal distinguisher that is, the amount of data needed to have a given level and power of test (with classical approach) or total error probability (with Bayesian approach). The data complexities are the same for both approaches in linear cryptanalysis. Therefore, we only give the proof in the classical case.

In order to find a formula for the data complexity, we have to determine the p.d. of the test statistic. By Lemma 3.4, the LLR-statistic has the following property:

Proposition 5.1. *Let p^0 and $p^1 \neq p^0$ be two distinct p.d.'s. Let $\mathbf{Q}$ be the vector of relative frequencies of a random sample that is drawn from p^0 or p^1. The LLR test statistic $\text{LLR}(\mathbf{Q}; p^0, p^1)$ given by (5.5) is asymptotically normal with mean and variance $N\mu_i$ and $N\sigma_i^2$, respectively, if the data is drawn from p^i, for $i = 0, 1$. The means and variances are given by*

$$\mu_0 = D(p^0 \| p^1) \quad \mu_1 = -D(p^1 \| p^0)$$
$$\sigma_0^2 = \sum_{\eta=0}^{M} p^0_\eta \log^2 \frac{p^0_\eta}{p^1_\eta} - \mu_0^2 \quad \sigma_1^2 = \sum_{\eta=0}^{M} p^1_\eta \log^2 \frac{p^0_\eta}{p^1_\eta} - \mu_1^2.$$

Moreover, if p^0 is close to p^1 in the sense of definition (3.15), the following estimates hold

$$\mu_0 \approx -\mu_1 \approx \frac{1}{2} C(p^0, p^1) \quad \sigma_0^2 \approx \sigma_1^2 \approx C(p^0, p^1). \tag{5.6}$$

Baignères, et al., presented the same result in [1].

The data complexity can now be determined using Proposition 5.1. The idea is to fix the level and the power of the test and then determine the threshold τ and data complexity N from the resulting equations. In [1] the calculations where simplified by assuming $\alpha = \beta$ and, therefore, we can set $\tau = 0$. The result is also known as the Chernoff-Stein lemma [14]. We give it as the following theorem:

Theorem 5.2. *Assume that p^0 is close to p^1. Let the level and the power of the test be α and $1-\beta$, respectively. The data complexity of distinguishing p^0 from p^1 is proportional to*

$$N = \frac{(z_\alpha - z_\beta)^2}{C(p^0, p^1)},$$

where $z_\alpha = \Phi^{-1}(\alpha)$ and $z_\beta = \Phi^{-1}(1-\beta)$.

Proof. Using its definition, the Type I Error is

$$\begin{aligned}\alpha &= \Pr(\delta(\mathbf{Q}) = 1 \mid H_0)\\ &= \Pr\left(\mathrm{LLR}(\mathbf{Q}; p^0, p^1) < \tau \mid \mathrm{LLR}(\mathbf{Q}; p^0, p^1) \sim \mathcal{N}(\mu_0, \sigma_0^2)\right)\\ &= \Phi\left(\sqrt{N}\frac{\tau - \mu_0}{\sigma_0}\right).\end{aligned}$$

Hence, N and τ must satisfy

$$\sqrt{N}\frac{\tau - \mu_0}{\sigma_0} = z_\alpha. \tag{5.7}$$

Similarly, for Type II Error, we get the condition

$$\sqrt{N}\frac{\tau - \mu_1}{\sigma_1} = z_\beta. \tag{5.8}$$

Using approximations (5.6) we have the result. □

We have the same result in Bayesian theory: Chernoff's theorem states that $P_e = \mathcal{O}(2^{-ND^*})$, where D^* is the Chernoff information in Definition 3.6. If p^0 is close to p^1, approximation (3.16) gives that the data complexity is again inversely proportional to the capacity $C(p^0, p^1)$.

In the next section we generalise this theory of two discrete, simple hypotheses to a general d-ary HTP, where $d \geq 2$.

5.1.3 Simple d-ary Hypothesis Testing Problem

As we stated earlier, the Bayesian approach is natural for testing multiple hypotheses. For simplicity, we assume equal *a priori* probabilities. Moreover, we consider only the discrete setting with sample space $\mathcal{X} = \{0, 1, \ldots, M\}$, for some $M \geq 1$. Then each hypothesis H_i, $i = 1, \ldots, d$, states that the sample population is $p^i = (p^i_0, \ldots, p^i_M)$. We assume $p^i \neq p^j$, if $i \neq j$. Let $\mathbf{X}_1, \ldots, \mathbf{X}_N$ be the random sample and $x_1, \ldots, x_N$ be the corresponding observations. Similarly as for the case $d = 2$, we can state the hypotheses using the multinomially distributed $\mathbf{Q}$ defined in (3.12): Each hypothesis H_i, $i = 1, \ldots, d$, states that $\mathbf{Q} \sim \mathrm{multi}(N, p^i)$.

Consider the likelihood function $\mathcal{L}(i; x_1, \ldots, x_N) = \mathcal{L}(i; q)$, where q is the empirical p.d. given by (3.13). Given q, the function $\mathcal{L}(i; q)$ should reach its maximum for the right p.d. p^i. Using the formula (3.14) of the p.d. of the multinomial distribution, the likelihood function can be written as

$$\mathcal{L}(i;q) = \frac{N!}{\prod_{\eta\in\mathcal{X}}(q_\eta N)!}\prod_{\eta\in\mathcal{X}}(p^i_\eta)^{Nq_\eta}.$$

Assume now that each $p^i \neq \theta$. We can define an equivalent distinguisher using a new test statistic $l(i)$ that is obtained from the likelihood function by taking logarithm and omitting terms that do not depend on i:

$$l(i) = l(i;q) = N\sum_{\eta\in\mathcal{X}} q_\eta \log p^i_\eta - N\sum_{\eta\in\mathcal{X}} q_\eta \log M^{-1} = \mathrm{LLR}(q;p^i,\theta). \quad (5.9)$$

The distinguisher outputs i that maximises $l(i)$. Hence, the LLR-statistics gives the optimal distinguisher for a multiple HTP for $d > 2$, also. The LLR measures whether the data is drawn from p^i or the uniform distribution. High values imply that the empirical p.d. q is "closer" in Kullback-Leibler distance to p^i than θ.

The data complexity of LLR for solving a simple d-ary hypothesis testing problem with equal prior probabilities is given in the following lemma.

Lemma 5.3. *Assume equal prior probabilities in a simple d-ary HTP, and $p^i \neq \theta$ for all $i = 1,\ldots,d$. The optimal distinguisher minimising the error probability P_e in* (5.3) *is given by the LLR test statistic defined by the function* (5.9)*. Moreover, if the p.d.'s p^i are all close to each other and θ in the sense of definition* (3.15)*, the upper bound of the data complexity is proportional to*

$$N = \log(d(d-1)/P_e)(\min_{i\neq j} C(p^i,p^j))^{-1}.$$

Proof. The optimality follows from the previous calculations and it suffices to prove the formula for the data complexity. Fix P_e. Assume first two hypotheses H_i and H_j with p.d.'s p^i and p^j. The data complexity of successfully distinguishing between these distributions is by Chernoff's theorem proportional to

$$N_{ij} = \frac{\log P_{ij}^{-1}}{C(p^i,p^j)},$$

where $P_{ij} \leq P_e$ is the error probability. For $d \geq 3$ hypothesis the total error probability is

$$P_e = d^{-1}\sum_{i=1}^{d}\Pr(\delta(\mathbf{Q}) \neq i \mid H_i) = d^{-1}\sum_{i=1}^{d}\sum_{j\neq i} P_{ij}.$$

Each probability $P_{ij} = \Pr(\delta(\mathbf{Q}) = j \mid H_i)$ corresponds to a binary HTP. Fix the largest of the probabilities P_{ij}, $i \neq j$, to be $P_e/(d(d-1))$. Then the total error probability is $\leq P_e$ and the total data complexity is given by

$$N = \max_{i\neq j} N_{ij} = \log(d(d-1)/P_e)(\min_{i\neq j} C(p^i,p^j))^{-1}.$$

□

The proof of the lemma is independent of the given statistic. Only the data complexity of the binary HTP is counted. Hence, we have the following result:

Lemma 5.4. *If for any test statistic, the data complexity for distinguishing between p^0 and p^1 is inversely proportional to $C(p^0, p^1)$, then the data complexity for solving the d-ary HTP is given by Lemma 5.3.*

The LLR belongs to a wider class of linear test statistics, where the statistic is of the form

$$\sum_{\eta \in \mathcal{X}} \Lambda^i_\eta \mathbf{Q}_\eta, \tag{5.10}$$

where Λ^i_η, $\eta \in \mathcal{X}$, are some properly chosen non-zero, real coefficients that depend on the parameter i. The distinguisher outputs i if i maximises (or, depending on the choice of the coefficients, minimises) $g(i;q) = \sum_{\eta\in\mathcal{X}} \Lambda^i_\eta q_\eta$ for given q.

Each linear test statistic has the distribution given by Lemma 3.4. The strength of the statistic depends on the coefficients Λ^i_η. We know that the optimal choice is $\Lambda^i_\eta = \log p^i_\eta$, for $\eta \in \mathcal{X}$ and $i = 1, \ldots, d$. However, we show in **VII** that another possibility $\Lambda^i_\eta = p^i_\eta$ for $\eta \in \mathcal{X}$ and $i = 1, \ldots, d$ has practically the same data complexity for two hypotheses, if the p.d's are close to each other. By Lemma 5.4, they are practically equivalent with the same data complexities for multiple HTP also.

5.1.4 Binary Hypothesis Testing Problem with Alternative of Multiple P.D.'s

Consider the following HTP: The simple null hypothesis H_0 states that the data is drawn from a discrete p.d. p. The composite alternative hypothesis H_1 states that the data is drawn according to one p.d. in a set $\mathcal{P}$ of discrete p.d.'s and $p \notin \mathcal{P}$. Let $p_{\min}$ be the element p' of $\mathcal{P}$ that is closest to p in Kullback-Leibler distance $D(p' \| p)$.

Baignères and Vaudenay argued that the problem of distinguishing H_0 from H_1 is equivalent to distinguishing p from $p_{\min}$ [2] and the problem reduces to a simple binary HTP that can be solved with LLR, with data complexity proportional to

$$N = C(p, p_{\min})^{-1}.$$

This turns out to be an optimistic estimate of the true data complexity. In reality, distinguishing p from the whole set $\mathcal{P}$ requires running the test for all $p' \in \mathcal{P}$ separately. Hence, applying Lemma 5.3, the data complexity of distinguishing H_0 from H_1 using LLR is given by

$$N = \log |\mathcal{P}| C(p, p_{\min})^{-1}.$$

5.2 GOODNESS-OF-FIT PROBLEMS

5.2.1 Problem Statement

Consider a situation where the problem is to determine whether the population sample is a given p.d. $\mathcal{D}$ or not. The hypotheses are then

$$\begin{aligned} H_0 &: \mathbf{X}_t \sim \mathcal{D}, \text{ for all } t = 1, \ldots, N \\ H_1 &: \mathbf{X}_t \nsim \mathcal{D}, \text{ for all } t = 1, \ldots, N. \end{aligned}$$

This is called a goodness-of-fit problem belonging to the class of non-parametric or distribution free statistics, see for example [23] and [28]. Since H_1 is a composite hypothesis, there is no distinguisher that is optimal for all goodness-of-fit problems. Some test statistics suit better some problems than others.

Assume now that $\mathcal{X} = \{0, 1, \ldots, M\}$ and that the p.d. corresponding to null hypothesis is $p = (p_0, \ldots, p_M)$. Let $\mathbf{Q}$ defined by (3.12) be the random vector of relative frequencies in the random sample. The null hypothesis H_0 of goodness-of-fit problem states that $\mathbf{Q} \sim \text{multi}(N, p)$. The alternative hypothesis H_1 states that $\mathbf{Q} \nsim \text{multi}(N, p)$, in other words, $\mathbf{Q} \sim \text{multi}(N, p')$, where $p' \neq p$ is an unknown p.d. Let $\mathcal{P}$ denote the set of p.d.'s corresponding to the alternative hypothesis. In the general case, $\mathcal{P} = \{p' = (p'_0, \ldots, p'_M) : p' \neq p\}$.

Cressie and Read [17] considered a class of test statistics R^λ defined by the divergence (3.17) between the empirical p.d. q and the p.d. p corresponding to H_0:

$$R^\lambda(q : p) = 2NI^\lambda(q : p). \tag{5.11}$$

The divergence $I^\lambda(q : p)$ between q and p increases if the data q does not fit with the p.d. p, that is, if the sample is drawn from $p' \in \mathcal{P}$. Hence, the distinguisher outputs H_0 if $R^\lambda(q : p) \leq \tau$, where τ is a properly chosen threshold, and otherwise it outputs H_1.

We may assume that $p_\eta > 0$ for all $\eta \in \mathcal{X}$. As we noted in Section 5.1.2, this assumption is not restrictive to the analysis: if for some $\eta \in \mathcal{X}$ we have $p_\eta = 0$, then if $q_\eta \neq 0$, we immediately know that the null hypothesis cannot be true. On the other hand, if $p_\eta = q_\eta = 0$, we set $q_\eta \left((q_\eta/p_\eta)^\lambda - 1\right) = 0$ in the definition (3.17) of divergence. Hence, the zero components of p can be omitted in this theory.

If H_0 is true, then $R^\lambda(\mathbf{Q} : p)$ is χ^2_M-distributed with M-degrees of freedom [17]. For a given level α of the test, we have

$$\alpha = \Pr(H_1 \mid H_0) = \Pr(R^\lambda(\mathbf{Q} : p) > \tau \mid H_0) = 1 - \chi^2_M(\tau), \tag{5.12}$$

and we can determine the threshold τ for given α.

Since H_1 is composite, we have no representation for the power of the test in the general case and the data complexity remains undetermined. Therefore, we cannot compare the efficiency of the different test statistics in general.

In the case of our application in linear cryptanalysis this problem can be solved, since the set $\mathcal{P}$ has the property that each element in $\mathcal{P}$ has the same divergence I^λ with the p.d. p. Also all the components of each p.d. in $\mathcal{P}$ are positive. Now the set of alternative p.d.'s to be considered is

$$\mathcal{P} = \left\{p' \neq p : I^\lambda(p', p) = I^\lambda, p'_\eta > 0 \text{ for all } \eta \in \mathcal{X}\right\}. \tag{5.13}$$

Provided that each $p' \in \mathcal{P}$ also satisfies condition

$$\max_{\eta \in \mathcal{X}}(p'_\eta - p_\eta) = \mathcal{O}(N^{-1/2}), \tag{5.14}$$

when $N \to \infty$, Drost, et al., calculated a series expansions for the power of the test for all λ [19]. We are interested in the parameter values $\lambda = 1$ (the χ^2-test) and $\lambda = 0$ (the G-test).

5.2.2 Solving Goodness-of-Fit with χ^2

If $\lambda = 1$, the test statistic $R^1(\mathbf{Q} : p)$ reduces to the Pearson's χ^2-statistic:

$$\chi^2(x_1, \ldots, x_N) = \chi^2(q) = N \sum_{\eta \in \mathcal{X}} \frac{(q_\eta - p_\eta)^2}{p_\eta}. \tag{5.15}$$

We denote $\chi^2(q; p)$, if we want to emphasise the null p.d. p.

The asymptotic distribution of $\chi^2(\mathbf{Q}; p)$ is [19]

$$\chi^2(\mathbf{Q}; p) \sim \chi^2_M(\nu), \tag{5.16}$$

where $\nu = 2NI^1(p : p) = NC(p, p) = 0$, for H_0 and $\nu = 2NI^1(p' : p) = 2NI^1 > 0$ for H_1. Hence, for fixed power $1 - \beta$, we have

$$\beta = \Phi\left(\frac{\tau - M - 2NI^1}{\sqrt{2M + 8NI^1}}\right),$$

provided that $M \geq 50$ such that we can approximate χ^2 with normal distribution. Then by (5.12) the threshold $\tau \approx \sqrt{2M}\Phi^{-1}(1 - \alpha) + M$. Since $\alpha \approx \beta$, we have

$$2NI^1 \approx \Phi^{-1}(1 - \alpha)\sqrt{2M}.$$

Hence, we have the following result:

Lemma 5.5. *Consider a discrete goodness-of-fit problem where H_0 states that the sample population is p and H_1 states that the population is in the set $\mathcal{P}$, given as in* (5.13) *with capacity $2I^1 = C(p', p)$, for all $p' \in \mathcal{P}$. Assume that each $p' \in \mathcal{P}$ satisfies* (5.14). *If the χ^2-test given by statistic* (5.15) *is used for solving this problem and the degree of freedom $M \geq 50$, then the data complexity of distinguishing between hypotheses H_0 and H_1 is proportional to*

$$N = M^{1/2}/I^1.$$

The problem of distinguishing p from the set $\mathcal{P}$ is similar to the problem in Section 5.1.4, where each element p' of the alternative hypothesis was given. If in the latter case each $p' \in \mathcal{P}$ has the same capacity $C(p', p) = 2I^1$ with p, then the data complexity is $\log |\mathcal{P}|/I^1$. If $|\mathcal{P}| = M$, the expected data complexity of the χ^2-test is significantly larger than for the binary HTP. This is because in the latter setting each p' is known, whereas in the goodness-of-fit problem we only know the capacity. If the set $\mathcal{P}$ is unambiguously determined such that we know each $p' \in \mathcal{P}$, it is better to use LLR to distinguish p from $\mathcal{P}$, see Section 5.1.4. On the other hand, if we only know that $\mathcal{P}$ is given by (5.13), then we must use a goodness-of-fit test, such as χ^2.

5.2.3 Solving Goodness-of-Fit with G-test

If $\lambda = 0$, the statistic $R^0(\mathbf{Q} : p)$ is equivalent to the Kullback-Leibler distance between q and p:

$$R^0(q : p) = D(q\|p). \tag{5.17}$$

The statistic is called, for example, the G-test, the log-likelihood test and G^2-test. Again, if $D(q\|p)$ is larger than some threshold value, the null hypothesis is rejected.

Drost, et al., showed that

$$R^0(\mathbf{Q} : p) \sim \chi^2_M(\nu) + \xi. \tag{5.18}$$

If H_0 is true, then the parameters are $\nu = \xi = 0$. If H_1 true, then

$$\nu = N \sum_{\eta \in \mathcal{X}} p'_\eta \log^2 \frac{p'_\eta}{p_\eta} - ND(p'\|p)^2 \quad \text{and} \quad \xi = 2ND(p'\|p) - \nu.$$

If p' is close to p in the sense of definition (3.15) then $\nu \approx NC(p', p)$ and $\xi \approx 0$. Hence, the χ^2-statistic and G-statistic are equal, which is also noted in [19].

Assume now that $M = 2^m - 1$ for some positive m such that $\mathcal{X} = \mathbb{F}_2^m$. We now show that when we have the data complexity N at most proportional to $2^{m/2}/I^1$, as suggested by Lemma 5.5, then the condition (5.14) follows from condition (3.15). The latter condition is easy to verify and holds in the practical situations we are considering. Hence, the condition (5.14) holds for data complexities that do not significantly exceed the limit $2^m/I^1$ and we can safely use the theory given by Drost, et al. [19]. On the other hand, practical experiments done for example in **IV** show that we can successfully distinguish between H_0 and H_1 with (at most) the predicted data complexity, and it is not necessary to consider N larger than $2^{m/2}/I^1$.

Assume now that p' is close to p in the sense of definition (3.15). Then $\max_{\eta \in \mathcal{X}} |p'_\eta - p_\eta| \leq \varepsilon\zeta$, for some positive $\varepsilon < 0.5$ and $\zeta = \max_\eta p_\eta$. Let $A = \zeta 2^{m/4-1}/C(p', p)^{1/2}$. Then

$$\max_{\eta \in \mathcal{X}} |p'_\eta - p_\eta| \leq \varepsilon\zeta < 0.5\zeta = AC(p', p)^{1/2} 2^{-m/4} = AN^{-1/2},$$

and condition (5.14) is satisfied and the approximate formulas (5.16) and (5.18) for the power of χ^2-test and log-likelihood can be used.

5.3 THE d-SAMPLE PROBLEM AND RANKING STATISTICS

In a basic d-sample problem we have d distributions $\mathcal{D}_1, \ldots, \mathcal{D}_d$ and the goal is to determine whether they are all equal or not [23]. In a non-parametric setting we know nothing about the distributions and we can only solve the hypothesis testing problem, where H_0 states that $\mathcal{D}_1 = \cdots = \mathcal{D}_d$ and H_1 states that at least one $\mathcal{D}_i$ is different. In a parametric setting we have a model for the distributions, for example, all $\mathcal{D}_1, \ldots, \mathcal{D}_d$ are normal distributions with equal variances and we may find information about their means.

In either case, the problem is solved by drawing from each distribution $\mathcal{D}_i$ a random sample $\mathbf{X}_{i,1}, \dots, \mathbf{X}_{i,N_i}$ of size N_i. Assume for simplicity that $N_i = N$ for all $i = 1, \dots, d$. Each random sample is statistically independent of the other samples. A statistic $\mathbf{M}_i = g(\mathbf{X}_{i,1}, \dots, \mathbf{X}_{i,N})$ is calculated for each sample. The realised value M_i of the random variable $\mathbf{M}_i$ is called a mark [43]. The statistics should be chosen such that $\mathbf{M}_i$ has a distribution $\mathcal{D}'_i$ that depends on the sample distribution $\mathcal{D}_i$. The distributions $\mathcal{D}_i$ or equivalently, the parameters i, are *ranked* by sorting them according to their marks.

Consider the order statistics $\mathbf{M}_{(1)}, \dots, \mathbf{M}_{(d)}$ of the d independent random variables $\mathbf{M}_1, \dots, \mathbf{M}_d$. If M_i is the realised value of the jth order statistic $\mathbf{M}_{(j)}$ for some j, then we call j the *rank* of i. The statistics g is called the *ranking statistics*. The resulting order statistics depends on the distributions $\mathcal{D}_1, \dots, \mathcal{D}_d$ and hence, the order statistics can be used in recovering information about $\mathcal{D}_1, \dots, \mathcal{D}_d$.

In this thesis, we consider the following special case, which we call the d-sample distinction problem. For an unknown $\omega \in \{1, \dots, d\}$, one p.d. $\mathcal{D}_\omega = \mathcal{D}_R$, a possibly unknown p.d. The other $d-1$ distributions $\mathcal{D}_i$, $i \neq \omega$ are equal to a given p.d. $\mathcal{D}_W \neq \mathcal{D}_R$. Then we have d independent random variables $\mathbf{M}_1, \dots, \mathbf{M}_d$, one of which follows distribution $\mathcal{D}'_R$ and the others are drawn from $\mathcal{D}'_W$. The goal is to determine the parameter ω, that is, to state which sample is taken from population $\mathcal{D}_R$. The ranking statistic g gives the solution if it gives the highest rank to the parameter ω, such that $\mathbf{M}_\omega = \max_{i=1,\dots,d} \mathbf{M}_i = \mathbf{M}_{(1)}$. We will later consider how to properly choose g.

The d-sample distinction problem resembles the simple d-ary HTP in Section 5.1.3 since in both cases we have d possible parameter values among which we have to find the correct one. However, the two problems are different. In the simple d-ary HTP the same data is used in calculating the test statistics for each parameter value and the random variables $\mathbf{X}_1, \dots, \mathbf{X}_N$ are i.i.d., whereas different data is drawn from each distribution to calculate $\mathbf{M}_1, \dots, \mathbf{M}_d$ and the independent random variables $\mathbf{M}_1, \dots, \mathbf{M}_d$ do not follow the same distribution in a d-sample distinction problem.

Key ranking studied in Section 6.2.2 is an application of the d-sample distinction problem.

6 LINEAR CRYPTANALYSIS IN ONE DIMENSION

This chapter discusses the one-dimensional linear cryptanalysis, proposed by Matsui 1993 [31]. We start by defining what we mean by a one-dimensional linear approximations in Section 6.1. We also give a general framework for finding these approximations. In Section 6.2 we show how the one-dimensional approximation and the results of Chapter 5 can be used for realising different kinds of statistical attacks. We consider distinguishing attacks, key recovery attacks for block ciphers and initial state recovery for stream ciphers, proposed by Berbain, et al., in [4].

6.1 ONE-DIMENSIONAL LINEAR APPROXIMATIONS

6.1.1 Linear Approximation of a Vector Boolean Function

Let the input of a multidimensional Boolean function $f : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^n$ be $x \in \mathbb{F}_2^{n'}$. The one-dimensional linear approximation of f with input mask u and output mask w is then the Boolean function

$$x \mapsto u \cdot x \oplus w \cdot f(x), \tag{6.1}$$

and its correlation $c(u \cdot x \oplus w \cdot f(x), 0)$ is denoted by $c_f(u; w)$ or $c(u; w)$, if f is clear from the context. If there are two inputs x_1 and x_2 to f with corresponding input masks u_1 and u_2, the correlation $c(u_1 \cdot x_1 \oplus u_2 \cdot x_2 \oplus w \cdot f(x_1, x_2), 0)$ is denoted by $c_f(u_1, u_2; w)$.

We say that the approximation (6.1) is *strong* if it has a correlation that is non-negligible, i.e., the absolute value of the correlation is large enough to be exploited in a statistical attack. The next section studies some special functions that are often used in symmetric ciphers.

6.1.2 Approximations for Some Special Functions

Let $f(x_1, x_2) = x_1 \oplus x_2$, where $x_1, x_2 \in \mathbb{F}_2^n$. Then the only linear approximation of f with non-zero correlation is $w \cdot f \oplus w \cdot (x_1 \oplus x_2)$ that is, the input and output masks must be equal.

Let f be some Boolean function with linear approximation given by

$$u \cdot x \oplus w \cdot f(x),$$

and correlation $c_f(u; w)$. Let $f = L \circ g$, where L is a linear mapping and g is a Boolean function. By (3.1), there is a mask $w_L = L^T w$ such that g has linear approximation

$$u \cdot x \oplus w_L \cdot g(x), \tag{6.2}$$

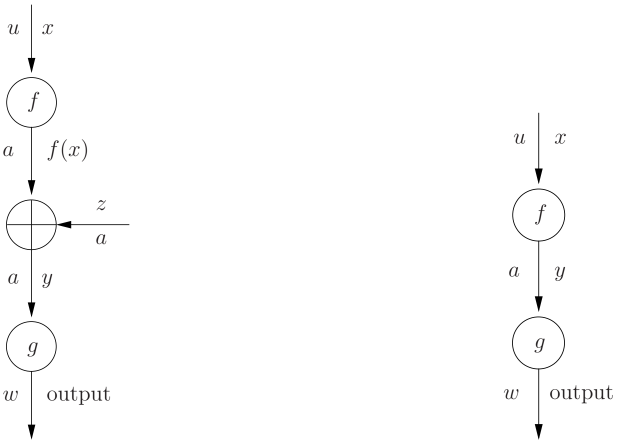
with correlation $c_g(u; w_L) = c_f(u; w)$. Hence, if it is straightforward to find the output mask and correlation for g given the output mask and correlation for f.

In [37] Wallén and Nyberg found a formula for determining the correlations $c_f(u_1, u_2; w)$, if f is sum modulo 2^n of its n-bit inputs. The result can be generalised for more than two inputs. This result is needed from example when linearising the stream ciphers SOSEMANUK and SNOW2.0, see Example 4.2.

6.1.3 Combining Approximations over Consecutive Functions

Non-linear filter functions and round functions of iterated block ciphers consist typically of several consecutive non-linear functions. Moreover, the round function of an iterated block cipher is used repeatedly. We now show how we can determine the correlation over the whole cipher by determining the correlations over parts of the cipher. The different approximations are combined to a "trail" through the cipher.

Assume two multidimensional Boolean functions $f : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^k$ and $g : \mathbb{F}_2^k \mapsto \mathbb{F}_2^n$ with inputs x and y. Let the linear approximation of each function be $u \cdot x \oplus a \cdot f(x)$ and $a \cdot y \oplus w \cdot g(y)$ with correlations $c_f(u;a)$ and $c_g(a;w)$, respectively. Recall that we can associate random variables with Boolean functions and their inputs.



(a) Piling Up lemma: since z is s.i. of x and y, the functions f and g are s.i. Note that a must be the mask of z and y, also.

(b) Correlation theorem: The functions f and g are statistically dependent

Figure 6.1: Obtaining correlation over several consecutive functions. The input to f and g are x and y, respectively. The input and output masks are u and w, respectively, and a is called the middle-mask.

First we study the case where f and g are s.i., see Fig. 6.1(a) for an example of such a situation. A third variable z is used as an input to the system such that $y = z \oplus f(x)$. If z is uniformly distributed and independent of x, then x and y are s.i. Hence, $f(x)$ and $g(y)$ are s.i. For example, in a non-linear filter generator z can be a block taken from the LFSR and x is another block such that z has no common bits with x.

Consider the s.i. binary random variables $\mathbf{X}$ and $\mathbf{Y}$, associated with $u \cdot x \oplus a \cdot f(x)$ and $a \cdot y \oplus w \cdot g(y)$, respectively. By Piling up Lemma 3.3 the correlation $c_{g\circ f}(u,a;w) = c(u \cdot x \oplus a \cdot z \oplus w \cdot g(f(x)), 0) = c(\mathbf{X} \oplus \mathbf{Y}) = c(\mathbf{X})c(\mathbf{Y})$. Hence, the correlation over the two functions is

$$c_{g\circ f}(u,a;w) = c_f(u,a)c_g(a,w). \qquad (6.3)$$

Sometimes there is no "randomising" input z between f and g, see Figure 6.1(b). Then the consecutive functions are not s.i. For example Nyberg studied this problem in [36] and proved the following result:

Theorem 6.1 (Correlation Theorem). *Let $f : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^k$ and $g : \mathbb{F}_2^k \mapsto \mathbb{F}_2^n$ be Boolean functions. Let $u \cdot x \oplus a \cdot f(x)$ and $a \cdot y \oplus w \cdot g(y)$ be the linear approximations of f and g with correlations $c_f(u; a)$ and $c_g(a; w)$, respectively. Then the correlation of $u \cdot x \oplus w \cdot g(f(x))$ is given by the sum*

$$c_{g \circ f}(u; w) = \sum_{a \in \mathbb{F}_2^k} c_f(u; a) c_g(a; w). \tag{6.4}$$

When using Correlation theorem, all the correlations over all the "middle-masks" a need to be considered. If there are several non-negligible linear trails through the cipher, they are called the linear hull of the cipher [35].

Often the cipher has several consecutive functions. This holds for example for block ciphers, where the approximations have to be done over dozens of rounds. Therefore, it is not feasible to calculate the accurate correlation using Theorem 6.1. Instead, the cryptanalyst assumes that the round keys are statistically independent of each other and the input x. The total correlation of the cipher is then approximated by the Piling Up lemma. If there is only one strong linear trail through the cipher, the Piling Up lemma gives an accurate result.

If there are more than one strong linear trail the actual correlation maybe smaller or larger than the Piling Up lemma states. Since some of the correlations in the sum (6.4) may be negative, there is no equality similar to (6.3). The following inequality holds when the Correlation theorem is used:

$$|c_{g \circ f}(u, w)| \leq \sum_{a \in \mathbb{F}_2^k} |c_f(u; a)||c_g(a; w)|. \tag{6.5}$$

Hence, the absolute value of the total correlation is only upperbounded by the absolute values of the correlations over parts of the system.

6.1.4 Linear Approximation of a K.S.G

Let us study a simple example of a key stream generator (k.s.g.) consisting of an LFSR with L state blocks of size n bits each, and a filter function $f : \mathbb{F}_2^l \mapsto \mathbb{F}_2^n$, where l is a multiple of n. See also Figure 4.4. The LFSR recursion coefficients in (4.1) are $b_0, \ldots, b_{L-1} \in \mathbb{F}_2^n$, where $b_0 \neq 0$.

At time t, the output of the k.s.g. is $z_t = f(S_t)$, where the input $S_t \in \mathbb{F}_2^l$ is some fixed subset of the LFSR state blocks $s_t, \ldots, s_{t+L-1}$. The analyst tries to find a strong approximation

$$w \cdot z_t \oplus v \cdot S_t, \text{ for all } t \geq 0 \tag{6.6}$$

for some masks $w \in \mathbb{F}_2^n$ and $v \in \mathbb{F}_2^l$. The correlation is $c(v; w)$. Approximation (6.6) is applicable to some stream ciphers directly, see Sect. 6.2.3. However, usually the analyst has to cancel the internal state words s_t, $t \geq 0$ from (6.6) using the linear recursion (4.1).

Assume for simplicity that each $b_i \in \{0, 1\}$, since the theoretical bounds presented in **II** are derived for binary coefficients only. Define the set of indices corresponding to non-zero coefficients by $J = \{i = 0, \dots, L-1 : b_i \neq 0\}$. We obtained in **II** the linear approximation containing only the keystream words

$$w \cdot \bigoplus_{j \in J} z_{t+j}, \text{ for all } t \geq 0, \tag{6.7}$$

with correlation $c(w)$ given by

$$c(w) = \sum_{v \in \mathbb{F}_2^n} c(v; w)^{|J|}. \tag{6.8}$$

The formula (6.8) shows that all the input masks v should be considered to find the correct value of the correlation. However, the correlation (6.8) is often approximated by

$$c(w) \approx c(v; w)^{|J|},$$

for some chosen input mask $v \in \mathbb{F}_2^l$ for which the correlation $c(v; w)$ of (6.6) is large. The simple examples studied in Section 8.2 show that this approximation can be severely flawed.

6.1.5 Linear Approximation of a Block Cipher

Consider an iterated block cipher with plaintext x and ciphertext y after R rounds, see also Figure 4.1. We denote by K_R the expanded (inner) key, that is, a vector consisting of all (fixed) round key bits used in R rounds, see Section 4.1. A linear approximation of this block cipher over R rounds is given by the Boolean function

$$u \cdot x \oplus w \cdot y \oplus v \cdot K_R. \tag{6.9}$$

The vectors u and w are the input and output mask, respectively. The vector v is called the key mask.

We can determine the correlation c of the approximation (6.9) using the methods of Sect. 6.1. We assume that each round key is independent of the other keys. Then we can use the Piling Up lemma (6.3) to obtain an approximation for the correlation. The trail through the block cipher given by the masks is called its linear trail. The Correlation theorem 6.1 gives a more accurate result, however, it is usually unfeasible to use in practice.

Since the ciphertext $y = E_{K_R}(x)$ depends on the key, the actual correlation c depends on the key, also, and Piling Up lemma gives just an approximation for the correlation. Hence, it is possible that some keys cannot be recovered using linear cryptanalysis, because the corresponding correlation is too small. On the other hand, for some keys the approximative correlation c is a lower bound for the true correlation [35], and linear cryptanalysis is stronger than predicted. For the rest of this thesis, we assume that we have a good approximation c of the true correlation of (6.9) and c is independent of the key.

6.2 ATTACKS IN ONE DIMENSION

6.2.1 Distinguishing Attack

The output of a cipher should be pseudorandom, that is, it should not be possible to distinguish it the from the output of a random source. Hence, while a distinguisher is only able to determine whether a given sequence is produced by the cipher or not, resistance against linear distinguishers is considered an integral property of contemporary ciphers. The amount of data needed to realize the attack (with certain probability of success, say, 95%) is called the data complexity of the attack. A distinguisher is considered to be successful against a cipher if the data complexity is less than the complexity of an exhaustive search.

In practice, distinguishing attacks are applied only on stream ciphers. Although it is possible to realise a distinguishing attack against a block cipher the same statistical methods can be used in realising key recovery attacks, which we shall study in the next sections.

We now show how we can use approximation (6.7) for realising a distinguishing attack against a non-linear filter generator. We assume that the keystream words $z_1, \ldots, z_N$ are i.i.d. with uniform population. Let J be the set of non-zero indices defined in Section 6.1.4 and consider the linear approximation (6.7). If the sequence $z_1, \ldots, z_N$ is drawn from the cipher, then the observations $w \cdot \bigoplus_{j \in J} z_{t+j}$, $t = 1, \ldots, N$, are the realisation of a random sample from population Bernoulli$(1/2 + c/2)$, where c is the correlation of (6.7) given in (6.8). On the other hand, if $z_1, \ldots, z_N$ are taken from a random source, the terms in the sequence $w \cdot \bigoplus_{j \in J}^{L} z_{t+j}$, $t = 1, \ldots, N$, are the realisations of a random sample from population Bernoulli$(1/2)$. Hence, the problem of distinguishing the cipher from a random source is a HTP, where the correlation is the parameter.

The null hypothesis H_0 states that the random sample is drawn from a random source with correlation 0. If we know the correlation $c \neq 0$, we have a simple binary HTP, where the alternative hypothesis H_1 states that the random sample is drawn from the cipher with correlation c. On the other hand, if c is unknown, we have a goodness-of-fit problem, with H_1 states that the correlation of the sample is not 0.

In cryptanalysis, the attacker has to prove that the data is not random. Therefore, the goal is to reject the null hypothesis. The more data the analyst needs to successfully reject H_0, the stronger the cipher is against the distinguishing attack.

For known c, it is natural to use the Neyman-Pearson philosophy in the distinguisher, since the null hypothesis can be considered to have a special status. Recall that $q = (q_0, q_1)$ is the empirical p.d. defined by (3.10). By Section 5.1.2, the optimal distinguisher rejects H_0 if $\mathrm{LLR}(q; \theta, p)$ is smaller than some threshold. Junod [26] considered an equivalent method using the empirical correlation $\rho = 2q_0 - 1$. The null hypothesis is rejected, if ρ is bound from above (below) by a threshold that depends on c, when $c \geq 0$ $(c < 0)$.

The data complexity of the distinguisher with given power and level of test is given by Theorem 5.2 to be proportional to

$$N = \frac{1}{c^2}. \tag{6.10}$$

Matsui gave this result in [31].

For an unknown c, the problem is solved using the χ^2-test statistics defined in (5.15). The realised value of χ^2-statistic is ρ^2. Hence, we reject H_0 if ρ^2 is too large. If we use the power approximation on Section 5.2 by assuming that we know the value c^2 but not the sign of c, the data complexity is given by Lemma 5.5 to proportional to $1/c^2$. The constant coefficient of N for χ^2 can be larger than for the LLR but the tests are practically equal in this case. Obviously, if c^2 is not known, we have no knowledge about the data complexity and we have to collect data until we are satisfied with the level of the χ^2-test. Hence, the data complexity can only be determined by experiments.

6.2.2 Key Recovery for Block Ciphers

Matsui's Algorithm 1

Assume that we have found a one-dimensional approximation (6.9) with non-negligible, constant correlation c. The output y is the ciphertext obtained from the cipher by encrypting plaintext x over R rounds. We denote $z = VK_R$. Our goal is to find the one bit of information of the key.

We draw N plaintext-ciphertext pairs (x_t, y_t), $t = 1, \dots, N$, from the cipher and compute the empirical correlation ρ by

$$\rho = 2N^{-1}\#\{t : u \cdot x_t \oplus w \cdot y_t = 0\}.$$

We assume that the plaintexts $x_1, \dots, x_N$ are i.i.d. with uniform population. Then the values $u \cdot x_t \oplus w \cdot y_t \oplus z$, $t = 1, \dots, N$, are the realised values of a random sample from Bernoulli$(1/2 + c/2)$ and the observations $u \cdot x_t \oplus w \cdot y_t$, $t = 1, \dots, N$ are the realised values of a random sample from population Bernoulli$(1/2 + (-1)^z c)$. We can then determine z using ρ: The key bit z is chosen to be 0 if $c\rho > 0$. Otherwise, $z = 1$. Equivalently, we find the z that minimises $((-1)^z c - \rho)^2$ or $-(-1)^z c\rho$.

We can also consider finding z as a simple binary HTP. We have to decide using the empirical p.d. $q = (\frac{1}{2}(1 + \rho), \frac{1}{2}(1 - \rho)$, whether the sample population is $p^0 = (\frac{1}{2}(1 + c), \frac{1}{2}(1 - c))$ or $p^1 = (\frac{1}{2}(1 - c), \frac{1}{2}(1 + c))$. We can solve this problem for example with LLR, which is the same as using the decision algorithm above. The data complexity is obtained for example by Theorem 5.2, but already Matsui showed that it is proportional to $1/c^2$ [31].

Before considering Matsui's Algorithm 2, we define key ranking and the quantity that is used in measuring the efficiency of key ranking, the advantage.

Key Ranking

In key ranking we have a set of key candidates and the problem is to determine which key κ is the right one. Usually the keys are searched from the set $\mathbb{F}_2^n$ of all 2^n strings of n bits. In linear cryptanalysis, we make the following assumption:

Assumption 6.2 (Wrong-key Hypothesis)**.** There are two p.d.'s $\mathcal{D}_R$ and $\mathcal{D}_W \neq \mathcal{D}_R$ such that for the right key κ_0, the sample population is $\mathcal{D}_R$ and for a wrong key $\kappa \neq \kappa_0$ the population is $\mathcal{D}_W$.

Hence, the problem of finding κ_0 is the d-sample distinction problem with $d = 2^n$, given in Section 5.3. The problem can be solved by choosing a proper ranking statistic g for the keys. For each κ, the statistic g corresponds to a random variable $\mathbf{M}_\kappa$. The realised value M_κ of $\mathbf{M}_\kappa$ is called the mark of κ.

Vaudenay divides that attack to four phases: the *distillation phase*, the *analysis phase*, the *sorting phase* and the *search phase* [43]. The distillation phase is done on-line. We collect data from the cipher, for example, plaintext-ciphertext pairs and store it in a suitable way. In the analysis phase, for each key candidate κ, the mark M_κ is computed using g. The sorting phase is independent of g: we rank the candidates κ using their marks. Optimally, the right key, denoted by κ_0, should be at the top of the list. If this is not the case, then we must also run through a search phase, testing the keys in the list until κ_0 is found.

Biryukov, et al., measured the time complexity of the search phase given amount N of data using a special purpose quantity "gain" [6]. Selçuk defined a similar but more generally applicable concept of "advantage" in [40] as follows:

Definition 6.3. We say that a key recovery attack for an n-bit key achieves an advantage of a bits over exhaustive search, if the correct key is ranked among the top $r = 2^{n-a}$ out of all 2^n key candidates.

Hence, the time complexity of the search phase is 2^{n-a}. We can now derive a relationship between the data complexity N and advantage a. Assume that g ranks the keys is increasing order such that we expect M_{κ_0} to be the largest mark. Then we have for a fixed probability P_S and $r = 2^{n-a}$ that

$$P_S = \Pr(\mathbf{M}_{\kappa_0} > \mathbf{Y}), \tag{6.11}$$

where $\mathbf{Y}$ is the rth order statistic among the i.i.d. random variables $\mathbf{M}_\kappa, \kappa \neq \kappa_0$.

Assume that the ranking statistics g is normally distributed such that $\mathbf{M}_{\kappa_0} \sim \mathcal{D}'_R = \mathcal{N}(\mu_R, \sigma_R^2)$ and $\mathbf{M}_\kappa \sim \mathcal{D}'_W = \mathcal{N}(\mu_W, \sigma_W^2)$ for all $\kappa \neq \kappa_0$. Assume that $\mu_R > \mu_W$ and $\sigma_R^2 \approx \sigma_W^2$ such that $\sigma_R^2 \geq \sigma_W^2$. Then κ should have the highest rank among the parameters and the probability of success is given by (6.11). If 2^n is large, say, $n \geq 7$, by Theorem 3.11 the rth order statistic $\mathbf{Y}$ is normally distributed with mean

$$\mu = \Phi^{-1}_{\mu_W, \sigma_W^2}(1 - r/d) = \mu_W + \sigma_W \Phi^{-1}((1 - r/d),$$

and variance

$$\sigma^2 = \frac{(1 - r/d)r/d}{d\phi_{\mu_W, \sigma_W^2}(\mu)^2} = \frac{(1 - r/d)r/d^2}{\phi^2(b)}\sigma_W^2,$$

where $b = \Phi^{-1}((1 - r/d)$. Hence, $1 - \Phi(b) = r/d$. If r/d is small, then b is large and we may approximate $r/d = 1 - \Phi(b) \approx \phi(b)$. Hence, the variance becomes

$$\sigma^2 \approx \frac{(1-r/d)r/d^2}{(r/d)^2}\sigma_W^2 = \frac{1}{r}\left(1-\frac{r}{d}\right)\sigma_W^2 \approx \sigma_W^2/r.$$

Since the random variables $\mathbf{M}_k$ and $\mathbf{Y}$ are s.i. and normally distributed, we have

$$P_S = \Phi\left(\frac{\mu_R - \mu}{\sqrt{\sigma_R^2 + \sigma^2}}\right).$$

Note that this is slightly different from formula (9) in **IV**, since the variance σ^2 was directly approximated by zero. However, this affects the final results by only a constant coefficient that can be omitted.

The parameters μ_R, μ_W, σ_R^2 and σ_W^2 must satisfy

$$\frac{\mu_R - \mu_W - \sigma_W b}{\sqrt{\sigma_R^2 + \sigma_W^2/r}} = \Phi^{-1}(P_S). \tag{6.12}$$

The equation simplifies further if r is large and for small r if $\sigma_W^2 \approx \sigma_R^2$. All the means and variances depend on the data complexity N and hence, it is possible to find a relationship between the data complexity and P_S, for given advantage. Similarly, for fixed P_S, we can find a one-to-one mapping between the advantage and N. This lets us compare different ranking statistics, since the for fixed advantage and P_S the data complexity N should be as small as possible. On the other hand, we also know the trade-off between time complexity of the search phase and the data complexity.

Matsui's Algorithm 2

Selçuk considered key ranking for one-dimensional Alg. 2 in [40]. We study it here for completeness. Consider a block cipher with $R+1$ rounds, depicted in Figure 6.2. Let x be the plaintext and y' be the ciphertext after $R+1$ rounds. Let the round function and the $(R+1)$th round key be f and $k \in \mathbb{F}_2^l$, respectively. Then the output after R rounds is $y = f^{-1}(y', k) = E_{K_R}(x)$, where K_R is the key data over R rounds. Alg. 2 uses a strong linear approximation over R rounds given by (6.9) for finding the right last round key k_0 and possibly the right inner key bit $z_0 = VK_R$. Let the correlation be $c \neq 0$.

Consider first the straightforward Alg. 2 proposed in [31]. In the distillation phase we draw N plaintext-ciphertext pairs $(x_t, y_t'), t = 1, \ldots, N$. In the analysis phase, for each last round key k, we compute $y_t^k = f^{-1}(y_t', k)$, $t = 1, \ldots, N$ and obtain the empirical correlation

$$\rho^k = 2N^{-1}\#\{t : u \cdot x_t \oplus w \cdot y_t^k = 0\}. \tag{6.13}$$

The empirical correlations are the marks for the keys. This straightforward computing of empirical correlations takes time $N2^l$. In practice, this is not feasible. In [30] Matsui proposed a way to make the attack more efficient in practice for any SPN-cipher, whose round function is $y' = f(y) \oplus k$, where f consists of permutations and substitutions, see Section 4.1. Vaudenay followed this division in [43].

Since $y = f^{-1}(y' \oplus k)$, we only have to consider the l bits of y' corresponding to k. Let us denote these l bits by $y'(l)$ for simplicity. Hence, in

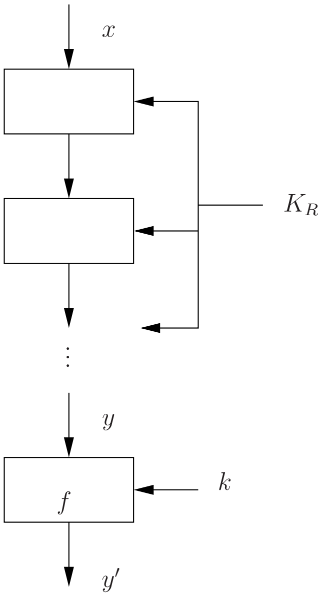


Figure 6.2: The linear approximation of an $R+1$-round block cipher for Algorithm 2. Notation: plaintext x, ciphertext y', round function f, last round key k, input to the last round y, key data in R-rounds K_R.

the distillation phase we collect the data and store it in a $2^l \times 1$ table T_D. For each plaintext-ciphertext pair (x_t, y'_t) we compute the parity $u \cdot x_t$ and store it such that the ith element in the table is

$$T_D(i) = 2N^{-1}\#\{t = 1, \ldots, N,\, y'_t(l) = i : u \cdot x_t = 0\} - 1.$$

This process takes time N.

In the analysis phase we run through the 2^l possible values of $y'(l)$, compute the decryption for key $k = 0$ by $y^0 = f^{-1}(y'(l))$ and obtain the parity bit $w \cdot y^0$. We store them in a table T_A. We can then obtain the other parity bits $w \cdot y^k$ by simply permuting the rows of T_A. It is then straightforward to obtain the bias for each k in time 2^{2l}, and the whole algorithm takes time $N + 2^{2l}$. Collard, et al., use FFT to reduce the time complexity to $N + l2^l$ [12]. Note that usually $N \gg 2^{2l}$. The data complexity of the attack does not depend on how we derive the empirical correlation ρ^k. Hence, we now assume that we are given the marks ρ^k and we show how they can be used for ranking the last round keys.

Let us now study the statistical model behind the Alg. 2 attack. Assume for simplicity that $c > 0$. If we use the wrong key $k \neq k_0$ to decrypt the ciphertext it means we essentially encrypt over one more round and the resulting data will be more uniformly distributed. This heuristics is behind the original Wrong-key Randomization Hypothesis, [31], [26], which we stated as Assumption 6.2. For each $\kappa \neq \kappa_0$, the observations $u \cdot x_t \oplus w \cdot y_t^k)$, $t = 1, \ldots, N$, are realised values of a random sample from population $\mathcal{D}_W =$ Bernoulli$(1/2)$. Consider now the ranking statistic whose realised value is the absolute value of the empirical correlation. By Section 3.3.4, it follows that $\mathcal{D}'_W = \mathcal{FN}(0, 1/N)$.

On the other hand, when decrypting with the correct key k_0, the observations $u \cdot x_t \oplus w \cdot y_t^{k_0}$, $t = 1, \ldots, N$ are the realised values of the random sample with population $\mathcal{D}_R$ = Bernoulli($1/2 + c/2$). Hence, the right key k_0 should have the highest mark and by Section 3.3.4, $\mathcal{D}'_R = \mathcal{N}(c, 1/N)$. Using (6.11) with given success probability P_S and advantage a, the data complexity is [40]

$$N = \frac{(\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-a-1}))^2}{c^2}.$$

After solving k_0 we have the empirical correlation ρ^{k_0}. It is then possible to use Alg. 1 for recovering the key parity bit z_0. The data complexity of Alg. 1 is less than the data complexity of ranking k_0 reasonably high. Hence, finding the key parity bit information is “free”.

6.2.3 Initial State Recovery for Stream Ciphers

Berbain, et al., used a linear approximation (6.6) for finding part of the initial state of the Grain stream cipher [4]. We describe the basic idea first. Let L and n be the length and the block size of the LFSR, respectively. Then the LFSR has 2^{Ln} possible initial states. Let $z_1, \ldots, z_N$ be the key stream output from the cipher and denote by Y_t the state of the LFSR at time $t \geq 0$. The linear approximation (6.6) can be written as

$$w \cdot z_t \oplus v \cdot Y_t, \text{ for all } t \geq 0, \tag{6.14}$$

where v is padded with zeros if necessary. It has correlation $c = c(v; w)$. Recall from Section 4.2.2 that $Y_t = A^t Y_0$, $t \geq 0$ where $Y_0 = Y$ is the initial state and A is given by (4.2). Using (3.1) we can write $v \cdot Y_t = (A^t)^T v \cdot y$, where T denotes transposition. Denote $v(t) = (A^t)^T u$. We can rewrite the approximation (6.14) as

$$w \cdot z_t \oplus v(t) \cdot Y,$$

and it still holds with the same correlation c for all $t \geq 0$. We assume as usual that the keystream words z_t, $t \geq 0$ are statistically independent. Hence, for given Y, we can draw N statistically independent samples from the population Bernoulli($1/2 + c/2$) by computing (6.14) for N consecutive times t.

We proceed by guessing the initial state Y. We obtain a sequence $z_1^Y, \ldots, z_N^Y$ from the cipher and we compute the empirical correlation by $\rho^Y = 2N^{-1}\#\{t : w \cdot z_t^Y \oplus v(t) \cdot Y\}$. If the guess is correct, then the sample population is Bernoulli($1/2 + c/2$). On the other hand, for any wrong guess, the LFSR state $v(t) \cdot Y$ and the keystream z_t should not have any correlation, that is, the sample population is Bernoulli($1/2$). This is again the Wrong-key Hypothesis, Assumption 6.2.

We must find the empirical correlation for all possible initial states Y. Then we have 2^{Ln} statistically independent random variables with realised values ρ^Y, $Y \in \mathbb{F}_2^{Ln}$. Moreover, since the sample population is the Bernoulli distribution (with either correlation c or 0), we have by Section 3.3.4 that each sample is normally distributed. For the right guess, the mean is $\mu_R = c$ and variance is $\sigma_R^2 = 1/N$. For all the wrong guesses, the mean is $\mu_W = 0$ and the variance is $\sigma_W^2 = 1/N$.

The problem of determining the right initial state is then the d-sample distinction problem studied in Section 5.3 and we can apply Selçuk’s key ranking theory described in Section 6.2.2. The mark for each initial state Y is the empirical correlation ρ^Y. We can find the data complexity using (6.12). If we wish to find the right initial state, we have $r = 1$, and obtain that the data complexity is proportional to

$$N = \left(\frac{\sqrt{2}\Phi^{-1}(P_S) + b}{c} \right)^2 ,$$

where P_S is a fixed success probability and $b = \Phi^{-1}(1 - 2^{-nL})$.

The problem with this straightforward approach is that the initial state is so large that it is not possible to run through the whole space $\mathbb{F}_2^{Ln}$. Instead, Berbain, et al., proposed the following method they called the second LFSR derivation technique. The purpose is to restrict the initial state Y to some sub-state of only, say, M bits. For simplicity, we may assume that we want to determine the M first bits in the LFSR. Then we only have to consider 2^M different Y. We denote the set of Ln-bit vectors, whose $Ln - M$ last components are zero, by Δ_M. We now show how to find many masks $v(t) \in \Delta_M$.

First, we sort the masks $v(t)$ according to their last $Ln - M$ bits. Then it is easy to divide the masks $v(t)$ to groups, where the $Ln - M$ last bits of the masks are the same. Let $v(t_1)$ and $v(t_2)$, where $t_1 \neq t_2$, belong to the same group. Then their XOR is in the set Δ_M. By Piling Up lemma, this approximation has correlation c^2. The number of pairs among N different $v(t)$ is $\binom{N}{2} \approx N^2$. The number of pairs, whose XOR is in Δ_M is on average $2^{m-Ln}N^2$. The number of masks $v(t) \in \Delta_M$ corresponding to correlation c is negligible when compared to the number of masks obtained by XOR. We may then omit them from the analysis.

The data complexity is proportional to [4]

$$N = 2^{(Ln-m)/2}/c^2.$$

After the M bits of the initial state are found, we may repeat the same procedure for some other initial state bits, provided that we find suitable approximations.

7 MULTIDIMENSIONAL LINEAR CRYPTANALYSIS

7.1 BACKGROUND

In one-dimensional linear cryptanalysis, the analyst tries to find a strong linear approximation such as (6.9) or (6.7). Sometimes it is possible to find several other approximations. That is, the analyst has, say m, approximations of the form

$$u_i \cdot x \oplus w_i \cdot f(x),\ i = 1, \ldots, m, \tag{7.1}$$

at disposal and each approximation has a non-negligible correlation c_i. The natural question is then if the analyst can somehow use all these approximations for either reducing the data complexity or for finding more information about the cipher. It is also important to know what is the best possible method for using all the approximations.

First Matsui in [30] and then Junod and Vaudenay in [26] used two approximations for key ranking in Alg.2. In [8], Kaliski and Robshaw considered m approximations of the form (6.9). They presented new versions of Matsui's Algorithms 1 and 2 assuming the same key mask for all approximations. They showed that the data complexity of finding one key parity bit is reduced when multiple approximations are used. However, they assumed that the approximations are statistically independent. As a different approach, Johansson and Maximov presented an idea of a multidimensional distinguishing attack against the stream cipher Scream [25].

Similarly as Kaliski and Robshaw, Biryukov, et al., used also the assumption about statistical independence, but they let the key mask vary [6]. Using their version of Alg. 1, they could determine m key parity bits with reduced data complexity. With a new version of Alg. 2, they could also determine the last round key. They measured the efficiency of their method using "gain". The method by Kaliski and Robshaw can be regarded as a special case of the method by Biryukov, et al., which we call the Biryukov method, for brevity.

In 2004, Baignères, et al., presented a true multidimensional distinguisher that did not rely on the assumption of statistical independence [1]. However, they did not provide a way to determine the p.d. that was needed in the attack. Englund and Maximov tried to solve this problem by determining the p.d. over a whole stream cipher for example in [21]. However, it is unfeasible to compute the p.d. directly if the word-size of the cipher is more than 32 bits. The problem of finding the multidimensional approximation in practice remained unsolved. Another open question was how Matsui's algorithms could be generalised to multiple dimensions.

The next sections give the answers to all these questions. Section 7.2 defines the multidimensional linear approximation. We show how the p.d. can be found efficiently and with no need to consider the whole wordsize or blocksize of the cipher. Only the necessary information, i.e., non-uniform behaviour of the cipher, has to be considered.

In Section 7.3 we study the distinguishing attack of Baignères, et al. Sections 7.4 and 7.5 give the generalisations for Matsui's algorithms. We give also the data, time and memory complexities for the algorithms in multiple dimensions. We conclude our findings of multidimensional Alg. 1 and Alg. 2

in Section 7.6. Finally, we show in Section 7.7, how multiple approximations can be used for making the one-dimensional initial state recovery attack of Section 6.2.3 more efficient.

7.2 MULTIDIMENSIONAL LINEAR APPROXIMATION

The m-dimensional linear approximation of a vector Boolean function $f : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^n$ is

$$Ux \oplus Wf(x), \tag{7.2}$$

where $U : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^m$ and $W : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ are linear mappings. If we have m linearly independent linear approximations (7.1), called *base approximations*, the multidimensional masks in approximation (7.2) are $U = (u_1, \ldots, u_m)^T$ and $W = (w_1, \ldots, w_m)^T$. The task is to find the p.d. p of the approximation.

By Lemma 3.8, the one-dimensional correlations $c(a) = c(a \cdot (Ux \oplus Wf(x)), 0)$, $a \in \mathbb{F}_2^m$ of all the one-dimensional approximations of f determine p. We search first for strong one-dimensional approximations over the whole cipher, where the correlations are given by the methods of Section 6.1. From the set of given strong approximations, we choose a suitable number m of linearly independent approximations and fix the multidimensional output masks. Then we determine the necessary correlations $c(a)$ and use Lemma 3.8 to find the p.d. p. Hence, we may omit negligible approximations and we can only consider information that is essential for the attack. If the correlations $c(a)$ are approximations of the true correlations, then p is also an approximation for the true p.d.

The most difficult task in linear cryptanalysis is finding the set of m strong base approximations. Moreover, it is not always easy to determine the correlations $c(a)$, $a \in \mathbb{F}_2^m$, even though the input and output masks are given. In this thesis we do not concentrate on the problem of finding the masks or the correlations. We assume that we have the approximations and our goal is to show the most efficient way to exploit the given information. We measure the efficiency of each method using the data complexity and if needed, also the time and memory complexities.

In practice, the p.d. p does not vary much from the uniform distribution θ. Hence, we assume that p is close to θ in the sense of the definition (3.15) and we can give the data complexities of the attacks using the capacity of p. Therefore, we should choose the base approximations such that the capacity $C(p)$ of the linear approximation (7.2) is as large as possible. We confirm the assumption about closeness in **IV** for the reduced round Serpent.

7.3 MULTIDIMENSIONAL LINEAR DISTINGUISHER

Consider a k.s.g. with the multidimensional approximation

$$W \bigoplus_{j \in J} z_{t+j}, \text{ for all } t \geq 0, \tag{7.3}$$

with p.d. $p \neq \theta$ and J the set of non-zero coefficients in the LFSR recursion, see Section 6.1.4. We assume that the keystream $z_t, \ldots, z_{t+N}$ is i.i.d. with uniform population. Then the corresponding approximations (7.3) for $t = 1, \ldots, N$ are the realisations of a random sample from the population p.

The distinguishing attack is equivalent to solving the binary HTP, where we have a random sample drawn from a discrete population and H_0 states that the population is θ [1]. The alternative hypothesis H_1 states that the population is $p \neq \theta$.

By Section 5.1.2, the optimal distinguisher for p and θ is given by the LLR test statistic LLR$(q; \theta, p)$. The empirical p.d. q is computed from the keystream by

$$q_\eta = N^{-1}\#\{t : W \bigoplus_{j \in J} z_{t+j} = \eta\}.$$

By Theorem 5.2, the data complexity is proportional to

$$N = 1/C(p).$$

Baignères, et al., obtained this result in [1]. However, they did not notice the relationship between p and the correlations $c(a)$. Hence, the distinguisher could not be used in practice.

By Lemma 3.9, the capacity is given by $C(p) = \sum_{a \neq 0} c(a)^2$, where $c(a)$ is the correlation of $a \cdot (W \bigoplus_{j \in J} z_{t+j})$. Hence, in an optimal case, the data complexity of using just one equation is reduced by the factor $1/(2^m - 1)$ when m equally strong approximations are used. If only one strong approximation is available, then it is not advantageous to use multiple approximations.

Vaudenay considered also the setting, where p is unknown [43]. The alternative hypothesis H_1 states that the sample population is not θ. The problem is solved using the χ^2-statistic defined in (5.15). By Lemma 5.5, the data complexity is upperbound by $2^{m/2}/C(p)$, which is significantly more than for the LLR-method. On the other hand, the capacity $C(p) \leq (2^m - 1)c^2$, where c is the correlation of strongest one-dimensional approximation, and in practice for large m the capacity increases slower than $2^{m/2}c^2$ when m increases. Hence, usually the χ^2-test does not benefit for using a large number m of approximations. If p is unknown, the most efficient method is based on the χ^2-test with a small number of linear approximations and if p is known, we should use LLR.

7.4 MULTIDIMENSIONAL EXTENSION OF MATSUI'S ALGORITHM 1

In this section we study Matsui's Alg. 1 in multiple dimensions. We show how the algorithm can be used for finding several parity key bits of the secret key used in a block cipher. We consider several different methods, proposed in **III**, **V** and **VII**. We also compare the methods and the Biryukov method in theory and in practice. We show that under certain conditions all the methods have equal data complexities. However, one method, the convolution method, is the most efficient in practice.

Section 7.4.1 defines the key recovery problem as a multiple HTP. We explain the Biryukov method in 7.4.2. In Section 7.4.3 we consider two multidimensional methods: the log-likelihood test or G-test from **III** and the χ^2-test from **V**. The optimal LLR-method proposed in **V** is studied in Section 7.4.4 and the convolution method, originally presented in **VII**, is studied in Section 7.4.5. Section 7.4.6 considers key ranking in Alg. 1 and Section 7.4.7 summarises the empirical tests done in **III**, **V** and **VII**.

7.4.1 Algorithm 1 as a Hypothesis Testing Problem

We use the same notation as in Section 6.2.2. The multidimensional Alg. 1 uses a linear approximation

$$Ux \oplus Wy \oplus VK_R, \tag{7.4}$$

where x and y are the plaintext and ciphertext and K_R is the expanded key data over R rounds. The matrix V divides the expanded keys to 2^m equivalence classes $z = VK_R \in \mathbb{F}_2^m$. The task is to find the right inner key class, denoted by z_0.

If $Ux \oplus Wy \oplus z \sim p$, then $Ux \oplus Wy \sim p^z$, a fixed permutation of p determined by z. Then all the p.d.'s p^z, $z \in \mathbb{F}_2^m$, are each other's permutations, and in particular,

$$p^z_{\eta \oplus a} = p^{z \oplus a}_{\eta}, \quad \text{for all} \quad z, \eta, a \in \mathbb{F}_2^m. \tag{7.5}$$

It then follows that

$$C(p^z) = C(p), \text{ for all } z \in \mathbb{F}_2^m. \tag{7.6}$$

Since p is close to θ, then also each p^z is close to θ, in the sense of definition (3.15). We showed in **V** that $\min_{z \neq z'} C(p^z, p^{z'}) = \min_{z \neq 0} C(p^z, p)$, which is a positive constant, denoted by $C_{\min}(p)$. Moreover, at least for block cipher Serpent, $C_{\min}(p) \approx C(p)$.

Consider N plaintext-cipher pairs taken from the cipher. We make the usual assumption that the plaintexts $x_1, \ldots, x_N$, are i.i.d., with uniform distribution. Then the values $Ux_t \oplus Wy_t$, $t = 1, \ldots, N$, are the realisations of the random sample from p^z.

Using the theory of Section 5.1.3, we can state the problem of finding z as a 2^m-ary HTP. Let $\mathbf{Q}$ be the random vector of relative frequencies in the random sample with population p^z, see definition (3.12). Each hypothesis H_z, $z \in \mathbb{F}_2^m$ states that $\mathbf{Q} \sim \text{multi}(N, p^z)$. If all the keys K_R are uniformly distributed and the key classes z have an equal number of elements, the hypothesis have equal *a priori* probabilities. If this is not the case, the chosen test statistic must be modified accordingly. We assume equal *a priori* probabilities, for simplicity, since the assumption holds in our experiments.

The realisation of $\mathbf{Q}$ is the empirical p.d q, given by

$$q_\eta = N^{-1} \#\{t = 1, \ldots, N : Ux_t \oplus Wy_t = \eta\}.$$

In **IV** and **V**, we call this part of the algorithm the *on-line* phase, see Figure 7.1.

The data complexity of the on-line phase is N and it depends on the chosen test statistic g. The time and memory complexities are Nm and 2^m, respectively. After finding q, we compute the values of the realisations $g(q; p^i)$

```
Output: empirical p.d. $q$
initialise $2^m$ counters $q_\eta, \eta \in \mathbb{F}_2^m$ ;
for $t = 1, \ldots, N$ do
    draw $(x_t, y_t)$ from cipher ;
    for $i = 1, \ldots, m$ do
        calculate bit $\eta_i = u_i \cdot x_t \oplus w_i \cdot y_t$;
    end
    increment counter $q_\eta = \#\{t : Ux_t \oplus Wy_t = \eta\}$, where $\eta$ is the
    vector $(\eta_1, \ldots, \eta_m)$ interpreted as an integer;
end
output $q/N$;
```

Figure 7.1: On-line phase for Alg. 1: Computing empirical p.d q

of the test statistic in *off-line* phase. The time and memory complexities of the off-line phase depend on g. Our division to on-line and off-line phases is natural in Alg. 1, since it is most efficient to compute q and then the marks. Before studying the different statistics we used in **III**, **V** and **VII**, we describe the Alg. 1 by Biryukov, et al.

7.4.2 Biryukov method for Algorithm 1

Biryukov, et al., proposed using the m linearly independent base approximations (7.1), by assuming them to be statistically independent [6]. For each approximation, they computed the empirical correlation ρ_i. Then they chose z that minimises

$$b(z) = \sum_{i=0}^{m} ((-1)^{z_i} c_i - \rho_i)^2.$$

We call this the basic Biryukov method.

The function $b(z)$ can be considered as an ℓ_2-distance between two vectors, whose components are given by the theoretical and empirical correlations. Each bit is determined independently of the other bits using the original Matsui's Alg. 1 [31] studied in Section 6.2.2.

The time and memory complexities of the attack in on-line phase are mN and m, respectively. Time and memory complexities in the off-line phase are $2^m m$ and m, respectively. Biryukov, et al., claimed that the data complexity of their method is proportional to $1/(c_1^2 + \cdots + c_m^2)$. However, this is not an accurate result. Murphy noted that the assumption about statistical independence of the base approximations does not hold in general [33]. In particular, linearly dependent approximations are also statistically dependent. Murphy also suggested to use the traditional measure of covariance of two linear approximations in verifying the assumption about linear independence. This method has been subsequently used by other researchers, for example in [22]. We noted in **VII** that the most natural way is to use the converse of the Piling Up lemma, see Theorem 3.3.

Biryukov, et al., also proposed an enhancement, where they added more linearly and statistically dependent one-dimensional approximations. We called in **VII** the method where all non-negligible correlations $c(a) = c(a \cdot$

```
Input: empirical p.d. $q$ and theoretical p.d.'s $p^z$
Output: key class $z'$
for key classes $z = 0, \dots, 2^m - 1$ do
    compute $h(z) = \sum_{\eta=0}^{2^m-1} q_\eta \log \frac{q_\eta}{p^z_\eta}$;
end
find $z'$ that minimises $h(z)$;
output $z'$;
```

Figure 7.2: Off-line for Alg. 1: Using log-likelihood method

$(Ux \oplus Wy \oplus z), 0)$, $a \in \mathbb{F}_2^m$ are used the full Biryukov method. For all $a \in \mathbb{F}_2^m$, we denote by $\rho(a)$ the empirical correlation corresponding to the Boolean function $a \cdot (Ux \oplus Wy)$. The full Biryukov outputs z if z minimises

$$b_F(z) = \sum_{a \in \mathbb{F}_2^m} ((-1)^{a \cdot z} c(a) - \rho(a))^2. \tag{7.7}$$

If the combined approximations were statistically independent, the data complexity would be proportional to $1/C(p)$, see Lemma 3.9. The time and memory complexity in the on-line phase are mN and 2^m, respectively, and in the off-line phase 2^{2m} and 2^m, respectively.

Since the linear combinations of base approximations cannot be statistically independent, we have no statistical proof for the validity of the method. Still, the experiments by Collard, et al., in [11] and our experiments in **III** show that the Biryukov method benefits from the use of linearly dependent approximations. It seems that the full Biryukov is as efficient as our multidimensional method that does not require the assumption about statistical independence. In **VII** we show that full Biryukov is actually just another version of a multidimensional method. Hence, the assumption of statistical independence is not necessary.

In the next sections we consider the different statistics we have used for solving the Alg. 1 key recovery problem.

7.4.3 Log-likelihood Method and χ^2-Method

In **III** we propose using a method that outputs $z \in \mathbb{F}_2^m$ minimising

$$h(z) = D(q \| p^z). \tag{7.8}$$

Hence, we test which p.d. p^z is closest to the empirical distribution in the Kullback-Leibler distance. We call the method the log-likelihood or G-method. The off-line phase of the log-likelihood method is given in Figure 7.2. Our calculations in **III** propose that the data complexity is proportional to

$$N = m/C_{\min}(p). \tag{7.9}$$

In **V**, we consider another approach to the problem, depicted in Figure 7.3: For each $z \in \mathbb{F}_2^m$, we test whether the sample population is p^z or not. Hence, for each z we have a goodness-of-fit problem that we solve using χ^2-statistic of Section 5.2. We call the method, where we choose the z that minimises

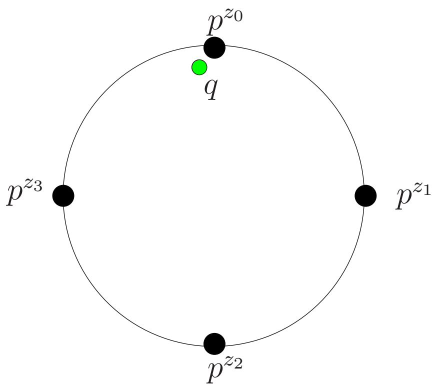


Figure 7.3: Alg. 1: The setting for goodness-of-fit when $m = 2$. The empirical p.d. q is near p^{z_0} and far away from p^z, $z \neq z_0$

$$s(z) = \chi^2(q; p^z) = \sum_{\eta \in \mathbb{F}_2^m} \frac{(q_\eta - p_\eta^z)^2}{p_\eta}, \tag{7.10}$$

the χ^2-method. The off-line phase for χ^2-method is the same as for the log-likelihood method given in Figure 7.2, with h replaced by χ^2.

We regard the problem of finding z_0 as distinguishing p^{z_0} from a set $\mathcal{P}$ of unknown alternative distributions. By property (7.6), the set $\mathcal{P}$ is given by (5.13) and we can use the power approximations by Drost, et al., [19]. Hence, the data complexity is given by Lemma 5.5 to be proportional to

$$N = 2^{m/2}/C(p). \tag{7.11}$$

Since each p.d. p^z is close to θ in the sense of (3.15), χ^2-statistic given in (5.15) is equivalent to the G-test. On the other hand, the log-likelihood method is equivalent to solving the goodness-of-fit problem using the G-test given in Section 5.2.3. Hence, we obtain in **V** that the log-likelihood and χ^2-method should have the data complexity

$$N = 2^{m/2}/C_{\min}(p). \tag{7.12}$$

Recall that $C_{\min}(p) \approx C(p)$.

The difference between the formulas (7.9), (7.11) and (7.12) can be explained as follows: We do not distinguish one distribution p^{z_0} from a set of p.d.'s $\mathcal{P}$. Instead, both log-likelihood method and χ^2-method given by (7.8) and (7.10), respectively, solve the multiple HTP of Section 5.1.3. This problem does not affect the power approximations of Section 5.2 and the χ^2-test has the same data complexity as the log likelihood test.

Consider first the data complexity for the binary HTP between two keys, z_0 and $z \neq z_0$ using $h(z) = D(q \| p^z)$. By Proposition 5.1, the error probability of choosing z instead of z_0 is

$$P_e = \Pr(\mathrm{LLR}(\mathbf{Q}; p^{z_0}, p^z) < 0) = \Phi\left(-\sqrt{NC(p^{z_0}, p^z)}/2\right).$$

We obtain that the data complexity is proportional to $N = C(p^{z_0}, p^z)^{-1}$. Hence, by Lemma 5.4, the total data complexity using $h(z)$ or $s(z)$ is given

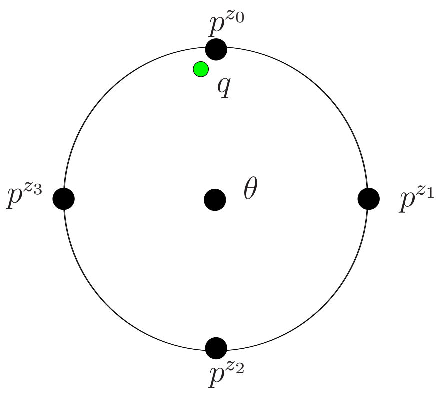


Figure 7.4: Alg. 1: The setting for LLR when $m = 2$. The empirical p.d. q is near p^{z_0} and far away from p^z, $z \neq z_0$ and θ

by (7.9). Note, that this result uses the definition (5.3) of the error probability and requires the closeness of the distributions.

From the description of the off-line phase in Figure 7.2 we can derive that the time and memory complexities for both log-likelihood and χ^2-method are 2^{2m} and 2^m, respectively.

7.4.4 The Optimal Test Statistic for Alg. 1

The optimal test statistic proposed in **V** is given by the LLR-statistic defined in (5.9). The distinguisher outputs z if $l(z) = \mathrm{LLR}(q; p^z, \theta)$ is maximised for z. We used the following heuristics to justify the LLR-method: For each $z \in \mathbb{F}_2^m$, we measure whether the data agrees better with p^z or θ. For the right key, denoted by z_0, the corresponding p.d. p^{z_0} should be easy to distinguish from θ and $l(z_0)$ should be large. On the other hand, for all the wrong keys $z \neq z_0$, the p.d.'s p^z and θ are indistinguishable and $l(z) = 0$ and for the "most wrong" key, namely, $\bar{z}_0 = \operatorname{argmax}_z D(p^z \| p^{z_0})$, the mark is negative. The heuristic is depicted in Figure 7.4. The data complexity is by Lemma 5.3 proportional to $N = m/C_{\min}(p)$

The off-line phase is again depicted in Figure 7.2, with $h(z)$ replaced by $l(z)$. Hence, the time and memory complexities for the off-line phase for both LLR-method are 2^{2m} and 2^m, respectively and the G-test, the χ^2-test and LLR are practically equal methods.

7.4.5 The Convolution Method

In **VII** we propose another method. It computes the convolution between the empirical p.d. q and the theoretical p.d. p. Hence, we call the method the convolution method.

The idea behind the method can be described as follows: The convolution $P = p * p$ is a p.d. of some r.v. The mode of the convolution is 0, that is, the largest probability $P_0 = \max_z P_z$. Similarly, for any $z \in \mathbb{F}_2^m$, the mode of $p^z * p$ is z. Hence, we consider the following approach: we find z that maximises

$$G(z) = (p * q)_z.$$

Rewriting $G(z)$ using (7.5) shows that it is a linear test with coefficients $\Lambda_\eta^z =$

```
Input: empirical p.d. $q$ and theoretical p.d. $p$ of the linear
       approximation (7.4)
Output: key class $z$
compute $p * q$ using FFT;
find mode $z'$ of $p * q$;
output $z'$;
```

Figure 7.5: Off-line phase of Alg. 1: Using convolution method

p_η^z mentioned in Section 5.1.3. Therefore, the convolution method has the same data complexity as the LLR-method.

On the other hand, the mark used in full Biryukov method can be rewritten using (3.20) that states that $c(a) = \hat{p}(a)$ and $\rho(a) = \hat{q}(a)$ and (3.21). In **VII** we obtain the following result:

Theorem 7.1. *The key z' minimises $b_F(z)$ if and only if it maximises $G(z)$. Hence, the full Biryukov method and the convolution method are equivalent.*

Since the convolution method does not need the assumption about statistical independence of the one-dimensional approximations, the assumption is also unnecessary for the full Biryukov method.

Now all the multidimensional Alg. 1 methods—the log-likelihood test (G-test), χ^2-test, LLR, full Biryukov and convolution—have the same data complexities given by Lemma 5.3. There is no difference in the on-line phases of the methods: The data, time and memory complexities are N, mN and 2^m, respectively. The memory complexity of both full Biryukov and convolution method can be reduced if we only use the strong approximations with non-negligible correlations $c(a)$, $a \in \mathbb{F}_2^m$. However, the same reduction works for both methods.

The off-line phase of the convolution method is depicted in Figure 7.5. Recall, that the time and memory complexities for the off-line phase of the full Biryukov, LLR and all the other multidimensional methods are 2^{2m} and 2^m, respectively. This is because we have to evaluate the statistic $g(q; z)$ separately for each z. In the convolution method, only one convolution is computed with Fast Fourier Transform (FFT). This takes time $m2^m$. The key is then directly given by the mode of the convolution. Hence, the total time complexity of the convolution method is also $m2^m$, which is the same as for the basic Biryukov method and significantly less than the time complexity 2^{2m} of the other methods. On the other hand, the data complexity of basic Biryukov is larger than the data complexities of the other methods. Therefore, of all the possible realisations of Alg. 1., the convolution method is most efficient in practice.

7.4.6 Ranking in Alg. 1

Similarly as in the d-sample problem in Section 5.3, we call in **V** the realised value of the test statistic $g(\mathbf{Q}; z)$ the mark of z. However, in Alg. 1 the r.v.'s $g(\mathbf{Q}; z)$ are not statistically independent. Hence, while there are several possible keys in multidimensional Alg. 1., the idea of ranking in Section 6.2.2 is not straightforward to generalise.

We assume in **V** that the r.v.'s $\mathrm{LLR}(\mathbf{Q}; p^z, \theta)$ used for ranking are independent and we obtain the following result about the advantage of LLR-based key ranking:

Theorem 7.2. *Assume that the r.v.'s* $\mathrm{LLR}(\mathbf{Q}; p^z, \theta)$ *are s.i. If the p.d.'s* p^z, $z \in \mathbb{F}_2^m$ *and* θ *are close to each other, in the sense of definition* (3.15), *the advantage of the LLR-method using statistic* (5.5) *can be approximated by*

$$a \approx \left(\frac{1}{2}\sqrt{NC(p)} - \Phi^{-1}(P_S)\right)^2, \tag{7.13}$$

where $P_S(\geq 0.5)$ *is the probability of success,* N *is the amount of data used in the attack and* $C(p)$ *and* m *are the capacity and the dimension of the linear approximation* (7.4), *respectively.*

The same holds for the other multidimensional Alg. 1 methods too, since they are equivalent for a binary HTP.

The assumption about statistical independence is needed in the derivation of (6.12), where we use Theorem 3.11. We are not aware of any general method of calculating the c.d.f. of the rth order statistic statistically dependent random variables. The asymptotic c.d.f. of the maximum of normal, identically distributed but dependent random variables for large $2^m - 1$, $m \geq 7$ is given in Corollary 3.13. However, the problem still remains that the random variables $g(\mathbf{Q}; z_0)$ and $\max_{z \neq z_0} g(\mathbf{Q}; z)$ are statistically dependent.

Denote by $N(z)$ the data complexity of ranking z with advantage a, if z is the right key. The assumption of statistical independence of $g(\mathbf{Q}; z)$'s could be avoided by drawing $\sum_{z \in \mathbb{F}_2^m} N(z) \approx 2^m \max_z N(z)$ words of data, as then the right key class z_0 would be ranked with advantage a and each mark $g(q; z)$, $z \in \mathbb{F}_2^m$ could be calculated from different data. However, the resulting complexity estimate is far too large to be of practical value.

In the next section we consider all the practical experiments presented in **III**, **V** and **IV** and compare the results with the given theoretical predictions.

7.4.7 Experiments with Alg. 1

The practical experiments are done on four-round Serpent, see Example 4.1. We use Serpent mainly because Collard, et al., used it as a test-bed for their experiments with the Biryukov method [11]. Moreover, due to Serpent's structure, it is possible to find several strong one-dimensional approximations that can be used in a multidimensional attack.

In **III** we compare the log-likelihood method to the Biryukov method, using quantity "gain" defined in [6]. We used $m = 10$ linearly independent base approximations computed over four-round Serpent. The results are depicted in Figure 1 in **III**. It shows that the gain of the log-likelihood method is larger than the gain of the Biryukov method for given data complexity. The tests also show that many linear combinations of the base approximations have large correlations and the linearly independent base approximations are statistically dependent. Thus, we confirm the claims by Murphy [33]. A more accurate description of the experiments, including the masks and correlations, is given in **III**.

In the experiments of **V** we use the advantage proposed by Selçuk in [40] instead of the gain. The two concepts are similar and the experimental results using either measure are consistent. The experiments confirm the claim in Section 7.4.3 that the two formulas (7.11) and (7.12) for the complexities of log-likelihood method and χ^2-method are over-estimations. The best approximation for the data complexity is given in (7.9), which is also the data complexity of the LLR-method of Section 7.4.4. Figures 1 and 2 in **V** depict the empirical and theoretical advantage for $m = 7$ and $m = 10$, respectively. Since the LLR-method is in practice the same as log-likelihood method and χ^2-method, the figures only depict the LLR-method.

The results show that the assumption about statistical independence of the r.v.'s $\mathrm{LLR}(\mathbf{Q}; p^z, \theta)$ has an effect: The method is in practice more efficient than the theory predicts. Hence, the theory gives too pessimistic results. On the other hand, the predictions are consistent and reasonably close to the empirical values.

7.5 MULTIDIMENSIONAL EXTENSION OF MATSUI'S ALGORITHM 2

In this section we study different ways to generalise Matsui's Alg. 2 to multiple dimensions. Most of the theory is presented in **IV**. In Section 7.5.1 we show how the theory of key ranking is applied to multidimensional Alg. 2. The Biryukov method is studied in Section 7.5.3. We consider two different approaches for solving the key recovery of Alg. 2: goodness-of-fit problem in Section 7.5.4 and a parametric distinguishing problem in Section 7.5.5. Section 7.5.6 studies how the convolution method can be used in Alg. 2. The experimental results of **IV** are described in Section 7.5.7.

7.5.1 Statistical Formalisation of Alg. 2

Recall the setting for the one-dimensional Alg. 2 in Section 6.2.2. The cipher has $R + 1$ rounds and x and y' are the plaintext and ciphertext, respectively. The round function and the $(R{+}1)$th round key (outer key) are f and $k \in \mathbb{F}_2^l$, respectively. The output after R rounds is $y = f^{-1}(y', k)$. Alg. 2 uses a linear approximation over R rounds given by (7.4) with p.d. p. We denote by k_0 and z_0 the right round key and right inner key class, respectively. The main goal of Alg. 2 is to find k_0 but it is also possible to find z_0.

Let us now consider the SPN-cipher, similarly in Section 6.2.2 and generalise Matsui's idea for the multidimensional algorithm. In the distillation phase we collect N plaintext-ciphertext pairs $(x_1, y_1'), \dots, (x_N, y_N')$, where the plaintexts are independent and uniformly distributed. Let d be the number of different values of Ux such that $d \leq 2^m$. In one dimension, we computed one table T_D of the biases of $u \cdot x$. Now we compute d tables T_D^{Ux} of size 1×2^l corresponding to each value of Ux. In each table, we store the frequencies of the different values of the l bits of y', denoted by $y'(l)$.

In the analysis phase we compute the decryption $y^0 = f^{-1}(y'(l))$ with key

$k = 0$ and store Wy^0 in a 1×2^l table T_A. Then for each Ux we combine the tables T_D^{Ux} and T_A and obtain the p.d. q^0 for $k = 0$. Permuting T_A lets us compute the other empirical p.d.'s q^k, $k \in \mathbb{F}_2^l$. The time complexity is then $N + 2^{2l+m}$. It is not clear if the trick proposed by Collard, et al., in [12], see Section 6.2.2, can be used in multiple dimensions. Nevertheless, usually $N \gg 2^{2l+m}$ and the time complexity is dominated by N.

In **IV**, we divide the algorithm to the on-line and off-line phases, similarly as in multidimensional Alg. 1. We compute all the empirical p.d.'s in the on-line phase and give the marks in the off-line phase. Our on-line phase has time complexity $N2^m$, which is infeasible. Therefore, in practice, we must divide the on-line phase to distillation phase (done on-line) and analysis phase (done off-line) as described above. The off-line phase consists of the marking phase, sorting phase and search phase.

In one-dimension the marks are simply the empirical biases and the marks are given in the analysis phase. In multiple dimensions we have to determine the marks after the empirical p.d.'s have been computed. This is the reason behind our original division to on-line phase and off-line phase: the complexities of the on-line phase are the same for all the statistical methods and the differences of the used statistics appear only in the off-line phase.

We now assume that we have computed the empirical p.d.'s q^k such that

$$q_\eta^k = N^{-1}\#\{t : Ux_t \oplus Wf^{-1}(y_t', k) = \eta\}, \text{ for all } \eta \in \mathbb{F}_2^m.$$

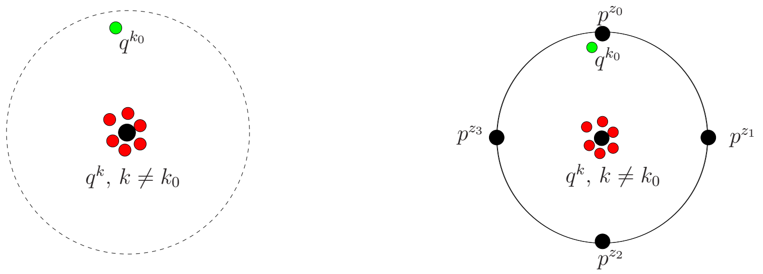
The remaining task is to give the marks for the keys in the marking phase. We concentrate on determining the most efficient way of using the acquired information for key ranking, that is, finding the best ranking statistic. For that, we need to consider the statistical model of the problem.

Similarly as in one-dimension, the Wrong-key Randomisation Hypothesis states that for each wrong round key candidate, deciphering with the wrong key produces uniformly distributed data that is statistically independent for different keys. Then Assumption 6.2 holds for $\mathcal{D}_W = \theta$. In other words, if $\mathbf{Q}^k$ denotes the vector of relative frequencies in the sample corresponding to the kth key, then $\mathbf{Q}^k$'s are s.i. for different k and $\mathbf{Q}^k \sim \text{multi}(N, \theta)$, for all $k \neq k_0$.

On the other hand, deciphering with the right key k_0 produces non-uniformly distributed data such that $\mathbf{Q}^{k_0} \sim \text{multi}(N, p^{z_0})$, where $\mathcal{D}_R = p^{z_0}$ is a fixed permutation of p that depends on the right key class z_0. The permutations have the properties (7.5) and (7.6) mentioned in Section 7.4. Both k_0 and z_0 are unknown, but in general, the test statistic $g(q^k; z)$ depends on both parameters. The problem now is determining k_0 without knowing z_0. If we are only interested in k, we can use a statistic that is independent of z. On the other hand, we can also try to find a unique z for each key k such that k_0 gets paired with z_0 and the right pair gets the highest mark. We consider the two options in the next section.

7.5.2 Different Scenarios in Multiple Dimensions

In **IV** we consider two different HTP settings for determining the last round key:



(a) Alg. 2 and χ^2-method: The wrong keys $k \neq k_0$ give empirical distributions q^k that are close to θ. The right key k_0 gives empirical distribution q^{k_0} that is further away from θ.

(b) Alg. 2 and LLR-method: The right key gives distribution that is close to p^{z_0} whereas the wrong keys are closer to θ than any p^z.

Figure 7.6: Alg. 2: Wrong-key Hypothesis for χ^2 and LLR when $m = 2$.

- Goodness-of-fit problem (usually solved with χ^2-statistic, see also [33] and [43]) and
- Distinguishing of an unknown p.d. from a given set of p.d.'s (Section 5.1.4)

The goodness-of-fit approach is a straightforward generalisation of the one-dimensional Alg. 2 of Section 6.2.2. It can be used for searching for the last round key. If we consider $\mathcal{D}_W = \theta$ as a given distribution and $\mathcal{D}_R = p^{z_0}$ as an unknown distribution, we have a goodness-of-fit setting: For each key candidate k we test between the null hypothesis H_0 stating that $\mathbf{Q}^k \sim \text{multi}(N, \theta)$ and alternative hypothesis H_1 stating that $\mathbf{Q}^k \nsim \text{multi}(N, \theta)$. The idea behind the goodness-of-fit approach is described in Figure 7.6(a).

By the symmetry properties (7.5) and (7.6), the set $\mathcal{P}$ of alternative p.d.'s is given by (5.13). Since the p.d.'s p^z are close to each other and the uniform distribution, we can use the power approximations by Drost, et al., given in Section 5.2. Therefore, we can restrict to using one divergence statistic, the χ^2-test and we call the method the χ^2-method. The method and mark in the goodness-of-fit approach do not depend on the inner key class z. Information about p.d. p is required only for measuring the strength of the test. After the right round key k is found, the data derived in Alg. 2 can be used in any form of Alg. 1 for finding the inner key class z. In this manner, this approach allows separating between Alg. 1 and Alg. 2. We study the χ^2-method in Section 7.5.4.

By Section 5.1.3, the LLR-statistic is the optimal distinguisher between two known p.d.'s. If we knew the right inner key class z_0, we could simply use the empirical p.d.'s q^k for distinguishing p^{z_0} and the uniform distribution and then choose the k for which this distinguisher is strongest [1]. In practice, the correct inner key class z_0 is unknown when running Alg. 2 for finding the last round key.

Our approach is the following. Recall that for each wrong key $k \neq k_0$, the random vector $\mathbf{Q}^k \sim \text{multi}(N, \theta)$. On the other hand, for the right key, $\mathbf{Q}^{k_0} \sim \text{multi}(N, p^{z_0})$, where $p^{z_0} \in \mathcal{P} = \{p^z : z \in \mathbb{F}_2^m\}$. Hence, for each key we consider the problem of distinguishing one known p.d. $\mathcal{D}_W = \theta$ from a given set $\mathcal{P}$ of p.d.'s. This is the HTP studied in Section 5.1.4. By

Baignères and Vaudenay, the optimal distinguisher for the problem is the LLR-statistic [2]. We call this LLR-based key ranking the LLR-method. The idea is described in Fig. 7.6(b).

In this case, we find the mark by fixing a unique inner key class z for each round key candidate k: We determine for each k the key class z, for which the LLR-statistic is the largest with the given data. The right key k_0 is expected to have z_0 such that the LLR-statistic with this pair (k_0, z_0) is larger than for any other pair $(k, z) \neq (k_0, z_0)$. Consequently, we also recover z_0 in addition to k_0. In this sense, the LLR-method is similar to the method presented in [6], where the Alg. 1 and Alg. 2 were combined together for finding both the outer and inner round keys. We study the LLR-method in Section 7.5.5.

We can use some other distinguishers for solving the HTP of Section 5.1.4. We consider the convolution method in Section 7.5.6, since it was the most efficient method for solving Alg. 1. Next we describe briefly the full Biryukov method for Alg. 2.

7.5.3 Biryukov method for Algorithm 2

Biryukov, et al., considered also Alg. 2 in [6], assuming that the one-dimensional approximations are statistically independent. Let us consider the full Biryukov method, where all the non-negligible one-dimensional approximations are used.

Draw N plaintext-ciphertext pairs. For each key candidate k and approximation $a \cdot (U \cdot x \oplus W \cdot y)$ with theoretical correlation $c(a)$, compute the empirical correlation $\rho^k(a)$. Consider the statistic with two parameters, z and k:

$$b(k, z) = \sum_{a \in \mathbb{F}_2^m} ((-1)^{a \cdot z} c(a) - \rho^k(a))^2 + \sum_{k' \neq k} \sum_{\eta \in \mathbb{F}_2^m} (\rho^{k'}(a))^2. \tag{7.14}$$

By the Assumption 6.2, the right round key should minimise the latter sum. On the other hand, given the right round key k_0, the first sum should be minimised for the right z_0. Hence, we use mark $b_k = \min_z b(k, z)$ and we choose the k with the smallest mark. The method gives also the class key z. The time and memory complexities of the analysis phase (see Section 6.2.2) are 2^{l+m} and 2^{m+l}, provided that $l \geq m$, which is usually the case.

7.5.4 Goodness-of-Fit Solution to Alg. 2

The mark for each key is given by the χ^2-statistic:

$$S_k = \chi^2(q^k; \theta) = 2^m N \sum_{\eta \in \mathbb{F}_2^m} (q_\eta^k - 2^{-m})^2. \tag{7.15}$$

The mark can be interpreted as the ℓ_2-distance between the empirical p.d. and the uniform distribution. By Assumption 6.2, the right round key should produce data that is farthest away from the uniform distribution and we choose the round key k for which the mark S_k is largest. Obviously, if $m = 1$, we get the $S_k = (\rho^k)^2$, where ρ^k is the empirical correlation given by (6.13). The marking phase for χ^2-method is given in Figure 4.

```
Input: table of empirical p.d.'s $q_\eta^k, k = 0, \ldots, 2^l - 1, \eta = 0, \ldots, 2^m - 1$
Output: store marks $S_k$ and possibly the corresponding p.d.'s $q^k$
for $k = 0, \ldots, 2^l - 1$ do
    compute $S_k = \sum_{\eta=0}^{2^m-1}(q_\eta^k - 2^{-m})^2$;
    if wish to recover $z_0$ then
        store $(S_k, q^k)$;
    else
        store $S_k$;
    end
end
```

Figure 7.7: Marking phase of Alg. 2 using χ^2-method. Determine the mark S_k for each k and store them. Store also q^k with the mark, if z_0 needs to be recovered.

The time complexity of the marking phase is 2^{l+m}. If we only store the marks S_k, the memory complexity is $2^l + 2^m$ and since usually $l > m$, the complexity is 2^l. If we want to determine z_0 using k_0, the memory complexity is 2^{l+m}. We do this in Alg. 1, using for example the convolution method of Section 7.4.5. Next we determine the advantage of the method as a function of the data complexity.

Formula (5.16) gives the distributions of the $\chi^2(\mathbf{Q}^k;\theta)$-statistic for the right and wrong keys. For all $k \neq k_0$, the statistic $\chi^2(\mathbf{Q}^k;\theta) \sim \chi^2_{2^m-1}$. By symmetry property (7.6), for the right key we have $\chi^2(\mathbf{Q}^{k_0};\theta) \sim \chi^2_{2^m-1}(NC(p))$. We can then use the normal approximations of χ^2-distribution and formula (6.12) to obtain the following result:

Theorem 7.3. *Suppose the cipher satisfies Assumption 6.2 where $\mathcal{D}_W = \theta$ and the p.d.'s p^z, $z \in \mathbb{F}_2^m$ and θ are close to each other. Then the advantage of the χ^2-method using mark* (7.15) *is given by*

$$a_{\chi^2} = \frac{(NC(p) - 4\varphi)^2}{2^{m+2}}, \; \varphi = \Phi^{-2}(2P_S - 1), \tag{7.16}$$

where P_S (> 0.5) is the probability of success, N is the amount of data used in the attack and $C(p)$ and m (> 5) are the capacity and the dimension of the linear approximation (7.4)*, respectively.*

We have to assume that $m > 5$, since for small m the χ^2-distribution does not have a simple asymptotic form.

For $m = 1$, the mark S_k reduces to the square of $|\rho^k|$ used by Selçuk. Hence, his theoretical derivations differ from our calculations and we get a slightly different formula for the advantage. Nevertheless, the methods are equivalent for $m = 1$.

Theorem 7.3 implies that the data complexity for given advantage is proportional to

$$N = \sqrt{2^m a}/C(p). \tag{7.17}$$

Hence, in order to strengthen the attack, the capacity should increase faster than $2^{m/2}$ when m is increased. This is a very strong condition and it suggests that in applications, only approximations with small m should be used with χ^2-attack.

While (7.16) and (7.17) depend on the theoretical distribution p, the actual method using (7.15) is independent of p. Hence, we do not need to know p accurately to realise the attack, we only need to find an approximation (7.4) that deviates as much as possible from the uniform distribution. On the other hand, if we use time and effort for computing an approximation of the theoretical p.d. and if we may assume that the approximation is accurate, we would also like to exploit this knowledge for finding the right inner key class with Alg. 1.

Recall the mark $b_k = \min_z b(k, z)$ used in the full Biryukov method in Section 7.5.3. For each k, the last sum in (7.14) corresponds to a goodness-of-fit test. If the sum is large, then one $k' \neq k$ produces a non-uniform distribution. On the other hand, if the sum is small, all $k' \neq k$ produce an output that is practically uniform. Hence, the right key k_0 should have the smallest value for the sum. By minimising the first sum over z, we determine for each k a unique key class z. This corresponds to a multiple HTP.

We can use Parseval's theorem to $b(k, z)$ and obtain

$$B(k, z) = \|q^k - p^z\|^2 + \sum_{k' \neq k} \|q^{k'} - \theta\|^2. \tag{7.18}$$

Then $B_k = \min_z B(k, z)$ is an equivalent mark with b_k and they both output the same key. Hence, we have shown that the assumption of statistical independence of base approximations is not necessary for Alg. 2.

We approximate $B(k, z)$ by

$$B'(k, z) = \chi^2(q^k; p^z) + \sum_{k \neq k'} S_{k'}. \tag{7.19}$$

Hence, the full Biryukov method joins the χ^2-tests for Alg. 2 and Alg. 1. It is not necessary to have the same coefficient for the sums in (7.19). Since the method is based on χ^2-statistic the full Biryukov has the advantage given by Theorem 7.3. Let us now consider the optimal method given by the LLR-statistic.

7.5.5 The Optimal Method for Alg. 2

The method is based on using the LLR-statistic defined in (5.5). We use the mark

$$L_k = \max_{z \in \mathbb{F}_2^m} \mathrm{LLR}(q^k; p^z, \theta). \tag{7.20}$$

Now k_0 should be the key for which this maximum over z's is the largest and ideally, the maximum should be achieved when $z = z_0$. While the symmetry property (7.5) allows one to develop statistical theory without knowing z_0, in practice we must search through $\mathbb{F}_2^l$ for k_0 and $\mathbb{F}_2^m$ for z_0 even if we are only interested in determining k_0.

The memory complexity of the marking phase is 2^{m+l}, which is the same as for the χ^2-method when z is determined. The time complexity for the marking phase is 2^{l+2m} (cf. Table 1 in **IV**, where it is 2^{m+l}). This is larger than for the χ^2-method. Next we determine the time complexity of the search phase, that is, the advantage for given data complexity.

We use Proposition 5.1 and obtain the following result in **IV**.

```
**Input**: table of empirical p.d.'s $q^k_\eta$, $k = 0, \dots, 2^l - 1$, $\eta = 0, \dots, 2^m - 1$
        and the theoretical p.d.'s $p$
**Output**: store mark $G_k$ and key class $z$ for each key candidate $k \in \mathbb{F}_2^l$
**for** $k = 0, \dots, 2^l - 1$ **do**
    compute $q^k * p$ using FFT;
    store $z(k) = \text{mode}(q^k * p)$ and mark $G_k = (q^k * p)_{z(k)}$;
**end**
```

Figure 7.8: Marking phase of Alg. 2 using the convolution method: We store one p.d. p and use it to determine the mark G_k. We can determine k_0 and z_0 simultaneously.

Theorem 7.4. *Suppose the cipher satisfies Assumption 6.2 where $\mathcal{D}_W = \theta$ and the p.d.'s p^z, $z \in \mathbb{F}_2^m$ and θ are close to each other. Then the advantage of the LLR-method for finding the last round key k_0 is given by*

$$a_{\text{LLR}} = (\sqrt{NC(p)} - \Phi^{-1}(P_S))^2/2 - m \approx NC(p) - m. \qquad (7.21)$$

Here N is the amount of data used in the attack, P_S (> 0.5) is the probability of success and $C(p)$ and m are the capacity and the dimensions of the linear approximation (7.4), *respectively.*

In the proof we assume that we pair the right key class z_0 with k_0. The data complexity of ranking k_0 paired with z_0 is according to (7.21) proportional to

$$N = (a + m)/C(p), \qquad (7.22)$$

where a is a fixed advantage. On the other hand, the data complexity of Alg. 1 is proportional to $N = m/C(p)$, which is at most the data complexity of Alg. 2. Hence, Theorem 7.4 gives the advantage of ranking the key k_0 paired with z_0 and it describes the trade-off between the search phase and the data complexity of the algorithm. With fixed N and capacity $C(p)$, the advantage decreases linearly with m whereas in (7.16) the logarithm of advantage decreases linearly with m. For fixed m and p, the advantage of the LLR-method is larger than the advantage of the χ^2-method.

7.5.6 The Convolution Method

Consider the convolution method of Section 7.4.5 used in multidimensional Alg. 1. We now show how the method can be applied to Alg. 2. We define the mark

$$G_k = \text{mode}(q^k * p). \qquad (7.23)$$

Hence, for each k we determine a unique z and the right round key k_0 should be paired with the right key class z_0. The marking phase is depicted in Figure 7.8.

The time complexity is $2^{l+m}m$, which is less than for the LLR-method and slightly more than for the χ^2-method. The memory complexity is the same as for LLR, 2^{m+l}. Note that we can also find $z(k)$ for each k using convolution method (in time $m2^m$) and then compute the mark $L_k = \text{LLR}(q^k; p^{z(k)}, \theta)$ in time 2^m. Empirical tests with different ciphers would show which approach is most efficient in practice.

We use the same approximations as in **VII**, where we prove that the convolution method in Alg. 1 has the same data complexity as the LLR-method. We get the following result:

Theorem 7.5. *Under the conditions of Theorem 7.4, the advantage of the convolution method is the same as for the LLR-method, given in Theorem 7.4.*

Hence, the LLR and convolution method have the same data complexities, whereas the χ^2-method has a significantly larger data complexity. On the other hand, the time complexity of the convolution method is slightly larger than for the χ^2-method and smaller than for the LLR-method.

In the next section we study the experimental results of **IV**.

7.5.7 Experiments with Alg. 2

Similarly as for Alg. 1, the experiments in **IV** are done on reduced round Serpent. We use a linear approximation over 4 rounds and obtain 12 bits of the 5th round key. The efficiency of the methods is compared in theory and practice by measuring the advantage for given amount of data. The results are presented in Figure 5 in **IV**.

We notice that the theoretical predictions agree with the empirical results. For χ^2-method, the optimal number of base equations is $m = 4$. Increasing $m \geq 5$ decreases the advantage. This agrees with Theorem 7.3. For $m < 5$, the theorem cannot be used and the results depend on the cipher. In general, we claim that $m = 4$ or $m = 5$ is the optimal number of approximations for χ^2. The full advantage of 12 bits is obtained at around $\log N = 28$ in both theory and practice.

For the LLR we see an increase in the advantage until $m = 12$. After that, the increase in capacity becomes negligible with the increase in m. This is consistent with Theorem 7.4. In general, the LLR-method gets stronger, when more approximations are used. Moreover, for given m, the LLR-method has a larger advantage than the χ^2-method both in theory and in practice. Both the empirical and theoretical curves show that the full advantage of 12 bits is obtained at $\log N = 26$ for LLR, using $m = 12$ approximations.

There are no empirical results about the convolution method for Alg. 2. We expect the results to be similar to those of Alg. 1.

7.6 CONCLUSIONS AND RECOMMENDATIONS FOR ALGORITHMS 1 AND 2

In Alg. 1 one method is obviously more efficient than any other method: the convolution method. It has the same data complexity, as all the other multidimensional methods, but the time complexity is significantly smaller than for the other methods. The empirical results confirm these predictions. Therefore, we suggest using the convolution method for multidimensional Alg. 1.

Table 7.1: Data and time complexities of the χ^2-method, the convolution method and the LLR-method for Alg. 2. We assume we recover z_0 after k_0 in the χ^2-method. The data complexities N_{χ^2} and N_{LLR} are given (7.17) and (7.22), respectively.

	Distillation			Marking		
	χ^2	conv	LLR	χ^2	conv	LLR
Data	N_{χ^2}	N_{LLR}	N_{LLR}	–	–	–
Time	N_{χ^2}	N_{LLR}	N_{LLR}	2^{l+m}	$m2^{l+m}$	2^{l+2m}

For Alg. 2 we do not have empirical results about the convolution method and we have to rely on the theoretical predictions and the results of Alg. 1. The time and data complexities of the different methods are given in Table 7.1. We only consider the distillation and marking phases as their complexities depend on the used statistic. For all the methods, the memory complexity of the marking phase is 2^{m+l}, if we wish to recover z_0.

The data complexities for the χ^2-method and LLR-method in Alg. 2 are given in (7.17) and (7.22), respectively. According to the theory, the convolution method is more efficient than the LLR-method. On the other hand, the χ^2 is slightly faster than the convolution method, but it has a significantly larger data complexity. Therefore, if a good approximation of the p.d. p of (7.4) is available, we suggest using the convolution method. Otherwise, χ^2 with at most $m = 4$ approximations must be used.

7.7 MULTIDIMENSIONAL INITIAL STATE RECOVERY FOR A K.S.G

We considered the one-dimensional LFSR initial state recovery attack in Section 6.2.3. It is tempting to try to generalise the method to multiple dimensions. However, this is not feasible in practice because the second LFSR derivation technique works only for one-dimensional masks. For multidimensional masks, there is no efficient way to generate an adequate number of input masks, whose rows belong to the set Δ_M. Since it is not feasible to consider the whole space of possible initial states Y, we cannot use the multidimensional approach. However, we can still exploit multiple approximations. We do this in **VI** and apply the results to the stream cipher SOSEMANUK.

Assume we have m approximations $w_i \cdot z_t \oplus v_i \cdot x_t$, $i = 1, \dots, m$, which hold at each time $t \geq 0$ with correlation c. Similarly as with one approximation, we find the time dependent input masks $v_i(t) = (A^t)^T v_i$, for all $i = 1, \dots, m$. Here A is the matrix given by the linear recursion of the LFSR (4.2). In Section 6.2.3 we had only one v, which generated all the other input masks $v(t)$. Now, we have m masks v_i, $i = 1, \dots, m$, each of which generates N masks $v_i(t)$. We proceed similarly as in the one-dimensional case. We sort the masks according to their $Ln - M$ last bits to groups. Then we XOR masks pairwise in a group to obtain input masks belonging to the set Δ_M.

Practical experiments showed that the number of masks in the same group is small, at most 3 and usually 0, 1 or 2. This is because the space of all possible masks is much larger than the space of generated masks $v_i(t)$. Consider two input masks in the same group. Due to the size of the input mask space $\mathbb{F}_2^{nL}$, the number of mask pairs in the same group for which $t_1 = t_2$ is negligible. Therefore, the XOR's have correlations c^2 and the output key words z_{t_1} and z_{t_2} give non-trivial data. The data complexity is proportional to

$$N = \frac{2^{(nL-M)/2}}{mc^2}.$$

Hence, we can reduce the data complexity by the factor $1/m$. The time complexity for recovering M bits of the LFSR is $M2^M + mN \log(mN)$. The memory complexity is $nmN + 2^M(M - n - 1 + 2\log(Nm))$.

Lee, et al., found a linear approximation for stream cipher SOSEMANUK with correlation $|c| = 2^{-21.4}$ [27]. Their data, time and memory complexities were $2^{145.5}$, $2^{147.9}$ and $2^{147.1}$. The structure of SOSEMANUK makes it possible to derive several strong approximations using one strong approximation. Therefore, we could reduce the complexities of the attack by using multiple approximations with correlations between $2^{-25.5}$ $2^{-21.4}$. The data, time and memory complexities of the new attack are $2^{135.7}$, $2^{147.4}$ and $2^{146.8}$.

8 MULTIDIMENSIONAL LINEARITY PROPERTIES OF BOOLEAN FUNCTIONS

In this chapter we consider some applications of multidimensional linear cryptanalysis and draw some theoretical bounds for the multidimensional attacks. In Section 8.1 we study the resistance of certain Boolean functions against linear cryptanalysis. Rothaus showed that bent functions—and only them—are optimal against one-dimensional linear cryptanalysis [39]. We give in **I** a new concept of multi-bent functions that are optimal in multidimensional linear cryptanalysis, since they achieve the smallest possible capacity. We also show that multi bent functions are the vector bent functions, defined in the classical way. We determine the capacities for some other functions, too.

Section 6.1.4 studies simple examples of a k.s.g. with some highly nonlinear filter functions. We see that the multidimensional method has a significantly smaller data complexity than the one-dimensional method. In Section 8.3 we consider the effect of combining functions in multidimensional linear cryptanalysis. We give multidimensional versions of the Piling Up lemma and the Correlation Theorem.

8.1 PROPERTIES OF SOME BOOLEAN FUNCTIONS UNDER MULTIDIMENSIONAL LINEAR APPROXIMATION

The results of Chapter 7 show that the data complexity of the multidimensional attack is inversely proportional to the capacity of the linear approximation. In this section, we determine the capacities for some Boolean functions that have high resistance against one-dimensional linear cryptanalysis.

In one dimension, bent function offer optimal resistance against linear cryptanalysis, that is, all the non-trivial linear approximations of bent functions have the same correlations in absolute value. We use the multi-Walsh transform in **I** to define resistance against multidimensional linear cryptanalysis. We define multi-bent functions and prove the following result:

Theorem 8.1. *The capacity of a multi-bent Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ satisfies*

$$C(f) = 2^{m-n} - 2^{-n}. \tag{8.1}$$

On the other hand, (8.1) is the smallest capacity that holds for all linear approximations $Wf \oplus U$ of f. In **I** we also show that f is multi-bent if and only if it is bent. Therefore, bent functions are optimal against multidimensional linear cryptanalysis. However, it remains an open question if there are other functions than bent functions that are optimal against multidimensional linear cryptanalysis.

Consider a multidimensional power function $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ of the form $f(x) = x^d$, where $d \geq 1$. Our goal is to determine the capacity of the linear approximation of the form $Wf(x) \oplus Ux$. To simplify the derivations, we assume that W is invertible and $Ux = ux$, for some $u \in \mathbb{F}_2^n$. Note that ux denotes multiplication in $\mathbb{F}_2^n$. We obtain the following result in **II**:

Theorem 8.2. *If* $\gcd(d, 2^n - 1) = \gcd(d - 1, 2^n - 1) = 1$ *then the capacity* $C(x^d \oplus ux) = 1$, *if* $u \neq 0$ *and* $C(x^d \oplus ux) = 0$, *if* $u = 0$.

Consider two special cases, with $d = 3$ and $d = -1$, of non-bent functions that are highly non-linear and offer strong resistance against one-dimensional linear cryptanalysis. We have the following corollary:

Corollary 8.3. *Let* $f(x) = x^d$ *in* $\mathbb{F}_2^n$, *where* $d = -1$ *and* n *is arbitrary, or* $d = 3$ *and* n *is odd. Then capacity* $C(x^d \oplus ux) = 1$, *if* $u \neq 0$ *and* $C(x^d \oplus ux) = 0$, *if* $u = 0$.

Hence, increasing dimension n will make x^{-1} more resistant against one-dimensional linear attacks. If one uses n approximations the capacity does not depend on n and the resistance against multidimensional linear cryptanalysis is the same for all $n > 1$.

8.2 EXAMPLES OF KEY STREAM GENERATORS

We consider the k.s.g. represented in Section 6.1.4, with binary coefficient b_i for the LFSR recursion and filter function f. The goal is to determine the capacity $C(W)$ of the approximation (7.3).

The first example is a filter function that is based on the AES S-box [34]. The non-linearity of the AES is obtained by using the function $Ax^{-1} + b$, where A and b are some constant matrix and vector, respectively. It suffices to consider the function x^{-1}, since the linear transformations do not affect the statistical properties of the S-box. Recall that J is the set of indices corresponding to the non-zero coefficients in the LFSR recursion equation. For $|J| = 3$ we have the following result in **II**:

Theorem 8.4. *Let* $|J| = 3$ *and* $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ *be the filter function of the k.s.g. described above obtained from the function* x^{-1} *in* $\mathbb{F}_2^n$ *by truncating its output to* m *bits. Then the correlations* $c(w)$ *in* (6.8) *are the same for all* $w \neq 0$. *Moreover, for any invertible* $m \times m$ *output mask* W, *the capacity of the multidimensional approximation is*

$$C(W) = \begin{cases} (2^m - 1)2^{4-2n}, & \textit{if } n \textit{ even} \\ (2^m - 1)2^{2-2n}, & \textit{if } n \textit{ odd.} \end{cases}$$

Now any n linearly independent one-dimensional masks can be chosen to be the base masks and $W \cdot \bigoplus_{j \in J} z_{t+j}$ is an optimal multidimensional linear approximation, i.e., it has the largest possible capacity. Note also that since the correlations $c(w)$ are the same for all $w \neq 0$, then by Theorem 3.3 the base approximations cannot be statistically independent.

For another filter function x^3 we have a similar result for $|J| = 3$:

Theorem 8.5. *Let* $|J| = 3$ *and let* $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ *be the filter function of the k.s.g. described above obtained from function* x^3 *in* $\mathbb{F}_2^n$ *by truncating its output to* m *bits. Then the correlations* $c(w)$ *in* (6.8) *are the same for all* $w \neq 0$ *and the capacity is* $C(W) = (2^m - 1)2^{2-2n}$ *for any invertible mask* W.

Finally, we consider the bent functions. The result applies for any $|J| \geq 3$:

Theorem 8.6. *Let $f(x)$ be a bent filter function of the k.s.g. described above. Then, for any fixed even $|J| \geq 4$, the correlations $c(w)$ in (6.8) are the same, for all $w \neq 0$, and for any fixed odd $|J| \geq 3$ the absolute values of the correlations $|c(w)|$ are the same, for all $w \neq 0$. The capacity is $C(W) = \sum_{a\neq 0} 2^{-2n} = (2^m - 1)2^{-2n}$ for any invertible mask W.*

We note that in all the cases, if $n = 2m$, the data complexity of a multidimensional attack is 2^{3m}, whereas for a one-dimensional attack it is 2^{4m}.

8.3 COMBINING APPROXIMATIONS

We know that in one-dimension, if consecutive functions used in the cipher are s.i., their combined correlation is given by (6.3) and the correlation decreases when more functions are used. Now we want to know how combining non-linear functions affects the resistance of a cipher against multidimensional linear cryptanalysis. In other words, we consider a “road” through the cipher instead of a one-dimensional trail.

By Lemma 3.10 we have the following result about combining multidimensional approximations:

Lemma 8.7 (Piling Up Lemma in Multiple Dimensions). *Let $f : \mathbb{F}_2^{n'} \mapsto \mathbb{F}_2^k$ and $g : \mathbb{F}_2^k \mapsto \mathbb{F}_2^n$ be consecutive but s.i. Boolean functions, where the output of f is used as an input to g. Let p and q be the p.d.'s of the linear approximations of f and g, respectively, such that the output mask of f is the input mask of g. Then the p.d. of the approximation of $g \circ f$ is the convolution $p * q$.*

In one-dimension we have an equality between the combined correlation over f and g and the product of the partial correlations. Hence, the same holds for the absolute values of the correlations and we know that the power of the linear distinguisher is reduced for each round, provided that the functions are properly chosen. As we noted in Section 3.3.4, we do not have such a strong result in multiple dimensions. Unfortunately, we only have an upper bound for the combined capacity (3.22), which gives only a *lower bound* for the data complexity.

In **II** we are also interested in generalising the Correlation theorem 6.1 to multiple dimensions. The proof uses the multi-Walsh transform. Actually, the need for a tool for studying the combination of multidimensional p.d.'s was the original motivation for multi-Walsh transform. We have the following result in **II**:

Theorem 8.8 (Correlation Theorem in Multiple Dimensions). *Let $f : \mathbb{F}_2^l \to \mathbb{F}_2^n$, $g : \mathbb{F}_2^n \to \mathbb{F}_2^k$. Let U, V and W be binary matrices of size $m \times l$, $m \times n$ and $m \times k$, respectively. Let the p.d.'s of the m-dimensional approximations $Vf(x) \oplus U(x)$, $x \in \mathbb{F}_2^l$ and $Wg(y) + V(y)$, $y \in \mathbb{F}_2^n$ be $p_f(U;V)$ and $p_g(V;W)$, respectively and denote the p.d. of $W(g \circ f)(x) \oplus Ux$ by $q(U;W)$. For all matrices U and W*

$$q(U;W) = 2^{-mn+n} \sum_{V \in (\mathbb{F}_2^n)^m} p_f(U;V) * p_g(V;W) - (2^n - 1)\theta, \qquad (8.2)$$

where θ is the uniform distribution with 2^m components.

Similarly as in one-dimension in (6.5), we have just an upper bound for the combined capacity. Therefore, we only have a lower bound for the data complexity and we do not know how the combining of functions affects the resistance against multidimensional linear cryptanalysis. An open question is to find a non-trivial lower bound for the combined correlations either for the multidimensional Piling Up lemma 8.7 or Correlation Theorem 8.8.

In the previous examples we saw how the efficiency of linear cryptanalysis increases with the number of approximations. If the one-dimensional approximation through a cipher is a trail, then the multidimensional approximation over the whole block-size of the cipher is a "highway". In these theoretical examples the capacity through a highway does not even depend on the block-size. However, in practice we cannot compute the p.d.'s for the highway. Even computing directly through roads of smaller dimension using Lemma 8.7 or Theorem 8.8 is usually infeasible. Instead, we determine the p.d. through a road by finding several strong one-dimensional approximations and combining them using Lemma 3.8.

9 CONCLUSIONS

9.1 RESULTS

In this thesis we studied multidimensional linear cryptanalysis and its application to symmetric cryptography. We showed how multiple linear approximations can be used for making linear cryptanalysis attacks more efficient. We studied different ways of realising the attacks and compared them using well-known statistical tools. We did also practical experiments, mainly with the block cipher Serpent.

In the introduction, Chapter 1, we introduced the basic problems when using multiple approximations. We described the solutions provided by the publications **I**– **VII** and also the contribution of the author of this thesis. Chapters 2– 6 were mostly a repetition of known theory about ciphers, statistics and related concepts, essential for this thesis.

Most of the new results were described in Chapters 7 and 8. The multidimensional linear distinguisher was already proposed by Baignères, et al., in [1] but there was no practical way of using the distinguisher until we proposed in **II** to use one-dimensional approximations for constructing the multidimensional distribution.

Baignères, et al., showed how the data complexity of the distinguishing attack is determined by the deviation of the p.d. from the uniform distribution. Following Biryukov, et al., we called this measure the capacity and generalised it to non-uniform distributions. Then we were able to obtain simple formulas for the data complexities for other than distinguishing attacks. We could derive theoretical bounds for multidimensional linear cryptanalysis as discussed in Chapter 8.

We considered a straightforward generalisation of the multidimensional method for the LFSR initial state recovery attack presented by Berbain, et al., in [4]. However, we noticed that this is not feasible. Rather, we showed in **VI** how multiple approximations can be used in a one-dimensional attack.

We discovered that while in one-dimension there is essentially only one way of realising Matsui's Alg. 1 or Alg. 2, both algorithms have several generalisations in multiple dimensions. We discussed the algorithms in Chapter 7 using statistical methods and analysing the empirical results with reduced round Serpent, presented in **III**, **V**, **IV** and **VII**. In Section 7.4 we explained the different results between the theories presented in **III**, **V** and **VII** about multidimensional Alg. 1. We showed the proper way of interpreting the problem using statistics.

We showed that if we know the p.d. of the multidimensional approximation, the theoretically optimal way for realising both methods is to use the LLR-statistic. However, the convolution method, presented in **VII**, is in practise the most efficient method, if the p.d. is given. On the other hand, if the p.d. is unknown, we should use the χ^2-method. In fact, Cho studied in [9] a situation where the p.d. varies significantly with the key and the best option is to use the multidimensional χ^2-method.

9.2 FUTURE WORK

While most of our experiments were consistent with the theory, we noticed that the experimental results were better than expected. For Alg. 1, we proposed using key ranking in **V**, although Alg. 1 is not the same statistical problem as Alg. 2. This "abuse" of key ranking theory can explain part of the results for Alg. 1, but not for Alg. 2.

We suggest that there is an underlying property in the cipher that makes the attacks more efficient than expected. For example, the basic assumption used in the linear cryptanalysis is that the correlation or p.d. of the linear approximation is independent of the key. If this is not the case, there may be keys for which the capacity of the approximation is significantly larger than the average capacity over all keys, hence making the data complexity smaller than expected. This could explain the results at least for Alg. 2. We propose studying more closely the dependence of the capacity (or correlation) of the used key and how this supposed dependence affects linear cryptanalysis.

We suggested using convolution method for Alg. 2 in Section 7.5.6 of this thesis. Practical experiments should also be performed.

The theoretical results in Chapter 8 leave some open questions. In one-dimensional linear cryptanalysis bent functions—and only them—are optimal. While we showed that multi-bent functions are optimal against multidimensional linear cryptanalysis, we do not know whether there are other optimal functions. It would also be interesting to find a non-trivial lower bound for the joint capacity of two multidimensional approximations. However, we assume that in a general case, such an inequality does not exist.

Finally, some ciphers may have properties that make them particularly vulnerable to multidimensional linear cryptanalysis. For example, symmetry in the cipher's structure can make it easier to find many strong linear approximations. Hence, there are many ciphers that can be attacked with multidimensional linear cryptanalysis.

BIBLIOGRAPHY

[1] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT '04, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5–9, 2004. Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450, Berlin/Heidelberg, 2004. Springer.

[2] Thomas Baignères and Serge Vaudenay. The Complexity of Distinguishing Distributions (Invited Talk). In Reihaneh Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10–13, 2008. Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 210–222, Berlin/Heidelberg, 2008. Springer.

[3] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, L. Goubin, A. Gouget, L. Granboulan, C. Lauradoux, M. Minier, T. Pornin, and H. Sibert. SOSEMANUK, a fast software-oriented stream cipher. eSTREAM, ECRYPT Stream Cipher Project, Report 2005/027, 2005. http://www.ecrypt.eu.org/stream/sosemanukpf.html.

[4] Côme Berbain, Henri Gilbert, and Alexander Maximov. Cryptanalysis of Grain. In M.J.B. Robshaw, editor, *Fast Software Encryption, 3th International Workshop, FSE 2006, Graz, Austria, March 15–17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 15–29, Berlin/Heidelberg, 2006. Springer.

[5] Eli Biham, Ross Anderson, and Lars Knudsen. Serpent: A New Block Cipher Proposal. In Serge Vaudenay, editor, *Fast Software Encryption*, volume 1372 of *Lecture Notes in Computer Science*, pages 222–238, Berlin/Heidelberg, 1998. Springer.

[6] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. On Multiple Linear Approximations. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO '04, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 2004. Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 1–22, Berlin/Heidelberg, 2004. Springer.

[7] Martin Boesgaard, Mette Vesterager, Thomas Pedersen, Jesper Christiansen, and Ove Scavenius. Rabbit: A New High-Performance Stream Cipher. In Thomas Johansson, editor, *10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 307–329, Berlin/Heidelberg, 2003. Springer.

[8] Jr. Burton S. Kaliski and M. J. B. Robshaw. Linear Cryptanalysis Using Multiple Approximations. In Yvo G. Desmedt, editor, *Advances*

in Cryptology – CRYPTO '94, 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings, volume 839 of *Lecture Notes in Computer Science*, pages 26–39, Berlin/Heidelberg, 1994. Springer.

[9] Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. In *In Topics in Cryptology – CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1–5, 2010.*, Lecture Notes in Computer Science, Berlin/Heidelberg. Springer. To appear.

[10] Ronald Christenssen. Testing Fisher, Neyman, Pearson and Bayes. *The American Statistician*, 59(2):121–126, May 2005.

[11] B. Collard, F.-X. Standaert, and J.-J. Quisquater. Experiments on the Multiple Linear Cryptanalysis of Reduced Round Serpent. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10–13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 382–397. Springer, 2008.

[12] Baudoin Collard, F. X. Standaert, and Jean-Jacques Quisquater. Improving the Time Complexity of Matsui's Linear Cryptanalysis. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology – ICISC 2007, 10th International Conference, Seoul, Korea, November 29–30, 2007.*, volume 4717 of *Lecture Notes in Computer Science*, pages 77–88, Berlin/Heidelberg, 2007. Springer.

[13] Nicolas T. Courtois and Willi Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359, Berlin/Heidelberg, 2003. Springer.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*, chapter 11. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2nd edition, 2006.

[15] H. Cramèr and H. Wold. Some theorems on distribution functions. *J. London Math. Soc.*, s1-11(4):290–295, Oct 1936.

[16] Harald Cramér. *Mathematical Methods of Statistics*. Princeton Mathematical Series. Princeton University Press, 7 edition, 1957.

[17] Noel Cressie and Timothy R. C. Read. Multinomial Goodness-of-Fit Tests. *Journal of the Royal Statistical Society. Series B*, 46(3):440–464, 1984.

[18] H. A. David. *Order Statistics*. A Wiley Publication in Applied Statistics. John Wiley & Sons, Inc., 1 edition, 1970.

[19] F.C. Drost, W.C.M. Kallenberg, D.S.Moore, and J.Oosterhoff. Power Approximations to Multinomial Tests of Fit. *Journal of the American Statistican Association*, 84(405):130–141, Mar 1989.

[20] Patrik Ekdahl and Thomas Johansson. A New Version of the Stream Cipher SNOW. In Kaisa Nyberg and Howard Heys, editors, *9th Annual International Workshop, SAC 2002 St. John's, Newfoundland, Canada, August 15–16, 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 47–61, Berlin/Heidelberg, 2003. Springer.

[21] H. Englund and A. Maximov. Attack the Dragon. In Subhamoy Maitra and C.E. Veni Madhavan, editors, *Progress in Cryptology – INDOCRYPT '05, 6th International Conference on Cryptology in India, Bangalore, India, December 10–12, 2005. Proceedings*, volume 3797 of *Lecture Notes in Computer Science*, pages 130–142, Berlin/Heidelberg, 2005. Springer.

[22] B. Gérard and J.P. Tillich. On linear cryptanalysis with many linear approximations, 2009.

[23] Jean Dickinson Gibbons. *Nonparametric Statistical Inference*. Statistics: Textbooks and Monographs. Marcel Decker, Inc., 2nd edition, 1985.

[24] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. *New Stream Cipher Designs*, volume 4986 of *Lecture Notes in Computer Science*, chapter The Grain Family of Stream Ciphers, pages 179–190. Springer, Berlin/Heidelberg, 2008.

[25] Thomas Johansson and Alexander Maximov. A Linear Distinguishing Attack on Scream. *IEEE Transactions on Information Theory*, 53(9):3127 – 3144, 2007. Previously appeared in ISIT 2003. Yokohama, Japan, June 29 – July 4,2003.

[26] P. Junod and S. Vaudenay. Optimal Key Ranking Procedures in a Statistical Cryptanalysis. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24–26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 235–246, Berlin/Heidelberg, 2003. Springer.

[27] Jung-Keun Lee, Dong Hoon Lee, and Sangwoo Park. Cryptanalysis of Sosemanuk and SNOW 2.0 Using Linear Masks. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7–11, 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 524–538, Berlin/Heidelberg, 2008. Springer.

[28] E.L. Lehmann. *Testing Statistical Hypotheses*. Springer Texts in Statistics. Springer, Berlin/Heidelberg, 2nd edition, 1997.

[29] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their application*. Cambridge University Press, 1994. Revised Edition.

[30] Mitsuru Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. In Yvo G. Desmedt, editor, *Advances in Cryptology – CRYPTO '94, 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pages 1–11, Berlin/Heidelberg, 1994. Springer.

[31] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In Tor Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 2327, 1993 Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397, Berlin/Heidelberg, 1994. Springer.

[32] Robert N. McDonough and Anthony D. Whalen. *Detection of Signals in Noise*, chapter 5. Academic Press, 2nd edition, 1995.

[33] S. Murphy. The Independence of Linear Approximations in Symmetric Cryptology. *IEEE Transactions on Information Theory*, 52(12):5510–5518, Dec 2006.

[34] NIST. A request for Candidate Algorithm Nominations for the Advanced Encryption Standard AES. http://csrc.nist.gov/archive/aes/index2.html, 1997.

[35] Kaisa Nyberg. Linear Approximation of Block Ciphers. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 439–444, Berlin/Heidelberg, 1995. Springer.

[36] Kaisa Nyberg. Correlation theorems in cryptanalysis. *Discrete Applied Mathematics*, 111(1–2):177–188, July 2001.

[37] Kaisa Nyberg and Johan Wallén. Improved Linear Distinguishers for SNOW 2.0. In Matthew Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15–17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 144–162, Berlin/Heidelberg, 2006. Springer.

[38] Greg Rose and Philip Hawkes. On the Applicability of Distinguishing Attacks Against Stream Ciphers, 2002.

[39] O. S. Rothaus. On "bent" functions. *Journal of Combinatorial Theory, Series A*, 20(3):300–305, May 1976.

[40] Ali Aydin Selçuk. On Probability of Success in Linear and Differential Cryptanalysis. *Journal of Cryptology*, 21(1):131–147, January 2008.

[41] C.E. Shannon. Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, 28:656–715, Oct 1949.

[42] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, 30(5):776–780, 1984.

[43] Serge Vaudenay. An Experiment on DES Statistical Cryptanalysis. In *CCS '96: Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pages 139–147, New York, NY, USA, 1996. ACM.

[44] Serge Vaudenay. *A Classical Introduction to Cryptography*. Springer-Verlag, Berlin/Heidelberg, 2006.

[45] Samuel S. Wilks. *Mathematical Statistics*. A Wiley Publication in Mathematical Statistics. John Wiley & Sons, Inc., New York, 1962.

TKK DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

ISBN 978-952-60-3189-7 (Print)
ISBN 978-952-60-3190-3 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)