Department of Information and Computer Science

Results on Linear Models in Cryptography

Risto M. Hakala

Aalto University

DOCTORAL
DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 29/2013

Results on Linear Models in Cryptography

Risto M. Hakala

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall AS1 of the school on 8 March 2013 at 12 noon.

Aalto University
School of Science
Department of Information and Computer Science

Supervising professor
Prof. Kaisa Nyberg

Thesis advisor
Prof. Kaisa Nyberg

Preliminary examiners
Dr. Pascale Charpin, INRIA, France
Prof. Willi Meier, University of Applied Sciences and Arts
Northwestern Switzerland, Switzerland

Opponents
Prof. Willi Meier, University of Applied Sciences and Arts
Northwestern Switzerland, Switzerland
Prof. Gregor Leander, Technical University of Denmark, Denmark

Aalto University publication series
DOCTORAL DISSERTATIONS 29/2013

ISBN 978-952-60-5028-7 (printed)
ISBN 978-952-60-5027-0 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5027-0

Unigrafia Oy
Helsinki 2013

Finland

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Risto M. Hakala

Name of the doctoral dissertation
Results on Linear Models in Cryptography

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 29/2013

Field of research Theoretical Computer Science

Manuscript submitted 12 December 2012 **Date of the defence** 8 March 2013

Permission to publish granted (date) 29 January 2013 **Language** English

□ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Many cryptanalytic techniques are based on exploiting linearity properties of cryptosystems. One of such techniques is linear cryptanalysis, invented by Matsui in 1993. Originally developed for block ciphers FEAL and DES, it has become a standard method for analyzing all kinds of symmetric ciphers. Linear cryptanalysis of a block cipher is traditionally based on a biased linear combination of the input and output bits of the cipher. Mathematically speaking, such a combination can be seen as a linear mapping to a one-dimensional binary vector space. Several authors have considered the use of other types of linear mappings as well, such as multidimensional and nonbinary mappings. To find suitable mappings, one usually has to analyze linearity properties of the individual components used in the cipher. The more the components resemble linear functions, the less secure the cipher is against linear cryptanalysis.

Linear cryptanalysis is a method for analyzing the formal description of a cryptographic primitive. Side-channel attacks form another class of cryptanalytic methods in which an implementation of the primitive is analyzed instead of the description. They are based on doing physical measurements which may reveal critical information about the internal state of the primitive.

This dissertation presents several cryptanalytic results related to linearity of cryptographic primitives. The work contains results concerning both formal specifications and real-life implementations of primitives. Related to the former area of cryptography, we describe a framework for estimating resistance against general linear cryptanalysis in which linear mappings over arbitrary finite Abelian groups can be used. As applications, we present a linear distinguishing attack on the stream cipher Shannon and on the block cipher DEAN. In addition, we study individual cryptographic components and present results regarding their linearity properties in different domains. In particular, we give evidence that certain functions based on discrete logarithm are highly nonlinear. Related to the implementation side of cryptography, we present a technique for automated analysis of side-channel data and show that it works in practice by using it to attack the ECDSA implementation in OpenSSL. The technique is based on modeling the implementation as a linear dynamical system which allows efficient analysis of the situation.

Keywords cryptography, linear cryptanalysis, distinguishing attack, side-channel analysis, linearity, nonlinearity

ISBN (printed) 978-952-60-5028-7 **ISBN (pdf)** 978-952-60-5027-0

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2013

Pages 171 **urn** http://urn.fi/URN:ISBN:978-952-60-5027-0

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Risto M. Hakala

Väitöskirjan nimi
Lineaaristen mallien tuloksia kryptografiassa

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietojenkäsittelytieteen laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 29/2013

Tutkimusala Tietojenkäsittelyteoria

Käsikirjoituksen pvm 12.12.2012 **Väitöspäivä** 08.03.2013

Julkaisuluvan myöntämispäivä 29.01.2013 **Kieli** Englanti

□ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Useat kryptoanalyyttiset menetelmät perustuvat salausmenetelmien lineaarisuusominaisuuksien hyödyntämiseen. Eräs tällainen menetelmä on Matsuin vuonna 1993 esittämä lineaarinen kryptoanalyysi. Se kehitettiin alun perin FEAL- ja DES-lohkosalausmenetelmille, mutta siitä on tullut standardi menetelmä kaikentyyppisten symmetristen salausmenetelmien analysointiin. Lohkosalausmenetelmän perinteinen lineaarinen kryptoanalyysi perustuu jonkun syöte- ja tulostebittien lineaariyhdistelyn tilastolliseen vinoumaan. Matemaattisesti tällainen yhdistely voidaan nähdä lineaarisena kuvauksena yksiulotteiselle binääriselle vektoriavaruudelle. Useat tutkijat ovat tarkastelleet myös muunlaisten lineaarikuvausten käyttämistä, kuten moniulotteisia ja ei-binäärisia kuvauksia. Sopivien kuvausten löytämiseksi täytyy tavallisesti analysoida salausmenetelmässä käytettyjen yksittäisten komponenttien lineaarisuusominaisuuksia. Mitä enemmän komponentit muistuttavat lineaarisia funktioita sitä turvattomampi salausmenetelmä on lineaarista kryptoanalyysia vastaan.

Lineaarinen kryptoanalyysi on menetelmä, jolla analysoidaan kryptografisen primitiivin muodollista kuvausta. Toisen tyyppisen menetelmäluokan muodostavat sivukanavahyökkäykset, joilla muodollisen kuvauksen asemesta analysoidaan primitiivin toteutusta. Ne perustuvat fysikaalisiin mittauksiin, jotka voivat paljastaa kriittistä tietoa primitiivin sisäisestä tilasta.

Tässä väitöskirjassa esitetään useita kryptografisten primitiivien lineaarisuuteen liittyviä kryptoanalyyttisia tuloksia. Työ sisältää tuloksia sekä primitiivien muodollisista kuvauksista että niiden todellisista toteutuksista. Edelliseen kryptoanalyysin alueeseen liittyen esitetään viitekehys vastustuskyvyn arvioimiseksi lineaarista kryptoanalyysia vastaan tilanteessa, jossa käytetään lineaarisia kuvauksia mielivaltaisissa äärellisissä Abelin ryhmissä. Sovelluksena esitetään lineaarisia erotteluhyökkäyksiä Shannon-jonosalausmenetelmää ja DEAN-lohkosalausmenetelmää vastaan. Sen lisäksi tutkitaan erillisiä salausteknisiä komponentteja ja esitetään niiden lineaarisuusominaisuuksia koskevia tuloksia erilaisissa määrittelyjoukoissa. Erityisesti esitetään tiettyjen diskreettiin logaritmiin perustuvien funktioiden epälineaarisuutta tukevia tuloksia. Kryptografisten toteutusten puolelta esitetään tekniikka sivukanavadatan automaattiseksi analysoimiseksi ja osoitetaan että se toimii käytännössä hyökkäyksessä OpenSSL-järjestelmän ECDSA-toteutusta vastaan. Tekniikka perustuu toteutuksen mallintamiseen lineaarisena dynaamisena järjestelmänä, joka mahdollistaa tilanteen tehokkaan analyysin.

Avainsanat kryptografia, lineaarinen kryptoanalyysi, erotteluhyökkäys, sivukanava-analyysi, lineaarisuus, epälineaarisuus

ISBN (painettu) 978-952-60-5028-7 **ISBN (pdf)** 978-952-60-5027-0

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2013

Sivumäärä 171 **urn** http://urn.fi/URN:ISBN:978-952-60-5027-0

Contents

Preface

This thesis is the result of the research I have conducted in the Department of Information and Computer Science at Aalto University School of Science. This research has been funded by Helsinki Doctoral Programme in Computer Science - Advanced Computing and Intelligent Systems, Academy of Finland under project #122736, and partly by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II. The work has also been supported by personal grants kindly awarded by the Nokia Foundation and the KAUTE Foundation.

I am deeply indebted to my supervisor Prof. Kaisa Nyberg. She has been absolutely invaluable for suggesting me research topics to work on and helping me in research throughout the years. I am also grateful to all my coauthors whom I have had the pleasure to collaborate with. I would in particular like to thank Dr. Billy Bob Brumley for asking me to collaborate on an interesting topic that was outside my main research area. I also thank my colleagues for creating a positive working atmosphere in the department.

I thank the pre-examiners Prof. Willi Meier (University of Applied Sciences and Arts Northwestern Switzerland) and Dr. Pascale Charpin (INRIA, France) for taking their time to review this thesis. I am grateful to Prof. Gregor Leander (Technical University of Denmark) and Prof. Willi Meier for the honor of having them as opponents.

Last but not least, I wish to thank my family and friends for all the support and encouragement that I have received.

Espoo, February 4, 2013,

Risto M. Hakala

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Risto M. Hakala and Atle Kivelä and Kaisa Nyberg. Estimating Resistance against Multidimensional Linear Attacks: An Application on DEAN. Accepted for publication in *Inscrypt 2012*, 17 Pages, December 2012.

II Risto M. Hakala. An upper bound for the linearity of Exponential Welch Costas functions. *Finite Fields and Their Applications*, Volume 18, Issue 4, Pages 855–862, July 2012.

III Risto M. Hakala and Kaisa Nyberg. On the Nonlinearity of Discrete Logarithm in $\mathbb{F}_{2^n}$. In *SETA 2012*, Volume 6338 of Lecture Notes in Computer Science, Pages 333–345, September 2010.

IV Zahra Ahmadian and Javad Mohajeri and Mahmoud Salmasizadeh and Risto M. Hakala and Kaisa Nyberg. A practical distinguisher for the Shannon cipher. *Journal of Systems and Software*, Volume 83, Issue 4, Pages 543–547, April 2010.

V Billy Bob Brumley and Risto M. Hakala. Cache-Timing Template Attacks. In *ASIACRYPT 2009*, Volume 5912 of Lecture Notes in Computer Science, Pages 667–684, December 2009.

VI Billy Bob Brumley and Risto M. Hakala and Kaisa Nyberg and Sampo Sovio. Consecutive S-box Lookups: A Timing Attack on SNOW 3G. In *ICICS 2010*, Volume 6476 of Lecture Notes in Computer Science, Pages 171–185, December 2010.

Author's Contribution

Publication I: "Estimating Resistance against Multidimensional Linear Attacks: An Application on DEAN"

The current author is responsible for proposing the presented generalizations related to linear cryptanalysis, proposing and implementing the practical experiments for the distinguishers, and the related writing.

Publication II: "An upper bound for the linearity of Exponential Welch Costas functions"

The current author is the sole contributor.

Publication III: "On the Nonlinearity of Discrete Logarithm in $\mathbb{F}_{2^n}$"

The current author is responsible for studying character sums related to the nonlinearity of discrete logarithm, implementing the experiments, and the related writing.

Publication IV: "A practical distinguisher for the Shannon cipher"

The current author is responsible for proposing techniques to obtain an accurate estimate for the attack complexity, implementing the linear distinguishing attack in practice, making the experiments, and the related writing.

Publication V: "Cache-Timing Template Attacks"

The current author is responsible for proposing improvements to the presented automated side-channel data analysis method, and the related writing.

Publication VI: "Consecutive S-box Lookups: A Timing Attack on SNOW 3G"

The current author is responsible for proposing improvements to the state recovery algorithm, partially implementing it, and the related writing.

1. Introduction

A mathematical system becomes generally easier to understand if it exhibits linear rather than nonlinear behavior. Indeed, many cryptanalytic techniques are based on exploiting linearity properties of cryptosystems. This dissertation presents cryptanalytic results related to such techniques.

One cryptanalytic method that exploits linear behavior is linear cryptanalysis [34, 35]. Originally developed for block ciphers FEAL and DES, it has become one of the most widely used methods to formally analyze symmetric ciphers. Nowadays, resistance against linear cryptanalysis is held as a basic design principle when new ciphers are designed and many methods to ensure that no linear attacks are possible have been developed. Linear cryptanalysis of a block cipher is traditionally based on a biased linear combination of the input and output bits of the cipher. Mathematically speaking, such a combination can be seen as a linear mapping to a one-dimensional binary vector space. Several authors have considered the use of other types of linear mappings as well, such as multidimensional and nonbinary mappings. To find suitable mappings, one usually has to analyze linearity properties of the individual components used in the cipher. The more the components resemble linear functions, the less secure the cipher is against linear cryptanalysis. Some basic problems in linear analysis are finding good linear mappings, estimating the data complexity of the attack, and finding the right statistical hypothesis test to conduct the attack.

Traditional linear cryptanalysis is a technique to analyze the formal description of the system. Side-channel attacks (e.g., [22]) form another class of cryptanalytic methods in which a specific implementation of the system is analyzed instead of the specification of it. They are based on information obtained by doing physical measurements on the system. Side-channel attacks can be based on, e.g., power consumption, electromagnetic radiation, acoustics, or timings. The measurements may reveal the internal state of the system and can be used in conjunction with other cryptanalytic techniques to break the system. Side-channel analysis can be done for implementations of any cryptographic primitive. For software implementations, attacks based on cache timings are a real threat and an implementation can be vulnerable to such attacks if it uses table lookups involving state or secret key bits. Even if a vulnerability is found, implementing an attack in practice can still be quite complicated. For example, the attack may require analysis of large amounts of side-channel data which is infeasible to do by hand.

In this thesis, we present contributions concerning both formal specifications and real-life implementations of cryptographic primitives. Related to formal cryptanalysis, we examine symmetric ciphers and their building blocks using linear cryptanalysis. Related to implementation-specific analysis, we present a framework for automated side-channel data analysis. Real primitive designs are studied in both fields of cryptanalysis. The next section gives a more detailed overview of the contributions based on the six publications included in the thesis.

1.1 Contributions of the Thesis

The main contributions of the thesis are in the six publications that can be summarized as follows.

Publication I. The work gives a technique for estimating average capacities for multidimensional linear approximation for any iterated block cipher. The technique combines linear hulls of different linear approximations used in multidimensional analysis. Thus, it provides a more accurate estimate for the capacity than if a single linear trail was used. The technique computes capacity estimates in an iterative manner which makes it more efficient. Both binary and nonbinary linear approximations can be used with the technique. It is used to study linear approximations of the toy cipher DEAN which is a nonbinary SPN network. The obtained results show that the security bounds given by the designers of DEAN are not sufficient for multidimensional attacks.

Publication II. Linearity is a quantity that indicates the maximum correlation of a function and affine functions. The work derives an upper bound for the linearity of Exponential Welch Costas functions. They are nonbinary functions that can be utilized, e.g., as S-boxes in ciphers. The derived upper bound shows that Exponential Welch Costas functions are asymptotically more nonlinear than previously conjectured by Drakakis et al. [12]. It also shows that their asymptotic nonlinearity is high: their linearity is larger by at most a logarithmic factor than the minimum linearity achieved by generalized bent functions. Thus, Exponential Welch Costas functions provide asymptotically good resistance to linear cryptanalysis.

Publication III. The work studies linearity properties of the discrete logarithm defined in a finite field with 2^n elements. The discrete logarithm function is extended to a bijection in the binary vector space which is also the domain where linearity properties are studied. The work shows how the discrete logarithm function is related to a function studied previously by Feng et al. [16]. The nonlinearity of this class of functions was studied by Carlet and Feng [6]. In this work, the approach is different than what they used and the derived bound is slightly better, while asymptotically, the bounds are equal. The work identifies a certain geometric sum which is the basis for the lower bound on the nonlinearity. The sum is studied using a certain type of polynomials, called mask polynomials, that depend on what the linear approximation being used. In addition, the sum is investigated experimentally to identify how it affects the final bound.

Publication IV. The work presents a practical linear distinguishing attack on the Shannon stream cipher that uses at most a 256-bit secret key. The attack is able to distinguish the keystream generated by Shannon from a uniformly random sequence using about 2^{31} keystream words, a single counter, and about 2^{31} computations. The specification of Shannon states that there should be no distinguishing attacks on Shannon requiring less than 2^{80} keystream words and less than 2^{128} computations.

Publication V. Cache-timing attacks are based on cache-timing data that reflects the behavior of the algorithm and reveals information that can be used to recover the state of the algorithm or some secret key bits. The work describes a novel method for automated analysis of cache-timing data, which is often very time consuming if done by hand. The method uses a hidden Markov model to describe the relationship between side channel observations and internal states of the algorithm. The hidden Markov model represents the operation of the implemented algorithm as a linear dynamical system that is observed through a noisy channel. The developed method also makes use of vector quantization for classification of timing data. It was demonstrated by running an attack on the elliptic curve portion of OpenSSL. In combination with a lattice attack, we managed to carry out a practical key recovery attack on live cache-timing data without simulating the side channel.

Publication VI. The work proposes a cache-timing attack on the SNOW 3G stream cipher. The attack exploits cache-timing data that is caused by operations in the implementation that involve table lookups. In particular, the attack exploits the table lookups involved in the two consecutive S-boxes and the two multiplication operations in the shift register update function. The information revealed by the two S-boxes makes it possible to determine the input of one of the S-boxes almost uniquely. The attack is capable of recovering the full cipher state from the empirical timing data in a matter of seconds. It requires no known keystream only observations of a small number of cipher clocks.

Other Contributions. Related to Publication III, we derive a new lower bound for the nonlinearity of a linear combination of output bits of discrete logarithm in a finite field with 2^n elements. Unlike the bound in Publication III, the new bound is dependent on the Hamming weight of the masking vector used to define the linear combination. We also give new results on the linearity properties of constant addition modulo 2^n and constant multiplication modulo $2^n + 1$, which both are operations used in IDEA.

1.2 Outline of the Thesis

The thesis begins with a summary which is followed by the publications. The structure of the summary is as follows.

Chapter 2 outlines the main cryptographic concepts relevant for this thesis. We give an overview of basic symmetric ciphers, classification of traditional cryptographic attacks, and side-channel analysis.

Chapter 3 presents some basic algebraic tools related to group theory that are used throughout the text. The main tools relevant to linear cryptanalysis are algebraic transforms that can be studied using character sums.

Chapter 4 presents the statistical tools used in the cryptanalytic techniques relevant to this thesis. These techniques rely on basic probability theory, statistical testing, hidden Markov models, and vector quantization.

Chapter 5 presents a framework for general linear distinguishing attacks that can employ linear mappings over arbitrary finite groups. The presented attacks are essentially statistical tests based on either the LLR or the χ^2 test statistic which are also discussed in the chapter. To estimate the attack complexity, we study properties of linear approximations which are the foundation of linear attacks.

Chapter 6 presents results on linear properties of some mappings that can be used as building blocks in ciphers. The results indicate the resistance of these mappings to linear cryptanalysis. Novel results on the linearity of discrete logarithm are also given.

Chapter 7 presents the method for automated analysis of side-channel data. The method relies on hidden Markov models and vector quantization.

Chapter 8 draws conclusions.

2. Cryptography

Cryptography can be divided into secret-key and public-key cryptography, also known as symmetric and asymmetric cryptography. Secret-key encryption uses the same key for encrypting and decrypting. Public-key encryption commonly uses a pair of keys, where one of them is secret and the other one public. The security of these cryptosystems can be assessed in many ways. Traditional cryptanalysis studies the formal specification of the cryptosystem. Opposed to this, side-channel analysis tries to find weaknesses in a specific implementation of the system. In the following sections, we give a very short overview of basic symmetric ciphers, classification of attacks, and side-channel analysis. The reader is referred to, e.g., Menezes et al. [36] for a more extensive overview.

2.1 Symmetric Ciphers

2.1.1 Block Ciphers

Regarding analysis of symmetric ciphers, our focus is on general techniques that can be applied for both block and stream ciphers. Given a secret key K, a block cipher on set A is defined by an encryption function

$$\mathcal{E}_K \colon A \to A.$$

For each K, the function $\mathcal{E}_K$ is required to be invertible and the inverse function $\mathcal{E}_K^{-1}$ is called the decryption function. Given $x, y \in A$ such that $\mathcal{E}_K(x) = y$ for some key K, we call x the plaintext and y the ciphertext.

Many block ciphers are constructed by iterating the same round function for a number of rounds. At each round, a round key is typically combined using some group operation, such as the bitwise XOR. The round keys are determined from the key by simple operations. The encryption function

$\mathcal{E}_K$ of an R-round iterated block cipher on A is specified by a sequence of round functions $G_{K_1}, G_{K_2}, \ldots, G_{K_R}$ on A and a key $K = (K_1, K_2, \ldots, K_R) \in A^R$ such that the encryption $\mathcal{E}_K(x)$ of plaintext $x \in A$ is computed as

$$\begin{aligned} x_0 &= x, \\ x_r &= G_{K_r}(x_{r-1}) \quad \text{for } r = 1, 2, \ldots, R, \\ \mathcal{E}_K(x) &= x_R. \end{aligned}$$

Key-alternating block ciphers are a class iterated block ciphers in which each round function G_{K_r} can be expressed with the function $g_r \colon A \to A$ as $G_{K_r}(x) = g_r(x + K_r)$.

2.1.2 Stream Ciphers

Stream ciphers are ciphers that encrypt the plaintext by combining it with a pseudorandom keystream. In this work, we consider only synchronous stream ciphers that produce a keystream independently of the ciphertext.

Synchronous stream ciphers are finite state machines containing an internal state and a state update function. In addition, they contain a keystream generating function that is used to produce the keystream, and an output function that is used to combine the keystream with the plaintext. The internal state at time t is represented by the vector $s_t = (s_{t1}, s_{t2} \ldots, s_{tl})$ of l individual components s_{ti}, $1 \leq i \leq l$. The state update function G produces the next state as

$$s_{t+1} = G(s_t, K),$$

where s_t is the current state and K is the key. The keystream function F produces a new keystream symbol z_t from the key K and the internal state s_t as

$$z_t = F(s_t, K).$$

The output function H is an injective function that produces a ciphertext symbol c_t by combining a plaintext symbol p_t and a keystream symbol z_t as

$$c_t = H(p_t, z_t).$$

The output function has to be injective so that it is possible to determine the plaintext from the ciphertext and the keystream. It is often chosen to be a simple operation such as the bitwise XOR.

2.2 Cryptanalysis

Cryptanalytic attacks can be classified according to what threat they pose to the cryptosystem. The classification is commonly done by evaluating (1) what knowledge and capabilities are needed as a prerequisite, (2) how much secret information is revealed, and (3) how much effort is required to perform the attack. In the following sections, we present a classification of formal attacks against symmetric ciphers. The same principles apply in side-channel analysis, which we also briefly discuss. In side-channel attacks, one should additionally consider implementation specific details.

2.2.1 Attack Scenarios

Cryptanalytic techniques are based on a number of assumptions about the amount of information the attacker has available. It is usually assumed that the algorithm description is known to the attacker, which is called Kerckhoff's principle. Other common attack scenarios include the following:

Ciphertext-only. The attacker has only a collection of ciphertexts.

Known-plaintext. The attacker has a set of ciphertexts and the corresponding plaintexts.

Chosen-plaintext. The attacker can choose a set of plaintexts and obtain the corresponding ciphertexts.

Adaptive chosen-plaintext. The attacker can choose subsequent plaintexts based on the previously obtained ciphertexts.

There also exist chosen-ciphertext and adaptive chosen-ciphertext attacks in which the assumptions are made for ciphertexts instead of plaintexts. In a related-key attack, the attacker can obtain ciphertexts generated with two different keys whose values are unknown, but there is a mathematical relationship between the keys that is known to attacker. There are also chosen-IV and known-IV attacks.

2.2.2 Success of the Attack

The main objective of cryptanalysis is recovery of the secret key since it allows decryption of any ciphertexts generated with the key. However, even if the attack does not lead to key recovery, it can give useful information about the security of the cipher. Knudsen [21] classified attacks on block ciphers based on the amount and quality of previously unknown information that the they reveal:

Total break. The attacker recovers the secret key.

Global deduction. The attacker finds an algorithm equivalent to encryption and decryption without learning the secret key.

Instance deduction. The attacker is able to generate previously unknown plaintexts or ciphertexts.

Information deduction. The attacker gains previously unknown Shannon information about the secret key, the plaintexts or the ciphertexts.

Distinguishing algorithm. The attacker can detect statistical anomalies in the cipher by applying the algorithm.

The presented classification is hierarchical: total break allows global deduction, global deduction allows instance deduction, and so on. It is also possible to use a distinguishing algorithm for gaining information about the secret key if, e.g., it is known how different keys affect the statistical properties of the ciphertexts. With stream ciphers, internal state recovery can be classified as instance deduction and initial state recovery as global deduction. A state recovery algorithm for a stream cipher can also lead to recovery of the initial state if the state update function G is bijective and independent of the key K.

The focus of the formal analysis in this thesis is on distinguishing attacks. The presented side-channel attacks are concerned with recovery of the secret key or the initial state.

2.2.3 Complexity of the Attack

Another characterization of attacks is based on the required computational resources:

Time. The number of computation steps that are needed to execute the attack.

Memory. The amount of storage required to perform the attack.

Data. The amount of data (e.g., plaintexts, ciphertexts, or keystream) required for the attack.

For distinguishing attacks, the data complexity of the attack is usually the most important indicator of attack complexity, but memory requirements can also be significant. A distinguishing attack against a stream cipher is commonly considered successful, if the keystream can be distinguished from a truly random sequence based on less than $|\mathcal{K}|$ keystream symbols, where $\mathcal{K}$ represents the set of possible keys K. With block ciphers, the size of the codebook can be used as the limit instead.

2.2.4 Side-Channel Analysis

Traditional cryptanalysis studies the formal description of the system. Opposed to this, side-channel attacks are based on information that is gained from the physical implementation of the system. Side-channel leakages might reveal information about the internal state of the system and may be used in conjunction with other cryptanalytic techniques to break the system. Side-channel attacks can be based on information obtained from, e.g., power consumption, timings, electromagnetic radiation or even sound. Active attacks in which the attacker manipulates the operation of the system by physical means are also considered side-channel attacks.

Due to the number of different types of side-channel attacks, it would be difficult to list different attack scenarios as with traditional cryptanalysis. In this thesis, our focus is on cache-timing attacks in which side channel information is gained by measuring cache access times when the cryptographic algorithm is running. To obtain these measurements, it is assumed that the attacker has some device or code in their possession that they can give input to, program, or modify in some way that forces it to perform in a certain manner, while at the same time obtaining measurements from the side channel.

3. Group Theory

This chapter presents some basic algebraic tools related to group theory that are used in the analysis. The main algebraic tools are based on characters and related sums.

3.1 Basic Notation

Let n be a positive integer and p be a prime. We use $\mathbb{Z}_n$ to denote the ring of integers modulo n and $\mathbb{F}_{p^n}$ to denote the finite field of order p^n. We associate every element of $\mathbb{F}_{p^n}$ to a unique vector of $\mathbb{F}_p^n$ using a fixed basis of $\mathbb{F}_{p^n}$ over $\mathbb{F}_p$. The vectors in $\mathbb{F}_2^n$ are identified with the elements in $\mathbb{Z}_{2^n}$ using the natural correspondence $(u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n \leftrightarrow u_n 2^0 + u_{n-1}2^1 + \cdots + u_1 2^{n-1} \in \mathbb{Z}_{2^n}$. We use $\oplus$ to denote the addition (also called the bitwise XOR) of vectors in $\mathbb{F}_2^n$. Given two vectors $u = (u_1, u_2, \ldots, u_n) \in \mathbb{F}_2^n$ and $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n$ we denote $u \cdot v = u_1 v_1 \oplus u_2 v_2 \oplus \cdots \oplus u_n v_n \in \mathbb{F}_2$.

Given two functions $f\colon A \to B$ and $g\colon B \to C$, their composition $g \circ f\colon A \to C$ is defined as $(g \circ f)(x) = g(f(x))$. Given a real number z, we denote $e(z) = e^{2\pi i z}$ and $e_n(z) = e(z/n)$.

3.2 Characters

Let A be a finite Abelian group written additively and U be the multiplicative group of complex numbers of absolute value one.

Definition 3.1. A character χ of A is a homomorphism $\chi\colon A \to U$.

In other words, if χ is a character of A, then

$$\chi(x + y) = \chi(x)\chi(y)$$

for all $x, y \in A$. A trivial character χ_0 is defined as $\chi_0(x) = 1$ for all $x \in A$. For every character χ of A, there is a conjugate character $\overline{\chi}$ defined by

$\overline{\chi}(x) = \overline{\chi(x)}$ for all $x \in A$, where the bar denotes complex conjugation. It follows that $\overline{\chi}(x) = \chi(-x)$. Given characters χ and ψ of two finite Abelian groups A and B, respectively, we use $\chi \times \psi$ to denote the function defined as $(\chi \times \psi)(x, y) = \chi(x)\psi(y)$, which is a character of $A \times B$.

The set of characters obviously forms an Abelian group under multiplication. In fact, there is a one-to-one correspondence between the characters of A and the elements of A [28]. Thus, we can identify every character of A as χ_u, where $u \in A$. Moreover, the identification can be done in such a way that χ_0 denotes the trivial character and $\overline{\chi_u} = \chi_{-u}$ for all $u \in A$.

3.2.1 Characters of a Finite Field

Let $\mathbb{F}_q$ denote a finite field of order $q = p^n$, where p is a prime and n is a positive integer. We call characters of the additive group of $\mathbb{F}_q$ the additive characters of $\mathbb{F}_q$. Similarly, characters of the multiplicative group $\mathbb{F}_q^*$ are called the multiplicative characters of $\mathbb{F}_q$.

Let $\mathrm{Tr}\colon \mathbb{F}_q \to \mathbb{F}_p$ denote the absolute trace function defined as

$$\mathrm{Tr}(x) = x + x^p + \cdots + x^{p^{n-1}}.$$

The trace function is a linear transformation, and therefore the function $\chi_1\colon \mathbb{F}_q \to U$ defined by

$$\chi_1(x) = e_p(\mathrm{Tr}(x))$$

is clearly an additive character of $\mathbb{F}_q$. The following theorems [28, pp. 190–191] describe how all additive and multiplicative characters of $\mathbb{F}_q$ can be defined.

Theorem 3.2. *For each $0 \le j \le q-1$, the function χ_j with $\chi_j(x) = \chi_1(jx)$ for all $x \in \mathbb{F}_q$ is an additive character of $\mathbb{F}_q$ and every additive character of $\mathbb{F}_q$ is obtained in this way.*

Theorem 3.3. *Let α be a fixed primitive element of $\mathbb{F}_q$. For each $0 \le j \le q-2$, the function ψ_j with*

$$\psi_j(\alpha^k) = e_{q-1}(jk)$$

for all $0 \le k \le q-2$ defines a multiplicative character of $\mathbb{F}_q$ and every multiplicative character of $\mathbb{F}_q$ is obtained in this way.

Alternatively, multiplicative characters of $\mathbb{F}_q$ can be defined using a discrete logarithm. The discrete logarithm $\log_a x$ of $x \in \mathbb{F}_q^*$ to the base a is the integer k such that $0 \leq k \leq q-2$ and $x = a^k$. The multiplicative characters of $\mathbb{F}_q$ can now be defined for all $x \in \mathbb{F}_q^*$ as

$$\psi_j(x) = e_{q-1}(j \log_\alpha x),$$

where $\alpha \in \mathbb{F}_q^*$ is primitive.

3.3 Character Sums

Much of the analysis in this thesis is based on studying the properties of certain character sums. Let A be a finite Abelian group written additively. The following fundamental theorem [28, pp. 188–189] yields several important identities in character sums.

Theorem 3.4. *If χ is a nontrivial character of A, then*

$$\sum_{x \in A} \chi(x) = 0.$$

If χ is the trivial character χ_0, then the above sum is clearly equal to $|A|$. It is then straightforward to prove the following result on character sums, which is often called the orthogonality relation of characters.

Theorem 3.5 (Orthogonality relation)**.** *If χ_u and χ_v are characters of A, then*

$$\frac{1}{|A|} \sum_{x \in A} \chi_u(x)\overline{\chi_v(x)} = \begin{cases} 0 & \text{if } u \neq v, \\ 1 & \text{if } u = v. \end{cases}$$

Gauss sums are an important tool for analyzing properties of mappings between additive and multiplicative groups of finite fields.

Definition 3.6. Let ψ be a multiplicative and χ an additive character of $\mathbb{F}_q$. The character sum

$$G(\psi, \chi) = \sum_{x \in \mathbb{F}_q^*} \psi(x)\chi(x).$$

is called a Gauss sum.

The exact value of a Gauss sum is generally not known. However, the absolute value of a Gauss sum can be determined as given in the following theorem [28, p. 193], which has many useful applications when one tries to obtain bounds for other character sums.

Theorem 3.7. *Let ψ be a multiplicative and χ an additive character of $\mathbb{F}_q$. Then the Gauss sum $G(\psi, \chi)$ satisfies*

$$G(\psi,\chi) = \begin{cases} q-1 & \textit{if } \psi = \psi_0,\ \chi = \chi_0, \\ -1 & \textit{if } \psi = \psi_0,\ \chi \neq \chi_0, \\ 0 & \textit{if } \psi \neq \psi_0,\ \chi = \chi_0. \end{cases}$$

If $\psi \neq \psi_0$ and $\chi \neq \chi_0$, we have

$$|G(\psi,\chi)| = \sqrt{q}.$$

3.3.1 Examples of Character Sums

The following special cases of the previous results are important for the analysis later on. For each $u \in \mathbb{Z}_m$, the function $\chi_u(x) = e_m(ux)$ defines a character on the additive group $\mathbb{Z}_m$. By Theorem 3.4, we then have

$$\sum_{x \in \mathbb{Z}_m} e_m(ux) = \begin{cases} m & \text{if } u = 0 \bmod m, \\ 0 & \text{if } u \neq 0 \bmod m, \end{cases}$$

According to the orthogonality relation for characters, we have

$$\sum_{r \in \mathbb{Z}_m} e_m(r(y-z)) = \begin{cases} m & \text{if } y = z \bmod m, \\ 0 & \text{if } y \neq z \bmod m, \end{cases}$$

for $y, z, m \in \mathbb{Z}$ with $m \geq 2$.

3.4 Fourier Transform

Suppose that A and B are finite Abelian groups (written additively). Let χ_u and ψ_v be unique characters of A and B, respectively, identified by $u \in A$ and $v \in B$.

Definition 3.8. The Fourier transform of $\phi\colon A \to \mathbb{C}$ is defined by

$$\widehat{\phi}(u) = \sum_{x \in A} \phi(x)\overline{\chi_u(x)}.$$

The inverse Fourier transform is then given by

$$\phi(x) = \frac{1}{|A|} \sum_{u \in A} \widehat{\phi}(u)\chi_u(x).$$

If A is the Cartesian product $B \times B \times \cdots \times B$ of n groups, we may denote the Fourier transform of ϕ by $\widehat{\phi}(u_1, u_2, \ldots, u_n)$, where $u_i \in B$ for all $1 \leq i \leq n$. In this case, χ_u is the character $\psi_{u_1} \times \psi_{u_2} \times \cdots \times \psi_{u_n}$.

Definition 3.9. The Fourier transform of $f\colon A \to B$ is defined by

$$\widehat{(\psi_v \circ f)}(u) = \sum_{x \in A} \psi_v(f(x))\overline{\chi_u(x)}.$$

We conclude this section by recalling Parseval's theorem [29, p. 385]. Given a function $\phi\colon A \to \mathbb{C}$, then

$$\sum_{u \in A} |\widehat{\phi}(u)|^2 = |A| \sum_{x \in A} |\phi(x)|^2. \tag{3.1}$$

4. Statistics and Probability Theory

This chapter presents the statistical tools relevant to the cryptanalytic techniques used in the thesis. These techniques rely on basic probability theory, statistical hypothesis testing, hidden Markov models, and vector quantization.

4.1 Probability Distributions

Let X and Y be random variables with finite sample spaces A and B, respectively.

Definition 4.1. The probability density function of X is the function $p_X\colon A \to [0,1]$ defined as

$$p_X(x) = \Pr(X = x).$$

If X is clear from the context, we will simply use p to denote p_X. The function p_X will also be referred to as the probability distribution of X. If $p_X(x) = |A|^{-1}$ for all $x \in A$, then p_X defines a uniform distribution and X is said to be uniformly distributed. We will use p_0 to denote the uniform distribution. The joint probability distribution of X and Y is defined as $p_{X,Y}(x,y) = \Pr(X = x, Y = y)$ for all $x \in A$ and $y \in B$. Random variables X and Y are said to be statistically independent if $p_{X,Y}(x,y) = p_X(x)p_Y(y)$ for all $x \in A$ and $y \in B$.

Let $f\colon A \to B$ be a function between finite sets A and B, and let X be a uniformly distributed random variable on A.

Definition 4.2. The output distribution of $f\colon A \to B$ is the function $p_f\colon B \to [0,1]$ defined as

$$p_f(y) = \Pr(f(X) = y).$$

Since X is uniformly distributed, we can alternatively define the output distribution of f as

$$p_f(y) = \frac{|\{x \in A \,:\, f(x) = y\}|}{|A|}.$$

4.2 Correlation

The correlation coefficient between two functions indicates similarity between the functions. Assume that A and B are groups defined as before and let U be the multiplicative group of complex numbers with absolute value one.

Definition 4.3. The correlation coefficient between functions $\phi\colon A \to U$ and $\theta\colon A \to U$ is the complex number defined as

$$c(\phi, \theta) = \frac{1}{|A|} \sum_{x \in A} \phi(x)\overline{\theta(x)}.$$

It is clear that the absolute value of a correlation coefficient lies always within the range $[-1, 1]$.

Suppose that $f\colon A \to B$ is a function and let χ_u and ψ_v be characters of A and B, respectively. The correlation coefficient between functions $\psi_v \circ f$ and χ_u is denoted by $c_f(u, v)$. It follows that

$$c_f(u, v) = c(\psi_v \circ f, \chi_u) = \frac{1}{|A|}(\widehat{\psi_v \circ f})(u).$$

We will simply use $c_f(v)$ to denote the correlation coefficient $c(\psi_v \circ f, \chi_0)$, where χ_0 is the trivial character of A.

4.2.1 Correlations and Probability Distributions

In this section, we give results regarding the values of a function and its homomorphic projection. Let A and B be defined as before and let ψ_v denote a character of B. Suppose that $f\colon A \to B$ is a function with the output distribution p_f. The following result describes a relationship between the values of f and $\psi_v \circ f$ for all $v \in B$.

Theorem 4.4. *Let $f\colon A \to B$ be defined as above. For all $v \in B$, we have*

$$c_f(v) = \widehat{p_f}(-v). \tag{4.1}$$

Proof. Using the definitions of correlation coefficient, Fourier transform, and output distribution, we get

$$\begin{aligned} c_f(v) &= \frac{1}{|A|}\sum_{x\in A}\psi_v(f(x)) = \frac{1}{|A|}\sum_{y\in B}|\{x \in A \,:\, f(x) = y\}|\psi_v(y) \\ &= \sum_{y\in B} p_f(y)\psi_v(y) = \sum_{y\in B} p_f(y)\overline{\psi_{-v}(y)} = \widehat{p_f}(-v) \end{aligned}$$

for all $v \in B$. □

Hence, if the correlation coefficients $c_f(v) = c(\psi_v \circ f, \chi_0)$ are known for all $v \in B$, then the output distribution p_f can be determined completely using the inverse Fourier transform by the formula

$$p_f(y) = \frac{1}{|B|}\sum_{v\in B} c_f(v)\psi_{-v}(y),$$

where $y \in B$. In other words, the output distribution of f is completely determined by its homomorphic projections $\psi_v \circ f$, $v \in B$. The general form of this result is known as the Cramér–Wold theorem [10]. Parseval's theorem (3.1) and the identity (4.1) yield the following corollary.

Corollary 4.5. *Let $f\colon A \to B$ be defined as above. Then*

$$\sum_{v\in B}|c_f(v)|^2 = |B|\sum_{y\in B} p_f(y)^2. \tag{4.2}$$

4.3 Capacity

The capacity of a probability distribution measures the nonuniformity of the distribution: larger capacity indicates higher nonuniformity. It is used to determine the relationship between the sample size and the error probability of a statistical hypothesis test based on the LLR statistic or the χ^2 statistic.

Definition 4.6. The capacity between probability distributions p and p' with the sample space A is

$$C(p,p') = \sum_{x\in A}\frac{(p(x) - p'(x))^2}{p'(x)}.$$

If p' is the uniform distribution, then $C(p,p')$ is denoted by $C(p)$ and called the capacity of p. The capacity of a function is defined using its output distribution in the following way.

Definition 4.7. The capacity of a function $f\colon A \to B$ with the output distribution p_f is

$$C(f) = C(p_f) = |B|\sum_{y\in B}\left(p_f(y) - \frac{1}{|B|}\right)^2.$$

4.3.1 Capacity and Correlations

According to Theorem 4.4, the distribution of $f\colon A \to B$ can be determined from the correlation coefficients $c_f(v)$, $v \in B$. The following theorem describes how the correlation coefficients can also be used to determine the capacity of f.

Theorem 4.8. *Let f be defined as before. Then*

$$C(f) = \sum_{v \neq 0} |c_f(v)|^2. \tag{4.3}$$

Proof. We get from the definition of $C(f)$ that

$$C(f) = |B| \sum_{y \in B} \left(p_f(y)^2 - 2p_f(y)\frac{1}{|B|} + \frac{1}{|B|^2} \right) = |B| \sum_{y \in B} p_f(y)^2 - 1.$$

The result follows from Parseval's theorem (3.1) when we observe that $c_f(v) = 1$ for $v = 0$. □

The corresponding theorem for mappings between binary vector spaces has been shown by Baignères et al. [1]. For a Boolean function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$, the capacity can be simply computed as

$$C(f) = |c_f(1)|^2 = \left| \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} \right|^2.$$

4.4 Statistical Testing

In this thesis, we are mainly concerned with statistical hypothesis tests that decide between two hypotheses. Given a data set with an empirical distribution q, we want to determine whether the given data has been drawn from some specific distribution or is uniformly random. These kind of hypotheses tests are relevant for statistical distinguishing attacks, where the attacker tries to distinguish whether the given data has been produced by a cipher (with a certain unknown key) or is a sample drawn from the uniform distribution. We present two statistical tests that can be used for this purpose. For a more detailed discussion about statistical testing, we refer to Cover and Thomas [9].

Let $X_1, X_2, \ldots, X_N$ be a collection of independent and identically distributed random variables with sample space A. The studied data set is formed by the realizations $x_1, x_2, \ldots, x_N$ of these random variables. Their empirical distribution q is computed by

$$q(y) = \frac{|\{1 \leq t \leq N \,:\, x_t = y\}|}{|A|}$$

for all $y \in A$.

4.4.1 Likelihood Ratio Test

In the likelihood ratio test, it is decided whether the given data has been drawn from the probability distribution p or p', where $p' \neq p$. It assumes that accurate estimates of both p or p' are available. According to the Neyman–Pearson lemma, the likelihood ratio test is the uniformly most powerful test for simple hypotheses, i.e., hypotheses which completely specify the distributions. The likelihood ratio test makes the decision based on the log-likelihood ratio (LLR) test statistic.

Definition 4.9. Let p and p' be probability distributions on A, and let q be an empirical distribution on A. The log-likelihood ratio is defined as

$$\mathrm{LLR}(q; p, p') = N \sum_{x \in A} q(x) \log \frac{p(x)}{p'(x)}.$$

The decision is made by computing the value of the LLR statistic for the given data set: the decision is p if $\mathrm{LLR}(q; p, p') \geq 0$; otherwise, it is p'.

Baignères et al. [1] proved the following theorem, which gives the data complexity of a distinguisher based on the LLR statistic.

Theorem 4.10. *Assume that probability distributions p and p' are close to each other. Then the data complexity of distinguishing p from p' is*

$$N = \frac{r}{C(p, p')},$$

where r is a small constant and $C(p, p')$ is the capacity between p and p'.

Hence, the data complexity of distinguishing whether the given data follows the uniform distribution $p' = p_0$ or is drawn from a close-to-uniform distribution p is $N = r/C(p)$. To achieve the success probability P_S, the constant r has to be chosen according to

$$r = 4\Phi^{-1}(P_S)^2,$$

where Φ is the cumulative distribution function of the standard normal distribution. For $P_S = 0.95$, we have $r \approx 8$.

4.4.2 Chi-Squared Test

In the χ^2 test, it is decided whether the given data set is drawn from a specific probability distribution or not. The χ^2 test can be used if enough information about the distributions is not available to use the LLR test. The test makes use of the following test statistic.

Definition 4.11. Let p' be a probability distribution and q an empirical distribution on A. The χ^2 test statistic is defined as

$$\chi^2(q;p') = N \sum_{x \in A} \frac{(q(x) - p'(x))^2}{p'(x)}.$$

Thus, large values of the χ^2 statistic indicate large differences between q and p'. The decision is made by comparing the value of $\chi^2(q;p')$ to a threshold τ, which depends on the size of the distribution (degrees of freedom) and the probabilities of the two types of errors, rejecting p' when it is right and accepting p' when it is wrong. The decision is p' if $\chi^2(q;p') \leq \tau$; otherwise, it is decided that the data has not been drawn from p'.

Vaudenay [43] proved the following result on the data complexity for χ^2 tests.

Theorem 4.12. *Assume that q is drawn from either a close-to-uniform distribution p or the uniform distribution $p' = p_0$. The data requirement of the χ^2 distinguisher is then given as*

$$N = \frac{r\sqrt{|A|}}{C(p)},$$

where r is a small constant and $C(p)$ is the capacity of p.

Assuming that $|A| \geq 2^8$, one can derive the estimate [38]

$$r \approx (\sqrt{2} + 2)\Phi^{-1}(P_S),$$

where P_S is the required success probability. For $P_S = 0.95$, we have $r \approx 8$.

4.5 Markov Chains

A stochastic process describes a dynamical system which can be in precisely one state at a time and the transitions between different states may involve some amount of uncertainty. A stochastic process is modeled as a sequence of random variables $(Q_t)_{t>0}$, where each random variable takes on values in some set S, which is called the state space of the process.

A Markov chain is a particular stochastic process, in which the next state of the system depends only on the current state and not on the preceding states. See Durrett [14] for an overview of Markov chains.

Definition 4.13. A Markov chain is a sequence $(Q_t)_{t>0}$ of random variables such that

$$\Pr(Q_{t+1} = q_{t+1} \,|\, Q_t = q_t, \ldots, Q_2 = q_2, Q_1 = q_1) = \Pr(Q_{t+1} = q_{t+1} \,|\, Q_t = q_t)$$

for all $t > 1$.

Any Markov chain can be represented as a linear dynamical system in the following manner. Let $(Q_t)_{t>0}$ be a Markov chain with the state space $S = \{s_1, s_2, \ldots, s_n\}$. To describe the probabilities of each state after t steps, we use the vector $r_t = (r_{t1}, r_{t2}, \ldots, r_{tn})^T$, where $r_{ti} = \Pr(Q_t = s_i)$ denotes the probability that the system is in state s_i at time t. We use

$$a_{ij} = \Pr(Q_{t+1} = s_j \,|\, Q_t = s_i)$$

to denote the transition probability that the system moves from state s_i to state s_j at time $t + 1$. The matrix

$$A = (a_{ij})_{n \times n}$$

is called the transition matrix of the system. According to the Chapman–Kolmogorov equation [14, p. 36], the state distribution of the Markov chain at time $t + 1$, $t > 0$, can then be determined as

$$r_{t+1} = Ar_t,$$

which defines a linear dynamical system.

4.6 Hidden Markov Models

A hidden Markov model (HMM) models a Markov chain with a finite number of possible states that are assumed to be directly unobservable. However, information about the state can be gained from the symbols that are emitted from each state at each time step. For an overview of HMMs, see Rabiner [41].

As Markov chains, HMMs can be represented as dynamical systems. An HMM is defined by the set of internal states, the set of transition probabilities between the states, the set of observable symbols, the set of probabilities that a certain symbol is emitted from a certain state, and the initial state distribution. The set of internal states is denoted by $S = \{s_1, s_2, \ldots, s_n\}$ and the state at time t is denoted by the random variable Q_t such that the sequence $(Q_t)_{t>0}$ forms a Markov chain. The set of transition probabilities in the system is denoted by $A = \{a_{ij}\}$, where

$$a_{ij} = \Pr(Q_{t+1} = s_j \,|\, Q_t = s_i)$$

for all $1 \leq i, j \leq n$ as before. The set of observable symbols is denoted by $V = \{v_1, v_2, \ldots, v_m\}$ and the observation emitted at time t is denoted by the random variable O_t. The set of emission probabilities is denoted by $B = \{b_j(k)\}$, where

$$b_j(k) = \Pr(O_t = v_k \,|\, Q_t = s_j)$$

for all $1 \leq j \leq n$ and $1 \leq k \leq m$. The initial state distribution indicates the probability distribution for the first state Q_1. It is denoted by $\pi = \{\pi_i\}$, where

$$\pi_i = \Pr(Q_1 = s_i)$$

for all $1 \leq i \leq n$. Using these parameters, we can define HMMs as follows.

Definition 4.14. A hidden Markov model is the tuple

$$\lambda = (A, B, \pi),$$

where A denotes the set of transition probabilities, B denotes the set of emission probabilities, and π denotes the initial state distribution.

4.6.1 The Three Basic Problems for HMMs

HMMs make it possible to analyze the relationship between the internal states of a dynamical system and the observations emitted from the system. We present three problems which are relevant for this kind of analysis and give a brief overview of the methods used to solve these problems. The presented problems are sometimes called the three basic problems for HMMs in literature, e.g., by Rabiner [41] who we also refer to for a more detailed discussion about these problems.

Given a model $\lambda = (A, B, \pi)$, we let $O = O_1 O_2 \ldots O_T$ and $Q = Q_1 Q_2 \ldots Q_T$ denote sequences of random variables representing emitted observations and visited states, respectively. We use n to denote the number of internal states in the HMM as before.

Problem 1. Given an observation sequence $o = o_1 o_2 \ldots o_T$ and a model λ, how do we efficiently compute $\Pr(O = o \,|\, \lambda)$, the probability of the observation sequence given the model?

Problem 2. Given an observation sequence $o = o_1 o_2 \ldots o_T$ and a model λ, what is the most likely state sequence $q = q_1 q_2 \ldots q_T$ that produced the observations?

Problem 3. Given an observation sequence $o = o_1 o_2 \ldots o_T$ and a model λ, how do we adjust the parameters of the model $\lambda = (A, B, \pi)$ to maximize $\Pr(O = o \,|\, \lambda)$?

Problem 1 is called the evaluation problem since it is concerned with finding the probability that a specific observation sequence has been produced by the given model. The problem is solved by the forward–backward algorithm, which is able to efficiently compute this probability.

Problem 2 is known as the decoding problem. It is related to understanding the behavior of a system based on the observations emitted from the system. The goal is to find the most probable state sequence that has produced the given observation sequence. One solution to this problem is offered by the Viterbi algorithm [17, 44], which is able to efficiently compute the state sequence q that maximizes $\Pr(Q = q \,|\, O = o, \lambda)$, where o is the given observation sequence. The time complexity of the Viterbi algorithm is $O(Tn^2)$. In cryptography, the decoding problem is relevant for side-channel cryptanalysis, since it allows the attacker to infer internal behavior of the system based on physical measurements if a suitable model for the system exists. An application of the Viterbi algorithm to cache-timing attacks is presented in Publication V.

Problem 3 is known as the learning problem. It asks how to adjust the parameters of the model $\lambda = (A, B, \pi)$ to maximize the probability that the given observation sequence is produced. There is no known analytical method to achieve this. However, it is possible to adjust the parameters such that the probability $\Pr(O = o \,|\, \lambda)$ is locally maximized using iterative procedures such as the Baum–Welch algorithm [3] or gradient techniques [27]. Each iteration of the Baum–Welch algorithm has the time complexity $O(Tn^2)$. Adjusting the parameters is often called training the HMM and it typically involves collecting a set of observation sequences from a real physical phenomenon, which are then used in training. Like the decoding problem, the learning problem is also relevant in side-channel cryptanalysis. In Publication V, an application of the Baum–Welch algorithm is presented for adjusting model parameters based on cache-timing data.

4.7 Vector Quantization

Vector quantization (VQ) is a statistical technique that can be used for classifying new observations into categories which have been determined using a training data set containing observations with known categories. An application of VQ is presented in Publication V, where it was used in combination with an HMM to analyze cache-timing data.

Each category can be defined using a prototype vector and a label. We use P to denote the set of prototype vectors called the codebook and L to denote the set of labels for each codebook vector. In addition, we have a mapping $l\colon P \to L$ that associates each codebook vector with a label. When a new observation v is obtained, it is classified using a vector quantizer q defined by $q(v) = \arg\min_{p \in P} d(v, p)$, where $d(v, p)$ is the Euclidean distance between v and the codebook vector p. The label of v is then $l(q(v))$.

We can employ learning vector quantization (LVQ) [23] to create the codebook. It works in the following way. First, we obtain a training data set D containing vectors with predetermined labels in L, and an initialization for the codebook P, which can be derived by performing k-means clustering [30] on all training vectors sharing the same label and taking the resulting centroids. The following process is then iterated until an acceptable error rate is achieved. We (randomly) pick a training vector from D and check if it is correctly classified using the vector quantizer q: if it is, we move the matching codebook vector closer to the training vector; otherwise, the codebook vector is moved away.

5. Linear Distinguishing Attacks

A distinguishing attack is a form of cryptanalysis, where the attacker is able to distinguish whether the given set of samples has been generated by a particular cipher or not. In linear distinguishing attacks, a linear transformation is first applied to the given set of samples and the resulting data set is then studied using a statistical hypothesis test. This chapter presents a framework for linear distinguishing attacks that can employ linear mappings over arbitrary Abelian finite groups. We first discuss statistical testing in distinguishing attacks of data sets and then linear approximation of functions and block ciphers. Linear approximations are used in estimating the efficiency of the attack. Finally, we briefly describe the linear distinguishing attacks presented in Publication I and Publication IV.

5.1 Linear Distinguishers

A distinguishing attack on a cipher is an attack, where the attacker is able to tell whether the given samples have been generated by the cipher or not. Distinguishing attacks are essentially statistical hypothesis tests, where the aim is to detect any statistical bias in the given set of samples. In linear distinguishing attacks, a linear transformation is first applied to the given set of samples. The purpose is to expose statistical nonuniformity in the data if it originates from the cipher. The decision is then made by studying the distribution of the resulting data set using a statistical hypothesis test.

Finding an efficient linear transformation is a cipher-specific task. It usually involves constructing an approximative linear model for the cipher by finding linear approximations for the nonlinear components of the cipher. The more accurate approximations are found, the more efficient linear transformation can be constructed. For nonlinear filter generators with a linear feedback function, one typically forms a linear approximation for the filter function and uses the linear feedback relation to cancel out the state variables involved in the approximation such that the remaining variables induce a nonuniform distribution. An example of this is presented in [39].

The efficiency of the transformation depends on how nonuniform distribution it is capable of inducing. Hence, it is inherently connected to the method used for measuring the nonuniformity in the distribution and also to the used hypothesis test. These aspects have been studied in many papers regarding attacks using linear transformations from $\mathbb{Z}_2^n$ to $\mathbb{Z}_2$. For example, the seminal work by Matsui and Yamagishi [35] and Matsui [34] is based on this kind of a transformation. Linear attacks using a number of statistically independent one-dimensional transformations (or equivalently, linear approximations) have been studied by Kaliski and Robshaw [20] and Biryukov et al. [4]. The first attack based on a truly multidimensional transformation was presented by Englund and Maximov [15]. Baignères et al. [1] proved several useful results regarding the foundations of such attacks and Hermelin et al. [19] presented a method for constructing multidimensional approximations from one-dimensional approximations. Publication IV gives a practical multidimensional distinguishing attack on the stream cipher Shannon. In this thesis, linear distinguishing attacks are presented in a general setting, where any linear transformations between finite Abelian groups can be used. This has been previously studied by Baignères et al. [2]. Publication I presents such an attack on the block cipher DEAN.

Our focus is on a block cipher defined using the encryption function $\mathcal{E}_K \colon A \to A$, where K is the secret key and A is a finite Abelian group. The transformation used in the distinguisher is a homomorphic projection $T \colon A \times A \to B$, where B is a subgroup of $A \times A$. Now suppose that we have N data pairs (x_t, y_t), $1 \leq t \leq N$, belonging to $A \times A$. The empirical distribution q used in the hypothesis test is computed as

$$q(y) = \frac{|\{1 \leq t \leq N \,:\, T(x_t, y_t) = y\}|}{N}$$

for all $y \in B$. We say that the distribution q originates from the cipher if $y_t = \mathcal{E}_K(x_t)$ for all $1 \leq t \leq N$ with some unknown key K. In the following sections, we present two hypothesis tests that can be used in the distinguisher to study the empirical distribution. They depend on the attack model which is chosen according to how much information obtained from the cipher in the analysis.

5.1.1 The LLR distinguisher

The LLR distinguisher makes the decision using the likelihood ratio test. The attack model assumes that if the empirical distribution q originates from the cipher, then the samples have been drawn according to an unknown distribution belonging to a collection $p_1, p_2, \ldots, p_l$ of known distributions. To decide whether the data set originates from the cipher or follows the uniform distribution p_0, the attacker will compute the LLR statistic

$$\mathrm{LLR}(q; p_i, p_0) = N \sum_{y \in B} q(y) \log \frac{p_i(y)}{p_0(y)}$$

for each $1 \leq i \leq l$. If the maximum of $\mathrm{LLR}(q; p_i, p_0)$ over all $1 \leq i \leq l$ is nonnegative, it is decided that q originates from the cipher.

Assuming that the distributions $p_1, p_2, \ldots, p_l$ are close-to-uniform, the data requirement for the LLR distinguisher can then be determined according to Hermelin et al. [19] from

$$N_{\mathrm{LLR}} = \frac{r}{\min_{1 \leq i \leq l} C(p_i)},$$

where r is a small constant that depends on the desired success probability P_S. To achieve $P_S = 0.95$, we can choose $r = 8$. In typical cipher construction where the secret key is added to the data using the group operation, particularly in so-called key-alternating ciphers, the distributions $p_1, p_2, \ldots, p_l$ tend to be permutations of each other and thus have the same capacity, which is equal to the average capacity taken over all keys [19]. We then estimate that the success probability is significant for about half of the keys with the sample size N_{LLR} when $r = 8$.

5.1.2 The χ^2 distinguisher

The χ^2 distinguisher performs a χ^2 test to make the decision. Unlike in the LLR distinguisher, the attack model does not assume knowledge of the distributions $p_1, p_2, \ldots, p_l$ originating from the cipher. The cipher distributions are usually key-dependent, so it tends to be infeasible to determine them for all keys. Moreover, if the size of B is small, the capacities of the distributions vary significantly with the key. When the size of B increases, the capacities may tend to get closer to the average capacity of the cipher taken over all keys [7]. A distinguisher based on the χ^2 statistic can be created under such an assumption.

Let us denote $M = |B|$. The χ^2 test statistic for the empirical distribution q and the uniform distribution p_0 is then given by

$$R = \chi^2(q; p_0) = NM \sum_{y \in B} \left(q(y) - \frac{1}{M} \right)^2 .$$

If the empirical distribution q is drawn from a cipher distribution, the expected value of R/N can be estimated with the expected capacity C taken over the cipher distributions $p_1, p_2, \ldots, p_l$. On the other hand, if q follows the uniform distribution p_0, then R follows the χ^2 distribution with $M-1$ degrees of freedom. The expected value of R/N is then $(M-1)/N$. If R/N is closer to C than $(M-1)/N$, then the hypothesis that q originates from the cipher is accepted.

Assuming that the cipher distributions $p_1, p_2, \ldots, p_l$ are close-to-uniform and that q is drawn from one of the cipher distributions or from the uniform distribution p_0, the data requirement of the χ^2 distinguisher can then be estimated by Theorem 4.12 to be

$$N_{\chi^2} = \frac{r\sqrt{M}}{C},$$

where r is a small constant depending on the success probability P_S and C is the expected capacity taken over the cipher distributions. To achieve $P_S = 0.95$, we choose $r = 8$.

5.2 Linear Approximation of Functions

Let A and B be finite Abelian groups and $f \colon A \to B$ be a mapping. Let χ_u and ψ_v be characters of A and B, respectively.

Definition 5.1. A linear approximation $L_f(u, v)$ of f is the function

$$x \mapsto \psi_v(f(x))\overline{\chi_u(x)}.$$

We call $u \in A$ the input mask and $v \in B$ the output mask of the approximation.

In linear cryptanalysis, the attacker usually tries to find linear approximations that have as nonuniform output distributions as possible. If the nonuniformity of the distribution can be exploited in a statistical attack, we say that the linear approximation is strong. In the optimal case (for the attacker), the function f is a homomorphism, which is possible only if the linear approximations are constant.

The linear approximation of a function $f\colon A \to B$ is sometimes based on characters of certain groups that are different from the groups A and B used in defining the function. To emphasize the difference, we will talk about the domain of the approximation: if characters of $A' \neq A$ and $B' \neq B$ are used for characters of the approximation, the groups A' and B' define the domain of the approximation.

5.2.1 Correlation of a Linear Approximation

The strength of the linear approximation is measured using a quantity called correlation. As shown later, the correlation of a linear approximation of a Boolean function reflects the Hamming distance of the Boolean function to a linear function.

Definition 5.2. The correlation of a linear approximation $L_f(u,v)$ of f is the correlation coefficient

$$c_f(u,v) = c(\psi_v \circ f, \chi_u)$$

between functions $\psi_v \circ f$ and χ_u.

Thus, a linear approximation $L_f(u,v)$ with a large correlation indicates a large correlation between the homomorphism χ_u and the homomorphic projection $\psi_v \circ f$ of f. If $u = 0$, the linear approximations of f have the form

$$x \mapsto \psi_v(f(x))$$

since χ_0 is a trivial character. In this case, the correlation is simply denoted as $c_f(v)$.

If f is bijective, we obtain the following relationship between the correlations of linear approximations of f and f^{-1} by making the substitution $y = f(x)$ in the Fourier transform of f.

Theorem 5.3. *If $f\colon A \to B$ is bijective, then*

$$c_{f^{-1}}(u,v) = \overline{c_f(v,u)}.$$

5.2.2 Approximation of Boolean Functions

Binary linear approximations are approximations in which characters of a binary vector space are used. Let $f, g\colon \mathbb{Z}_2^n \to \mathbb{Z}_2$ be Boolean functions. The Hamming distance $d_{\mathrm{H}}(f,g)$ between f and g is defined as the number of elements $x \in \mathbb{Z}_2^n$ for which $f(x) \neq g(x)$. Given a Boolean function f, the correlation of a linear approximation of f is related to the Hamming distance between f and a linear function as demonstrated by the following example.

Let $f\colon \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ be a vector-valued Boolean function. The characters of $\mathbb{Z}_2^n$ and $\mathbb{Z}_2^m$ can be defined as $\chi_u(x) = (-1)^{u\cdot x}$ and $\psi_v(x) = (-1)^{v\cdot x}$, respectively. The correlation of a linear approximation of f is then given by

$$c_f(u,v) = c(\psi_v \circ f, \chi_u) = \frac{1}{2^n} \sum_{x\in\mathbb{Z}_2^n} (-1)^{v\cdot f(x)\oplus u\cdot x}.$$

Let φ_a denote the linear function $x \mapsto a\cdot x$, where $a = u$ or $a = v$. It is then easy to verify that

$$c_f(u,v) = 1 - 2^{1-n} d_{\mathrm{H}}(\varphi_v \circ f, \varphi_u).$$

Thus, smaller Hamming distance between $x \mapsto v \cdot f(x)$ and $x \mapsto u \cdot x$ indicates larger correlation for the binary linear approximation $L_f(u,v)$. Indeed, a linear approximation of a vector-valued Boolean function f is often defined as the mapping $x \mapsto v\cdot f(x) \oplus u\cdot x$ and not as $x \mapsto (-1)^{v\cdot f(x)\oplus u\cdot x}$ as indicated by the general definition given before. The correlation, however, is the same.

5.2.3 Approximation of Homomorphic Functions

Suppose that $f\colon A \to B$ is a homomorphism, and let χ_u and ψ_v be characters of A and B, respectively. The composition of two homomorphisms is clearly a homomorphism, so $\psi_v \circ f$ is a character of A. By the orthogonality relation for characters, the correlation of a linear approximation of f can be determined as

$$c_f(u,v) = \frac{1}{|A|}\sum_{x\in A} \psi_v(f(x))\overline{\chi_u(x)} = \begin{cases} 0 & \text{if } \psi_v \circ f \neq \chi_u, \\ 1 & \text{if } \psi_v \circ f = \chi_u. \end{cases}$$

Since $\psi_v \circ f$ and χ_u are both characters of A, there always exists such $u \in A$ and $v \in B$ that the correlation of the linear approximation is one. Hence, for every homomorphism, there exists a linear approximation with correlation one.

5.3 Linearity and Nonlinearity

The linearity of a function indicates the maximum correlation between any nontrivial homomorphic projection of the function and any homomorphic function. In other words, it indicates the largest correlation of a nontrivial linear approximation of the function. Conversely, the nonlinearity of a function reflects the minimum correlation of a function to homomorphic functions. Higher nonlinearity generally means better resistance to linear cryptanalysis.

Let $f\colon A \to B$ be a function between finite Abelian groups A and B.

Definition 5.4. The linearity of f is defined as

$$\mathcal{L}(f) = |A| \max_{u \in A} \max_{\substack{v \in B \\ v \neq 0}} |c_f(u,v)|.$$

Definition 5.5. The nonlinearity of f is defined as

$$\mathcal{N}(f) = \frac{|A| - \mathcal{L}(f)}{|B|}.$$

Theorem 5.3 then yields the following result.

Theorem 5.6. *If f is bijective, then*

$$\mathcal{L}(f^{-1}) = \mathcal{L}(f).$$

5.3.1 Linearity of Boolean Functions

When A and B are binary vector spaces, the nonlinearity of a function reflects the minimum Hamming distance of the linear projection of the function to the set of all linear (and also affine) functions.

Let $f\colon \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ be a vector-valued Boolean function. The characters of $\mathbb{Z}_2^n$ and $\mathbb{Z}_2^m$ have the form $\chi_u(x) = (-1)^{u \cdot x}$ and $\psi_v(x) = (-1)^{v \cdot x}$, respectively. The nonlinearity of f can then be written as

$$\mathcal{N}(f) = 2^{n-1} - 2^{n-1} \max_{u \in \mathbb{Z}_2^n} \max_{\substack{v \in \mathbb{Z}_2^m \\ v \neq 0}} |c_f(u,v)|.$$

By the definition of nonlinearity and the relationship between correlation and Hamming distance, we can then deduce that

$$\mathcal{N}(f) = \min_{u \in \mathbb{Z}_2^n} \min_{\substack{v \in \mathbb{Z}_2^m \\ v \neq 0}} d_{\mathrm{H}}(\varphi_v \circ f, \varphi_u).$$

5.3.2 Limits for Linearity

High nonlinearity is one desirable property for S-boxes. The following result gives upper and lower bounds for the linearity of a mapping $f\colon A \to B$, where A and B are arbitrary finite Abelian groups.

Theorem 5.7. *Given a mapping* $f\colon A \to B$*, we have*

$$\sqrt{|A|} \le \mathcal{L}(f) \le |A|.$$

Proof. We follow the ideas of Drakakis et al. [13] to prove the result. Let $F\colon A \times B \to \{0,1\}$ be the indicator function of f defined as

$$F(x,y) = \begin{cases} 1 & \text{if } y = f(x), \\ 0 & \text{otherwise.} \end{cases}$$

Let χ_u be a character of A and ψ_v be a character of B. It follows that

$$\begin{aligned} \widehat{F}(u,v) &= \sum_{x\in A, y\in B} F(x,y)\overline{\chi_u(x)\psi_v(y)} = \sum_{x\in A} \overline{\chi_u(x)\psi_v(f(x))} \\ &= \sum_{x\in A} \psi_{-v}(f(x))\overline{\chi_u(x)} = |A|\, c_f(u,-v). \end{aligned} \tag{5.1}$$

Using Parseval's theorem (3.1), we get

$$\sum_{u\in A, v\in B} |\widehat{F}(u,v)|^2 = |A||B| \sum_{x\in A, y\in B} |F(x,y)|^2 = |A|^2|B|.$$

We cannot have $|\widehat{F}(u,v)| < \sqrt{|A|}$ for all $u \in A$ and $v \in B$, since it implies

$$\sum_{u\in A, v\in B} |\widehat{F}(u,v)|^2 < |A|^2|B|.$$

Therefore, $|\widehat{F}(u,v)| \ge \sqrt{|A|}$ for some $u \in A$ and $v \in B$. From the definition of linearity and (5.1) it then follows that $\mathcal{L}(f) \ge \sqrt{|A|}$. Obviously, $|\widehat{F}(u,v)| < |A|$ always holds. We have thus shown the result. □

The functions that achieve the lower bound in the previous theorem, are generally called bent functions, defined originally by Rothaus [42] who studied bent Boolean functions. Since then, the notion of bentness has been studied in many different domains. Kumar et al. [24] studied the class of functions from $\mathbb{Z}_m^n$ to $\mathbb{Z}_m$. Logachev et al. [31] adapted the notion of bentness to functions from any finite Abelian group A to the multiplicative group of complex numbers of magnitude one, which is essentially the same set of functions as considered above. In Publication II, it is shown that the linearity of Exponential and Logarithmic Welch Costas functions is larger than the lower bound only by a logarithmic factor. They are functions defined on $\mathbb{Z}_n$, where $n = p - 1$ and p is a prime. Publication III examines the linearity of discrete logarithm on $\mathbb{F}_2^n$ and establishes upper bounds which, depending of the type of the linear approximation, are close to the lower bound.

5.4 Approximation over Composed Functions

Block ciphers and stream ciphers commonly employ consecutive nonlinear functions. In this section, we will show how to determine the correlation of a linear approximation over composed functions using the correlations of linear approximations over individual functions.

Suppose that A, B, and C are finite Abelian groups. Let $f\colon A \to B$ and $g\colon B \to C$ be functions and let $L_f(u,w)$ and $L_g(w,v)$ be their linear approximations. We first consider a situation, where the input to g is computed as $f(x)+y$, where y is statistically independent of x and uniformly distributed. Such a composition of f and g is the function $F\colon A \times B \to C$ defined as

$$F(x,y) = g(f(x)+y)$$

and the correlations of its linear approximations are

$$c_F((u,w),v) = c_f(u,w)c_g(w,v).$$

The maximum of this value taken over $w \in B$ is often used for estimating the correlation of the composed function $g \circ f$, and is quoted as Piling-up lemma [34]. In cryptographic contexts, however, the variable y is the key, which is a fixed value. The accurate expression for the correlation of $g \circ f$ was given by Daemen et al. [11] in the binary case, and in the general case by Baignères et al. [2].

Theorem 5.8 (Correlation theorem)**.** *Let $f\colon A \to B$ and $g\colon B \to C$ be functions. The correlation of a linear approximation of $g \circ f$ is given by*

$$c_{g\circ f}(u,v) = \sum_{w\in B} c_f(u,w)c_g(w,v).$$

Proof. By expressing the correlations in terms of Fourier transform and using the orthogonality relation for characters, we get

$$\begin{aligned}
&\sum_{w\in B} c_f(u,w)c_g(w,v)\\
&= \sum_{w\in B} \frac{1}{|A|}\sum_{x\in A} \lambda_w(f(x))\overline{\chi_u(x)}\frac{1}{|B|}\sum_{y\in B}\psi_v(g(y))\overline{\lambda_w(y)}\\
&= \frac{1}{|A||B|}\sum_{x\in A, y\in B}\psi_v(g(y))\overline{\chi_u(x)}\sum_{w\in B}\lambda_w(f(x))\overline{\lambda_w(y)}\\
&= \frac{1}{|A|}\sum_{x\in A}\psi_v(g(f(x)))\overline{\chi_u(x)}\\
&= c_{g\circ f}(u,v)
\end{aligned}$$

for all $u \in A$ and $v \in C$. □

The correlation theorem indicates that the correlation over several consecutive functions is determined using all possible intermediate mask combinations. A fixed mask combination is often called a linear trail, and the set of all linear trails is called the linear hull of the cipher.

In practice it is usually not possible to compute the exact correlation using all linear trails. Instead, it is assumed that the round keys are statistically independent and uniformly distributed. With this assumption, it is possible to estimate the correlation using the Piling-up lemma. If there is one dominant linear trail, the obtained estimate is accurate. However, if there are many linear trails with significant correlations, this kind of estimate can be rather inaccurate.

Using the triangle inequality, we obtain

$$|c_{g \circ f}(u, v)| \leq \sum_{w} |c_f(u, w)||c_g(w, v)|.$$

5.5 Approximation of Block Ciphers

Many block ciphers are constructed by iterating the same round function for a number of rounds. At each round, a round key is typically combined with the data using some group operation, such as the bitwise XOR. The round keys are usually determined from the key using by simple operations. In this section, we study the correlation of an iterated block cipher, where round keys are used at each round.

Let A be a finite Abelian group written additively. Recall that the encryption function $\mathcal{E}_K$ of an R-round iterated key-alternating block cipher on A is specified by a sequence of round functions $g_1, g_2, \ldots, g_R$ on A and a key $K = (K_1, K_2, \ldots, K_R) \in A^R$ such that the encryption $\mathcal{E}_K(x)$ of plaintext $x \in A$ is computed as

$$\begin{aligned} x_0 &= x, \\ x_r &= g_r(x_{r-1} + K_r) \quad \text{for } r = 1, 2, \ldots, R, \\ \mathcal{E}_K(x) &= x_R. \end{aligned}$$

We denote by G_r the round function such that $G_r(x) = g_r(x + K_r)$. Although we have restricted ourselves to group A on each round, it would be straightforward to extend the following analysis to include the use of different groups at different rounds.

5.5.1 Expected Correlation

Let χ_u denote a character of A. Given a linear approximation $L_{G_r}(u_r, u_{r-1})$ of the round function G_r, its correlation is equal to

$$c_{G_r}(u_{r-1}, u_r) = \frac{1}{|A|}\sum_{x\in A}\chi_{u_r}(g_r(x+K_r))\overline{\chi_{u_{r-1}}(x)} = \chi_{u_{r-1}}(K_r)c_{g_r}(u_{r-1}, u_r).$$

Since the encryption function $\mathcal{E}_K$ is a composition of the round functions, its correlation can be determined using the correlation theorem as

$$c_{\mathcal{E}_K}(u,v) = \sum_{u_1,\ldots,u_{R-1}}\prod_{r=1}^{R}\chi_{u_{r-1}}(K_r)c_{g_r}(u_{r-1}, u_r),$$

where $u_0 = u$ and $u_R = v$. The proof of this result is a straightforward generalization of the same result in the binary case due to Daemen et al. [11].

As noted previously, the success probability of a linear attack depends on the squared absolute value of the correlation of the linear approximation that is used in the attack. Since the correlation may vary a lot with the key in modern block ciphers, the expected squared absolute value of the correlation is frequently used to measure the strength of a linear approximation. The proof of the following theorem is also a generalization of the proof in binary case due to Nyberg [37]. An alternative proof using the Markov property has been given by Baignères et al. [2].

Theorem 5.9. *Let $g_1, g_2, \ldots, g_R$ be the round functions of an R-round key-alternating iterated block cipher $\mathcal{E}_K$ on a finite Abelian group A with the key $K = (K_1, K_2, \ldots, K_R)$. If K is drawn from the uniform distribution on A^R, then*

$$\mathrm{E}_K|c_{\mathcal{E}_K}(u,v)|^2 = \sum_{u_1,\ldots,u_{R-1}}\prod_{r=1}^{R}|c_{g_r}(u_r, u_{r-1})|^2, \tag{5.2}$$

where $u_0 = u$ and $u_R = v$.

5.5.2 Expected Capacity

As before, let $\mathcal{E}_K$ be an iterated key-alternating block cipher defined on a finite Abelian group A with a key $K \in A^R$. Since the size of the whole distribution of values $\{(x, \mathcal{E}_K(x)) : x \in A\}$ is intractable, as computing its capacity would require to compute the sum of the squared absolute correlations for all characters of $A \times A$, a cryptanalyst must restrict attention to a subgroup of characters. In the applications to be considered in this paper, such character subgroups are formed by characters on subgroups of $A \times A$.

Assume that A' and B' are subgroups of A. Let $B = A' \times B'$ be a subgroup of $A \times A$ and let λ_w denote a character of B. Let $f_K \colon A \to A \times A \to B$ be the mapping defined as

$$x \mapsto (x, \mathcal{E}_K(x)) \mapsto T(x, \mathcal{E}_K(x)),$$

where T is a homomorphic projection from $A \times A$ to its subgroup B. From Theorem 4.8, it is then easy to see the following result.

Lemma 5.10. *The capacity of f_K can be computed as*

$$C(f_K) = \sum_{w \neq 0} |c_{f_K}(w)|^2. \tag{5.3}$$

A character of B can be written as $\lambda_w = \overline{\chi_u} \times \psi_v$, where χ_u and ψ_v are characters of A' and B', respectively. Because the characters of a subgroup are simply restrictions of the characters of the entire group, χ_u and ψ_v also define characters on A. We get

$$c_{f_K}(w) = c(\lambda_w \circ f_K, \chi_0) = c(\psi_v \circ \mathcal{E}_K, \chi_u) = c_{\mathcal{E}_K}(u, v). \tag{5.4}$$

We can then take the expected value of the squared absolute correlation of a linear approximation of $\mathcal{E}_K$ from (5.2) and use the identities (5.4) and (5.3) to determine the expected capacity of f_K as given in the following theorem.

Theorem 5.11. *Let $g_1, g_2, \ldots, g_R$ be the round functions of an R-round key-alternating iterated block cipher $\mathcal{E}_K$ on a finite Abelian group A with the key $K \in A^R$. Let f_K and T be functions defined as above. If K is drawn from the uniform distribution on A^R, then*

$$\mathrm{E}_K(C(f_K)) = \sum_{u_0, \ldots, u_R} \prod_{r=1}^{R} |c_{g_r}(u_{r-1}, u_r)|^2, \tag{5.5}$$

where the sum is taken over all $u_0 \in A'$ and $u_R \in B'$ such that either u_0 or u_R is nonzero.

5.5.3 Computing Expected Capacity

Cho [7] designed a customized algorithm to compute the expected capacity of a multidimensional linear approximation of the binary block cipher PRESENT iteratively from round to round. Based on similar approach, we can devise a general algorithm for the same purpose. By reordering the terms in (5.5), we get the following iterative algorithm for computing the expected capacity of f_K for a general key-alternating block cipher $\mathcal{E}_K$ in the case where B takes the form $A' \times B'$.

Theorem 5.12. *Let f_K be the mapping defined before and let $\Gamma_i[\cdot]$ denote an indexed array for $0 \leq i \leq R$. The expected capacity $C = \mathrm{E}_K(C(f_K))$ is computed by the following iterative procedure:*

$$\begin{aligned} \Gamma_0[u_0] &= 1 \quad \textit{for all } u_0 \in A', \\ \Gamma_r[u_r] &= \sum_{u_{r-1} \neq 0} |c_{g_r}(u_r, u_{r-1})|^2 \Gamma_{r-1}[u_{r-1}] \quad \textit{for } r = 1, 2, \ldots, R, \\ C &= \sum_{\substack{u_R \in B' \\ u_R \neq 0}} \Gamma_R[u_R]. \end{aligned}$$

Many correlations of linear approximations over intermediate rounds will trivially vanish, but still the number of characters with nonzero correlation is usually too large. This means that in practice only a lower bound of the capacity can be determined using this algorithm. How to select the sets of intermediate masks must be determined for each cipher separately, and requires different heuristic strategies to be used.

5.6 Applications

5.6.1 DEAN

DEAN [2] is a nonbinary AES-like key-alternating block cipher that encrypts blocks of 18 decimal digits. Its resistance to linear cryptanalysis was studied in Publication I using the iterative procedure presented in Theorem 5.12. A block in DEAN is represented as a 3×3 array of elements from the additive group $\mathbb{Z}_{10}^2$. Linear approximations of DEAN were thus constructed using characters of the group $A = (\mathbb{Z}_{10}^2)^9$.

Using a bound derived from a single linear trail, Baignères et al. [2] estimated that the best LLR distinguisher would require the full code book of known plaintext blocks to succeed over four rounds of DEAN. The results presented in Publication I indicate that the designers' bound is not sufficient: At least five rounds are needed before the data complexity of the χ^2 distinguisher exceeds the code book bound. If resistance against the LLR distinguisher is wanted, then at least seven rounds are needed.

5.6.2 Shannon

Shannon [18] is a stream cipher that is based on a nonlinear feedback shift register (NLFSR) and a nonlinear filter (NLF) function. A practical linear distinguisher for it was presented in Publication IV. The distinguisher is based on a particular linear combination of keystream words derived by combining certain linear approximations of the NLFSR and the NLF at specific time instances. The used nonlinear functions have strong linear approximations which makes the final linear combination biased. The decision is made using the LLR test, which is justified assuming that the initial state is uniformly distributed. The data complexity predicted by linear approximation of the cipher is about 2^{31} with the confidence level $P_S = 0.92$. In the practical tests, the distinguisher was observed to require about 2^{34} keystream words to achieve the same level of confidence, which shows that the determined linear approximation of the cipher gives a good estimate for the data complexity.

6. Nonlinearity of S-boxes

In this chapter, we study linearity properties of some individual functions in different domains. We show how the previous concepts can be used for analyzing nonlinear mappings. Our interest is in functions that have simple representations in certain algebraic structures, but can still be used as nonlinear mappings in conjunction with other functions. We give results on addition modulo 2^n, multiplication modulo $2^n + 1$, discrete logarithm in $\mathbb{F}_{2^n}$, and Exponential Welch Costas (EWC) functions that are based on discrete logarithm in $\mathbb{Z}_p$. An upper bound for the linearity of a linear combination of the output bits of discrete logarithm in $\mathbb{F}_{2^n}$ was derived in Publication III. In this chapter, we derive a new upper bound that gives more accurate results for certain linear combinations than the previous bound. The results on the linearity of EWC functions were presented in Publication II.

6.1 Addition Modulo 2^n

Addition modulo 2^n is one of the most commonly used operations in symmetric ciphers. It can be used in combination with other simple operations, such as bitwise XORs and rotations, to induce diffusion in the cipher. In IDEA [25], it is one of the operations used for combining subkeys at each round of the cipher.

Binary linear approximations of addition of two integers modulo 2^n have been studied in [39, 45]. Wallén [45] presented an $\Theta(\log n)$-time algorithm for computing the correlation of linear approximation of addition modulo 2^n, an optimal algorithm for generating all linear approximations with a given nonzero correlation coefficient, and determined the distribution of the correlation coefficients. Nyberg and Wallén [39] presented a more general technique for computing correlation coefficients for addition of more than two integers and generating linear approximations for addition of two integers.

As noted in the previous chapter, the data complexity of a linear distinguishing attack on an iterative block cipher is dependent on the capacity of the used linear approximation. An estimate for the capacity can be obtained by using the expected capacity over the round keys. In this section, we describe how to obtain the expected value of squared absolute correlation for constant addition modulo 2^n such that the algorithm of Wallén [45] can be used to speed up the computation.

For a constant $K \in \mathbb{Z}_{2^n}$, we let $\mathrm{add}_K \colon \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ denote the constant addition function

$$\mathrm{add}_K(x) = x + K.$$

In addition, we let $\mathrm{add} \colon \mathbb{Z}_{2^n} \times \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ denote addition modulo 2^n with two inputs. The correlation of a (binary) linear approximation of add is denoted by $c_{\mathrm{add}}((u,w),v)$, where $u, w \in \mathbb{Z}_2^n$ are the input masks of the approximation and $v \in \mathbb{Z}_2^n$ is the output mask. By Theorem 1 of Nyberg [37], it is straightforward to derive a relationship between the correlations for add_K and add.

Theorem 6.1. *Let* add_K *denote a constant addition function on* $\mathbb{Z}_{2^n}$ *and* add *denote addition on* $\mathbb{Z}_{2^n}$ *with two inputs. Suppose that* $u, v, w \in \mathbb{Z}_2^n$. *If* K *is drawn from the uniform distribution on* $\mathbb{Z}_{2^n}$, *then*

$$\mathrm{E}_K |c_{\mathrm{add}_K}(u,v)|^2 = \sum_{w \in \mathbb{Z}_{2^n}} c_{\mathrm{add}}((u,w),v)^2.$$

Every correlation coefficient $c_{\mathrm{add}}((u,w),v)$ can be computed in $\Theta(\log n)$ time using the algorithm of Wallén [45]. Hence, it takes $\Theta(2^n \log n)$ time to compute $\mathrm{E}_K |c_{\mathrm{add}_K}(u,v)|^2$ for any $u, v \in \mathbb{Z}_2^n$.

6.2 Multiplication Modulo $2^n + 1$

Multiplication modulo $2^n + 1$ is also an operation used in the block cipher IDEA to combine subkeys. It is used in combination with addition in $\mathbb{Z}_2^n$ and addition in $\mathbb{Z}_{2^n}$ to induce nonlinearity in the cipher. In this section, we derive a relationship between correlations of linear approximations of constant multiplication and discrete logarithm.

For a constant $K \in \mathbb{Z}_{2^n}$, we let $\mathrm{mult}_K \colon \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ denote the constant multiplication function used in IDEA defined as

$$\mathrm{mult}_K(x) = g^{-1}(g(x)g(K) \bmod 2^n + 1),$$

where $g\colon \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n+1}$ is defined by

$$g(x) = \begin{cases} x & \text{if } x \neq 0, \\ 2^n & \text{if } x = 0. \end{cases}$$

Let us now assume that $2^n + 1$ is prime as in IDEA. Suppose that α is a generator of $\mathbb{Z}^*_{2^n+1}$ and let $\log_\alpha x$ denote the discrete logarithm of $x \in \mathbb{Z}^*_{2^n+1}$ to the base α. We can now reformulate mult_K as

$$\mathrm{mult}_K(x) = f^{-1}(f(x) + f(K) \bmod 2^n), \tag{6.1}$$

where $f\colon \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ is defined by

$$f(x) = \begin{cases} \log_\alpha x & \text{if } x \neq 0, \\ \log_\alpha 2^n & \text{if } x = 0. \end{cases}$$

Since $2^n = -1 = \alpha^{2^n/2} \pmod{2^n+1}$, we can also deduce that $f(0) = \log_\alpha 2^n = \log_\alpha(-1) = 2^{n-1} \pmod{2^n}$. Using the correlation theorem, we can now compute the expected value of the squared absolute correlation of a linear approximation of mult_K in the following manner.

Theorem 6.2. *Let* mult_K *be the constant multiplication function on* $\mathbb{Z}_{2^n}$ *defined as before and suppose that* $u, v \in \mathbb{Z}_{2^n}$. *If* K *is drawn from the uniform distribution on* $\mathbb{Z}_{2^n}$, *then*

$$\mathrm{E}_K |c_{\mathrm{mult}_K}(u,v)|^2 = \sum_{w \in \mathbb{Z}_{2^n}} |c_f(u,w)c_f(v,w)|^2,$$

where f *is defined as above.*

Proof. Let χ_w denote a character of the additive group $\mathbb{Z}_{2^n}$ such that $w \in \mathbb{Z}_{2^n}$. Using the correlation theorem, we get

$$\begin{aligned} c_{\mathrm{mult}_K}(u,v) &= \sum_{w \in \mathbb{Z}_{2^n}} \chi_w(f(K)) c_f(u,w) c_{f^{-1}}(w,v) \\ &= \sum_{w \in \mathbb{Z}_{2^n}} \chi_w(f(K)) c_f(u,w) \overline{c_f(v,w)} \end{aligned}$$

The theorem follows by following the same steps as with the proof for the expected value of the squared absolute correlation of a linear approximation of an iterative block cipher in Theorem 5.9. □

Although we stated Theorem 6.2 for linear approximations over $\mathbb{Z}_{2^n}$, we could just as well use other groups of the same order in the approximations. For example, by identifying elements of $\mathbb{Z}_{2^n}$ and $\mathbb{Z}_2^n$ using the natural correspondence, we could use such linear approximations that

$u \in \mathbb{Z}_2^n$ and characters of $\mathbb{Z}_2^n$ are applied for the input of the function. Theorem 6.2 also shows that the linearity of constant multiplication modulo $2^n + 1$, where $2^n + 1$ is prime, is significantly tied to the linearity of discrete logarithm in $\mathbb{Z}_{2^n+1}$, which is studied in Section 6.4. The formulation (6.1) for mult_K shows that the multiplication of subkeys in IDEA can be represented using additions modulo 2^n.

6.3 Discrete Logarithm in $\mathbb{F}_{2^n}$

In this section, we study linearity properties of discrete logarithm in $\mathbb{F}_{2^n}$. We consider discrete logarithm as a nonlinear vector-valued Boolean function that could be used as a component in symmetric ciphers. Brandstätter et al. [5] studied the least significant bit of discrete logarithm in $\mathbb{F}_{2^n}$ and showed that it is highly nonlinear. In Publication III, an upper bound for the linearity of a linear combination of the coordinates of the discrete logarithm was derived. For fixed dimension n, this upper bound depends on the length of the masking vector determining the linear combination. In this section, we deduce an upper bound that depends on the Hamming weight of the masking vector. Compared to the bound of Publication III, this upper bound provides better estimates when the masking vector is sparse.

Let n be a positive integer and denote $q = 2^n$. Suppose that α is a primitive element of $\mathbb{F}_{2^n}$. The discrete logarithm $\log_\alpha x$ of $x \in \mathbb{F}_q^*$ to the base α is the integer l such that $0 \le l \le q-2$ and $x = \alpha^l$. We study properties of the function $f\colon \mathbb{F}_q \to \mathbb{Z}_q$ defined as

$$f(x) = \begin{cases} \log_\alpha x & \text{if } x \neq 0, \\ q-1 & \text{if } x = 0. \end{cases} \tag{6.2}$$

Given a vector $v \in \mathbb{F}_2^n$, we will use $v \lll i$ and $v \ggg i$ to denote the left and right cyclic shifts of v by i coordinates, respectively. For $1 \le i \le n$, we let $\delta_i \in \mathbb{Z}_2^n$ denote a unit vector, where the component at index i is one and the rest are zeros. Given a vector $v \in \mathbb{F}_2^n$, the new upper bound for the linearity is based on deriving an upper bound for the sum

$$S(v) = \sum_{w \in \mathbb{Z}_{q-1}} \left| \sum_{x \in \mathbb{Z}_{q-1}} e_2(v \cdot x) e_{2^n-1}(wx) \right|$$

that depends on the Hamming weight of v. For this purpose, we present lemmas that show how the value of $S(v)$ can be estimated using the upper bound for $S(\delta_n)$, which is known to be accurate. The bounds make use of the quantity

$$D(q) = \frac{4}{\pi^2}(q-1)\ln(q-1) + 0.38(q-1) + 0.608 + 0.116\frac{1}{q-1}$$

as in Publication III. We first prove the upper bound for $S(\delta_n)$.

Lemma 6.3. *Let $S(v)$ be defined as before for $v \in \mathbb{F}_2^n$. Then*

$$S(\delta_n) < q - 1 + 2D(q).$$

Proof. According to the orthogonality relation, we have

$$\sum_{r=0}^{m-1} e_m(r(y-z)) = \begin{cases} m & \text{if } y = z \bmod m, \\ 0 & \text{if } y \neq z \bmod m, \end{cases}$$

for $y, z, m \in \mathbb{Z}$ with $m \geq 2$. Let $v \in \mathbb{F}_2^n$ be a vector. The mapping $x \mapsto e_{q-1}(wx)$ is a character of $\mathbb{Z}_{q-1}$ for all $w \in \mathbb{Z}_{q-1}$. We have

$$\begin{aligned} S(v) &= \sum_{w=0}^{q-2} \left| \sum_{x=0}^{q-2} e_2(v \cdot x) e_{q-1}(wx) \right| \\ &= \sum_{w=0}^{q-2} \left| \sum_{v \cdot x = 0} e_{q-1}(wx) - \sum_{v \cdot x = 1} e_{q-1}(wx) \right| \\ &\leq q - 1 + 2 \sum_{w=1}^{q-2} \left| \sum_{v \cdot x = 0} e_{q-1}(wx) \right| \end{aligned}$$

For $v = \delta_n$, it then follows that

$$\begin{aligned} S(\delta_n) &\leq q - 1 + 2 \sum_{w=1}^{q-2} \left| \sum_{x=0}^{q/2-1} e_{q-1}(2wx) \right| \\ &= q - 1 + 2 \sum_{w=1}^{q-2} \left| \frac{1 - e_{q-1}(w)}{1 - e_{q-1}(2w)} \right| \\ &= q - 1 + 2 \sum_{w=1}^{q-2} \left| \frac{\sin(2^{n-1}\pi w/(q-1))}{\sin(\pi w/(q-1))} \right| \\ &< q - 1 + 2D(q), \end{aligned}$$

where the last step follows from the upper bound of Cochrane [8], see the proof of Corollary 2 in Publication III. □

Corollary 6.4. *For $n \geq 5$, we have*

$$\frac{S(\delta_n)}{q-1} < \log q.$$

Proof. The previous lemma yields

$$\frac{S(\delta_n)}{q-1} < \frac{8}{\pi^2}\ln(q-1) + 1.76 + \frac{1.216}{q-1} + \frac{0.232}{(q-1)^2}.$$

It is then straightforward to check that the result holds for $n \geq 5$. □

The following two lemmas show how $S(v)$ can be decomposed for any $v \in \mathbb{F}_2^n$ such that it is possible to use the upper bound for $S(\delta_n)$ to determine a bound for $S(v)$.

Lemma 6.5. *Let $v, v' \in \mathbb{Z}_2^n$ be vectors. Then*

$$S(v \oplus v') \leq \frac{S(v)S(v')}{q-1}.$$

Proof. Let $v, v' \in \mathbb{F}_2^n$ be vectors. We get

$$\begin{aligned}
S(v)S(v') &= \sum_{w=0}^{q-2} \left| \sum_{x=0}^{q-2} e_2(v \cdot x) e_{q-1}(wx) \right| \sum_{w'=0}^{q-2} \left| \sum_{y=0}^{q-2} e_2(v' \cdot y) e_{q-1}(w'y) \right| \\
&= \sum_{w=0}^{q-2} \sum_{w'=0}^{q-2} \left| \sum_{x=0}^{q-2} e_2(v \cdot x) e_{q-1}((w-w')x) \right| \left| \sum_{y=0}^{q-2} e_2(v' \cdot y) e_{q-1}(w'y) \right| \\
&\geq \sum_{w=0}^{q-2} \left| \sum_{x=0}^{q-2} \sum_{y=0}^{q-2} e_2(v \cdot x) e_2(v' \cdot y) e_{q-1}(wx) \sum_{w'=0}^{q-2} e_{q-1}(w'(y-x)) \right| \\
&= (q-1) \sum_{w=0}^{q-2} \left| \sum_{x=0}^{q-2} e_2((v \oplus v') \cdot x) e_{q-1}(wx) \right| \qquad (6.3) \\
&= (q-1) S(v \oplus v'),
\end{aligned}$$

where (6.3) follows from the orthogonality relation. □

Lemma 6.6. *Let $v \in \mathbb{Z}_2^n$ denote a vector. For any integer $i \in \mathbb{Z}$, we have*

$$S(v \ggg i) = S(v).$$

Proof. We have $x \lll i = 2^i x \pmod{2^n - 1}$ for any n-bit integer $x \in \mathbb{Z}_2^n$ and for any integer $i \in \mathbb{Z}$. Since the mapping $v \mapsto 2^i v$ defines a bijection on $\mathbb{Z}_{2^n-1}$, we get

$$\begin{aligned}
S(v) &= \sum_{w \in \mathbb{Z}_{q-1}} \left| \sum_{x \in \mathbb{Z}_{q-1}} e_2(v \cdot x) e_{q-1}(wx) \right| \\
&= \sum_{w \in \mathbb{Z}_{q-1}} \left| \sum_{x \in \mathbb{Z}_{q-1}} e_2(v \cdot (x \lll i)) e_{q-1}(2^i wx) \right| \\
&= \sum_{w \in \mathbb{Z}_{q-1}} \left| \sum_{x \in \mathbb{Z}_{q-1}} e_2((v \ggg i) \cdot x) e_{q-1}(wx) \right| \\
&= S(v \ggg i)
\end{aligned}$$

for all $v \in \mathbb{F}_2^n$. □

Lemma 6.7. *Assume that $n \geq 5$ and let $v \in \mathbb{F}_2^n$ be a vector. Then*

$$\frac{S(v)}{q-1} < (\log q)^{w_{\mathrm{H}}(v)}.$$

Proof. Denote $k = w_{\mathrm{H}}(v)$ and suppose that l_i denotes the index of the ith nonzero component in v. By Lemmas 6.5 and 6.6, we get

$$\frac{S(v)}{q-1} = \frac{S(\delta_{l_1} \oplus \delta_{l_2} \oplus \cdots \oplus \delta_{l_k})}{q-1} \leq \frac{S(\delta_{l_1})S(\delta_{l_2}) \cdots S(\delta_{l_k})}{(q-1)^k} = \left(\frac{S(\delta_n)}{q-1}\right)^k.$$

The result follows by Corollary 6.4. □

Using the previous results, we can deduce the following upper bound for the Fourier coefficients of f. The proof follows the same outline as the proof of Theorem 5 in Publication III. Given a vector $v \in \mathbb{F}_2^n$, we will use $v \cdot f$ to denote the function $x \mapsto v \cdot f(x)$.

Theorem 6.8. *Suppose that $n \geq 5$ and let $v \in \mathbb{F}_2^n$ be a vector. Then*

$$\max_{u \in \mathbb{F}_2^n} |\widehat{v \cdot f}(u)| < 1 + (\log q)^{w_{\mathrm{H}}(v)} \sqrt{q}.$$

Proof. Using the definition of Fourier transform and the orthogonality relation, we get

$$\begin{aligned}
|\widehat{v \cdot f}(u)| &= \left| \sum_{x \in \mathbb{F}_2^n} e_2(v \cdot f(x)) e_2(u \cdot x) \right| \\
&= \left| \sum_{x \in \mathbb{F}_q^*} e_2(v \cdot \log_\alpha x + u \cdot x) + e_2(w_{\mathrm{H}}(v)) \right| \\
&\leq \left| \sum_{x \in \mathbb{F}_q^*} e_2(v \cdot \log_\alpha x + u \cdot x) \right| + 1 \\
&= \frac{1}{q-1} \left| \sum_{w \in \mathbb{Z}_{q-1}} \sum_{x \in \mathbb{F}_q^*} e_2(v \cdot \log_\alpha x) e_{q-1}(w \log_\alpha x) \right. \\
&\qquad \left. \times \sum_{y \in \mathbb{F}_q^*} e_{q-1}(-w \log_\alpha x) e_2(u \cdot x) \right| + 1 \\
&\leq \frac{\sqrt{q}}{q-1} \sum_{w \in \mathbb{Z}_{q-1}} \left| \sum_{x \in \mathbb{F}_q^*} e_2(v \cdot \log_\alpha x) e_{q-1}(w \log_\alpha x) \right| + 1,
\end{aligned}$$

where the last step follows from triangle inequality and the result on Gauss sums. By substituting $z = \log_\alpha x$ and using Lemma 6.7. we get

$$|\widehat{v \cdot f}(u)| \leq 1 + \frac{\sqrt{q}}{q-1} S(v) < 1 + (\log q)^{w_{\mathrm{H}}(v)} \sqrt{q}$$

for $n \geq 5$. □

Thus, we obtain the following upper bound for the linearity of a linear combination of the coordinates of discrete logarithm.

Corollary 6.9. *Assume that $n \geq 5$ and let $v \in \mathbb{F}_2^n$ be a vector. Then*

$$\mathcal{L}(v \cdot f) < 1 + n^{w_{\mathrm{H}}(v)} 2^{n/2}.$$

6.4 Exponential Welch Costas Functions

In this section, we discuss linearity of Exponential Welch Costas (EWC) functions and their inverses, Logarithmic Welch Costas (LWC) functions. The EWC and LWC functions are nonlinear mappings used in SAFER [32, 33]. The mappings are bijections on $\mathbb{Z}_{p-1}$, where p is a prime. A lower bound on the linearity of EWC and LWC functions was derived in Publication II. According to Theorem 5.6, the linearity of a bijection and its inverse is the same, so the same bound applies for both functions.

We use p to denote an odd prime, and α to denote a generator of the multiplicative group $\mathbb{Z}_p^*$. The exponential function of $\mathbb{Z}_p$ is a mapping from $\mathbb{Z}_{p-1}$ to $\mathbb{Z}_p^*$ defined as $x \mapsto \alpha^x \bmod p$. We consider $\mathbb{Z}_{p-1}$ to be the set $\{0, 1, \ldots, p-2\}$ and $\mathbb{Z}_p^*$ to be the set $\mathbb{Z}_p \setminus \{0\} = \{1, 2, \ldots, p-1\}$.

Definition 6.10. An Exponential Welch Costas function is the mapping $f \colon \mathbb{Z}_{p-1} \to \mathbb{Z}_{p-1}$ defined as

$$f(x) = (\alpha^x \bmod p) - 1.$$

Its inverse function $f^{-1}(x) = \log_\alpha(x+1)$ is called a Logarithmic Welch Costas function.

We first present an alternative proof for Lemma 1 of Drakakis et al. [13] or Lemma 1 in Publication II.

Lemma 6.11. *For any $u \in \mathbb{Z}_{p-1}$ and $v \in \mathbb{Z}_p^*$, we have*

$$\left| \sum_{x=0}^{p-2} e_p(vf(x)) e_{p-1}(ux) \right| \le \sqrt{p}.$$

Proof. Viewing the exponential function as mapping to $\mathbb{Z}_p$, we can define its Fourier transform as

$$\begin{aligned} S(u,v) &= \sum_{x=0}^{p-2} e_p(vf(x)) e_{p-1}(ux) = e_p(-v) \sum_{x=0}^{p-2} e_p(v\alpha^x) e_{p-1}(ux) \\ &= e_p(-v) \sum_{x=1}^{p-1} e_p(vx) e_{p-1}(u \log_\alpha x). \end{aligned}$$

The function $\psi_u(x) = e_{p-1}(u \log_\alpha x)$ defines a character of the multiplicative group $\mathbb{Z}_p^*$ and $\chi_v(x) = e_p(vx)$ defines a character of the additive group $\mathbb{Z}_p$. By the definition of Gauss sum, we obtain

$$S(u,v) = e_p(-v) G(\psi_u, \chi_v),$$

where $G(\psi_u, \chi_v)$ denotes the Gauss sum with the multiplicative character ψ_u and the additive character χ_v. The result follows by Theorem 3.7. □

The proof also shows how the linearity of EWC and LWC functions is related to the value of Gauss sums. If the output domain of f is treated as $\mathbb{Z}_p$ as opposed to $\mathbb{Z}_{p-1}$, we have

$$|\widehat{\psi_v \circ f}(u)| = |G(\psi_u, \chi_v)|.$$

Thus, we obtain the following value for the linearity.

Theorem 6.12 (13)**.** *Let $f\colon \mathbb{Z}_{p-1} \to \mathbb{Z}_p$ be defined as an EWC function. Then*

$$\mathcal{L}(f) = \sqrt{p}.$$

As indicated by Theorem 5.7, the lower bound for the linearity is $\sqrt{p-1}$, which is almost achieved by EWC functions. If the output domain is chosen as in the definition, the situation is not so straightforward as is shown in Publication II. However, the bound in Publication II implies that also in this case the EWC functions have asymptotically high nonlinearity.

Theorem 6.13. *Let $f\colon \mathbb{Z}_{p-1} \to \mathbb{Z}_{p-1}$ be an EWC function. Then*

$$\mathcal{L}(f) < \frac{2}{\pi}\sqrt{p}\ln p + 4\sqrt{p}.$$

7. Side-Channel Analysis

In this chapter, we discuss cache-timing attacks and present a technique for automated analysis of side-channel data. We also briefly describe the cache-timing attacks presented in Publication V and Publication VI. Automated data analysis is critical if the attack involves processing of large amounts of data obtained from many executions of a particular process, such as an algorithm. Our technique for data analysis is based on modeling the environment using hidden Markov models (HMMs) and vector quantization (VQ). An application of the technique is shown in Publication V, where automated data analysis is performed for cache-timing data obtained from ECDSA scalar multiplication in OpenSSL. The resulting state information is then used to recover the secret key. Publication VI presents a cache-timing attack against the SNOW 3G stream cipher, where this kind of data processing is not needed.

7.1 Cache-Timing Attacks

Cache-timing attacks are based on measuring the latency it takes for the CPU to access data while the cryptographic algorithm is running. A CPU has a limited number of working registers to store data. Modern processors are equipped with a data cache to offset the high latency of loading data from main memory into these registers. When the CPU needs to access data, it first looks in the data cache, which is faster but with smaller capacity than main memory. If it finds the data in the cache, it is loaded with minimal latency and this is known as a cache hit; otherwise, a cache miss occurs and the latency is higher as the data is fetched from successive layers of caches or even main memory. Thus access to frequently used data has lower latency. Cache layers L1, L2, and L3 are commonplace, increasing with capacity and latency. Our focus is on data caches, but processors often have an instruction cache as well.

Cryptographic algorithms can be vulnerable to cache-timing attacks if they rely on table lookups that depend on some secret value, e.g., the secret key. These table lookups may be visible in cache-timings and thus leak critical information about the secret value. An attacker can detect this by concurrently running a spy process [40] that does nothing more than continually load its own data into the cache and measure the time required to read it. Fast cache access times indicate cache hits, which means that the cryptographic algorithm has not accessed those cache locations since the last iteration, because that would evict the spy process data, cause a cache miss, and thus slower cache access times for the spy process.

Simultaneous multithreading technology in processors allows active execution of multiple threads concurrently. In a cache-timing attack scenario, this relaxes the need to force context switches since the threads naturally compete for shared resources during execution, such as the data caches.

7.2 Automated Side-Channel Data Analysis

Recall that an HMM can be seen as a dynamical system which behaves like a Markov chain that has directly unobservable states. Information about the states can only be obtained through the emitted observations. An HMM is represented as a tuple

$$\lambda = (A, B, \pi),$$

where A denotes the set of transition probabilities between the states, B denotes the set of emission probabilities, and π denotes the initial state distribution. The unobservable part of the HMM can be seen as a probabilistic linear system since its state transitions can be modeled using the transition matrix of the Markov chain.

HMMs provide a natural way to model side-channel scenarios: the target system is seen as the hidden part of the HMM and the emitted observations are seen as the information leaked through the side channel. Using the techniques mentioned previously, side-channel data analysis can be performed using HMMs and VQ. The analysis process can be divided into two steps: We first create the HMM based on system specifications and adjust its parameters using a set of observation sequences obtained from the side channel. When a new sequence of observations arrives, we infer the most likely state sequence that has emitted these observations using VQ and the HMM. Details of this process are presented in the following sections. We assume that the attacker can modify the system in the first phase in such a manner that its behavior can be observed.

Building the HMM. The HMM should be constructed such that its hidden part reflects the operation of the system and the set of observables reflect the observations obtained from the side channel. In practice, one internal state of the real system can emit many observations which is not allowed by HMMs. To overcome this problem, one can use a sequence of states in the HMM to model one main state in the system. The set of observable symbols should be the same as the set of labels for VQ. When a new side-channel observation is obtained, it can then be mapped to an HMM observable using VQ.

Training the HMM. To train the HMM, we first initialize the model parameters with rough estimates which will be improved during the training process. We then obtain a set of observation sequences from the side channel by running the system with different parameters and observing how the system operates. Since we know how the system operates, we can label the obtained sequences in the HMM domain. The model parameters can then be adjusted by running the Baum–Welch algorithm using the observation sequences as the training data set. The VQ codebook is created using LVQ.

Inference of the State Sequence. The constructed model can now be used to infer the behavior of the target system in the following way. When we obtain a new observation sequence from the side channel, we tag each observation with the label of the closest VQ codebook vector. Thus, we get a sequence in the HMM observation domain. Using the Viterbi algorithm with the sequence, we can then infer the most likely state sequence that emitted the observations. The obtained state sequence is actually a sequence of substates; the actual operation sequence can be recovered based on the transitions that are taken in each state sequence. The obtained state sequences can then be used in conjunction with other methods to break the system.

7.3 Applications

7.3.1 OpenSSL ECDSA

The presented technique for automated side-channel data analysis is applied in Publication V for cache-timing data obtained by performing the ECDSA signature operation in OpenSSL. The scalar multiplication in the OpenSSL ECDSA implementation uses table lookups that depend on the nonce used when creating a signature. These table lookups are visible in the cache-timing data revealing significant amount of information about the nonce. Given enough such information, the private key can then be recovered using a lattice attack.

The model used to analyze obtained cache-timing data is constructed in the following way. The hidden part of the HMM reflects operation of the double-and-add scalar multiplication. An abstraction of it is depicted in Fig. 3 of Publication V. In the real HMM, each depicted state consists of four or five substates. The set of observations for the HMM is $\{D, A, E\}$. It is assumed that doublings mainly emit Ds and additions As. The start and end states are assumed to mainly emit Es. In the analysis phase, VQ is used to tag real cache-timing data with these labels. The most likely operation sequence can then be obtained using the HMM. A portion of nonce bits is revealed by the operation sequence.

7.3.2 SNOW 3G

Publication VI presents a cache-timing attack against SNOW 3G, which is an LFSR-based stream cipher used to preserve confidentiality and integrity in 3GPP networks. Leander et al. [26] presented a framework for cache-timing attacks on LFSR-based stream ciphers that use table lookups when the LFSR is clocked. They noted that many stream ciphers have an LFSR update function, where constant multiplications have been implemented using table lookups involving shift register bits, and cache-timing measurements reveal some of those bits. Since the LFSR feedback relation is linear, each observed bit can be expressed as a linear combination of the bits in the initial state. Once enough bits have been observed, the initial state can thus be recovered by solving the equation system.

SNOW 3G uses an LFSR and a finite state machine (FSM) to produce the output keystream. Our attack on SNOW 3G also uses the information obtained from the table lookups in the LFSR update function, but most importantly, it uses information obtained from the table lookups used by the two consecutive S-boxes, $S1$ and $S2$, in the FSM. Due to this structure, information about the input and output of $S1$ is revealed, which makes it possible to determine a small set of candidates for the inputs of $S1$. The state recovery algorithm obtains a set of candidate values for the cipher state at certain time instances based on side-channel information and then trims incorrect candidates using the LFSR feedback relation and a relation involving LFSR and FMS registers. The resulting attack is capable of recovering the cipher state in a matter of seconds, requiring no known keystream and using only a small number of cipher clocks. While the attack makes use of the information obtained from the LFSR table lookups, it also works only with the FSM lookups and does not rely on the linearity of the shift register feedback relation. The attack was shown to work with experimental side-channel data under an attack model, where the attacker can control the rate at which the cipher is clocked.

8. Conclusions

In this thesis, we studied how linearity can be exploited in cryptanalysis and how it affects the design of cryptographic primitives. We obtained results concerning both formal specifications and real-life implementations of cryptographic primitives. Related to formal cryptanalysis, we examined symmetric ciphers and their building blocks using linear cryptanalysis. Related to implementation-specific analysis, we presented a framework for automated analysis of side-channel data. Real primitive designs were studied in both areas of cryptanalysis.

Extending previous research on linear cryptanalysis, we presented a framework for (multidimensional) linear distinguishing attacks over finite Abelian groups. The presented techniques were employed in particular in the analysis of the block cipher DEAN, but the attack on the stream cipher Shannon can also be viewed within this framework. The analysis of DEAN shows that relevant security bounds against linear attacks cannot be obtained without considering the linear hull effect and multidimensional analysis. We presented a general purpose algorithm that can be used for obtaining more accurate security guaranties against linear cryptanalysis. The linear distinguishing attack on Shannon is based on low nonlinearity of the used S-boxes, but also on a certain structural weakness that makes the cipher operate in a more linear manner.

The results in this thesis give evidence that discrete logarithm can be used to create functions that have high nonlinearity in different domains. We studied linear combinations of the coordinates of discrete logarithm in $\mathbb{F}_{2^n}$ and derived lower bounds for their linearity. Although the bounds are meaningful only for certain linear combinations, the methods give some new insight to the linearity of discrete logarithm in $\mathbb{F}_{2^n}$. Determining a more accurate bound for it remains an interesting open problem. Exponential Welch Costas functions were shown to have high nonlinearity.

We presented efficient cache-timings attacks on OpenSSL's ECDSA implementation and SNOW 3G. The attack on the OpenSSL ECDSA implementation shows that the data analysis framework can be used to significantly facilitate side-channel attacks. The results on both cache-timing attacks prove that simply by obtaining timing information from table lookups, it is possible to create very powerful attacks on implementations of cryptographic primitives.

A common theme in the presented linear analysis of symmetric ciphers and individual functions, was to employ nontraditional domains in linear approximations. One possible research topic in this direction is to study combinations of different domains in linear cryptanalysis. Another area of research is to continue to study linearities of functions using the presented methods.

Bibliography

[1] T. Baignères, P. Junod, and S. Vaudenay. How far can we go beyond linear cryptanalysis. In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004.

[2] T. Baignères, J. Stern, and S. Vaudenay. Linear cryptanalysis of non binary ciphers. In C. M. Adams, A. Miri, and M. J. Wiener, editors, *SAC 2007*, volume 4876 of *Lecture Notes in Computer Science*, pages 184–211. Springer, 2007.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41(1):164–171, 1970.

[4] A. Biryukov, C. D. Cannière, and M. Quisquater. On multiple linear approximations. In M. K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2004.

[5] N. Brandstätter, T. Lange, and A. Winterhof. On the non-linearity and sparsity of Boolean functions related to the discrete logarithm in finite fields of characteristic two. In Ø. Ytrehus, editor, *WCC 2005*, volume 3969 of *Lecture Notes in Computer Science*, pages 135–143. Springer, 2006.

[6] C. Carlet and K. Feng. An infinite class of balanced vectorial Boolean functions with optimum algebraic immunity and good nonlinearity. In Y. M. Chee, C. Li, S. Ling, H. Wang, and C. Xing, editors, *IWCC 2009*, volume 5557 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2009.

[7] J. Y. Cho. Linear cryptanalysis of reduced-round PRESENT. In J. Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 302–317. Springer, 2010.

[8] T. Cochrane. On a trigonometric inequality of Vinogradov. *Journal of Number Theory*, 27(1):9–16, 1987.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2nd edition, 2006.

[10] H. Cramér and H. Wold. Some theorems on distribution functions. *Journal of the London Mathematical Society*, 11(4):290–294, 1936.

[11] J. Daemen, R. Govaerts, and J. Vandewalle. Correlation matrices. In B. Preneel, editor, *FSE 1994*, volume 1008 of *Lecture Notes in Computer Science*, pages 275–285. Springer, 1994.

[12] K. Drakakis, R. Gow, and G. McGuire. APN permutations on $\mathbb{Z}_n$ and Costas arrays. *Discrete Applied Mathematics*, 157(15):3320–3326, 2009.

[13] K. Drakakis, V. Requena, and G. McGuire. On the nonlinearity of Exponential Welch Costas functions. *IEEE Transactions on Information Theory*, 56 (3):1230–1238, 2010.

[14] R. Durrett. *Essentials of Stochastic Processes*. Springer Texts in Statistics. Springer, 1999.

[15] H. Englund and A. Maximov. Attack the Dragon. In S. Maitra, C. E. V. Madhavan, and R. Venkatesan, editors, *INDOCRYPT 2005*, volume 3797 of *Lecture Notes in Computer Science*, pages 130–142. Springer, 2005.

[16] K. Feng, Q. Liao, and J. Yang. Maximal values of generalized algebraic immunity. *Designs, Codes and Cryptography*, 50(2):243–252, 2009.

[17] G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[18] P. Hawkes, C. McDonald, M. Paddon, G. G. Rose, and M. Wiggers de Vries. Design and primitive specification for Shannon. Technical report, Qualcomm Australia, 2007.

[19] M. Hermelin, J. Y. Cho, and K. Nyberg. Multidimensional linear cryptanalysis of reduced round Serpent. In Y. Mu, W. Susilo, and J. Seberry, editors, *ACISP 2008*, volume 5107 of *Lecture Notes in Computer Science*, pages 203–215. Springer, 2008.

[20] B. S. Kaliski, Jr. and M. J. B. Robshaw. Linear cryptanalysis using multiple approximations. In Y. Desmedt, editor, *CRYPTO 1994*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39. Springer, 1994.

[21] L. R. Knudsen. Contemporary block ciphers. In I. Damgård, editor, *Lectures on Data Security: Modern Cryptology in Theory and Practice*, volume 1561 of *Lecture Notes in Computer Science*, pages 105–126. Springer, 1999.

[22] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. J. Wiener, editor, *CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[23] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, 3nd edition, 2001.

[24] P. V. Kumar, R. A. Scholtz, and L. R. Welch. Generalized bent functions and their properties. *Journal of Combinatorial Theory, Series A*, 40(1):90–107, 1985.

[25] X. Lai and J. L. Massey. A proposal for a new block encryption standard. In I. Damgård, editor, *EUROCRYPT 1990*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 1991.

[26] G. Leander, E. Zenner, and P. Hawkes. Cache timing analysis of LFSR-based stream ciphers. In M. G. Parker, editor, *Cryptography and Coding*, volume 5921 of *Lecture Notes in Computer Science*, pages 433–445. Springer, 2009.

[27] S. Levinson, L. Rabiner, and M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal*, 62(4):1035–1074, 1983.

[28] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2nd edition, 1997.

[29] R. Lidl and G. Pilz. *Applied Abstract Algebra*. Undergraduate Texts in Mathematics. Springer, 2nd edition, 1997.

[30] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[31] O. A. Logachev, A. A. Salnikov, and V. V. Yashchenko. Bent functions on a finite Abelian group. *Discrete Mathematics and Applications*, 7(6):547–564, 1997.

[32] J. L. Massey. SAFER K-64: A byte-oriented block-ciphering algorithm. In R. J. Anderson, editor, *FSE 1993*, volume 809 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1994.

[33] J. L. Massey. SAFER K-64: One year later. In B. Preneel, editor, *FSE 1994*, volume 1008 of *Lecture Notes in Computer Science*, pages 212–241. Springer, 1995.

[34] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1994.

[35] M. Matsui and A. Yamagishi. A new method for known plaintext attack of FEAL cipher. In R. A. Rueppel, editor, *EUROCRYPT 1992*, volume 658 of *Lecture Notes in Computer Science*, pages 81–91. Springer, 1993.

[36] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[37] K. Nyberg. Linear approximation of block ciphers. In A. D. Santis, editor, *EUROCRYPT 1994*, volume 950 of *Lecture Notes in Computer Science*, pages 439–444. Springer, 1995.

[38] K. Nyberg. Accurate data complexity estimates for χ^2 distinguishers. Presentation at Dagstuhl Seminar 12031 “Symmetric Cryptography”, 2012.

[39] K. Nyberg and J. Wallén. Improved linear distinguishers for SNOW 2.0. In M. J. B. Robshaw, editor, *FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 144–162. Springer, 2006.

[40] C. Percival. Cache missing for fun and profit. In *Proc. of BSDCan*, 2005.

[41] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

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

[43] S. Vaudenay. An experiment on DES statistical cryptanalysis. In L. Gong and J. Stern, editors, *CCS 1996*, pages 139–147. ACM Press, 1996.

[44] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[45] J. Wallén. Linear approximations of addition modulo 2^n. In T. Johansson, editor, *FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 261–273. Springer, 2003.

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

Aalto-DD45/2012	Viitaniemi, Ville Visual Category Detection: an Experimental Perspective. 2012.
Aalto-DD51/2012	Hanhijärvi, Sami Multiple Hypothesis Testing in Data Mining. 2012.
Aalto-DD56/2012	Ramkumar, Pavan Advances in Modeling and Characterization of Human Neuromagnetic Oscillations. 2012.
Aalto-DD97/2012	Turunen, Ville T. Morph-Based Speech Retrieval: Indexing Methods and Evaluations of Unsupervised Morphological Analysis. 2012.
Aalto-DD115/2012	Vierinen, Juha On statistical theory of radar measurements. 2012.
Aalto-DD117/2012	Huopaniemi, Ilkka Multivariate Multi-Way Modelling of Multiple High-Dimensional Data Sources. 2012.
Aalto-DD137/2012	Paukkeri, Mari-Sanna Language- and domain-independent text mining. 2012.
Aalto-DD133/2012	Ahlroth, Lauri Online Algorithms in Resource Management and Constraint Satisfaction. 2012.
Aalto-DD158/2012	Virpioja, Sami Learning Constructions of Natural Language: Statistical Models and Evaluations. 2012.
Aalto-DD20/2013	Pajarinen, Joni Planning under uncertainty for large-scale problems with applications to wireless networking. 2013.

Aalto-DD 29/2013

A mathematical system is generally easier to understand if it exhibits linear rather than nonlinear behavior. Indeed, many cryptanalytic techniques take advantage of linear properties of cryptosystems. In this thesis, several cryptanalytic results related to such techniques are presented. The thesis includes analysis of real symmetric-key and public-key primitives, general cipher constructions and also their building blocks. The presented results have impact on the design of both formal specifications and real-life implementations of cryptographic primitives.

9 789526 050287

ISBN 978-952-60-5028-7
ISBN 978-952-60-5027-0 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi