Correlations Aplenty - Linear Cryptanalysis of Block Ciphers

Vejre, Philip Søgaard

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Vejre, P. S. (2018). Correlations Aplenty - Linear Cryptanalysis of Block Ciphers. Kgs. Lyngby: DTU Compute. DTU Compute PHD-2018, Vol.. 497

Correlations Aplenty

Linear Cryptanalysis of Block Ciphers

Philip S. Vejre

Ph.D. Thesis

October 2018

Document compiled on October 16, 2018.

Supervisor:	Andrey Bogdanov
Co-supervisor:	Lars R. Knudsen

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Section for Cyber Security

ISSN:	0909-3192
Serial no.:	PHD-2018-497

Deciphering is, in my opinion, one of the most fascinating of arts,
and I fear I have wasted upon it more time than it deserves.

– Charles Babbage

Abstract

The advent of the Digital Age has brought upon us a world where information is a primary commodity. Thanks to the near omnipresence of computing devices, the collection and exchange of information is easier and more frequent than ever before. Soon, almost all electrically powered devices will contain a computer, and furthermore they will all be communicating with each other. The consequence is that a wealth of information about each of us is being recorded and transmitted around the world – information that most people will likely prefer to keep as private as possible. Thus, the trends of the modern age also bring with them an increasing focus on – and importance of – *cryptography* in order to provide secure communications.

A core research area of cryptography is the construction of *secure block ciphers* – a so-called cryptographic primitive, their importance arise from the fact that a plethora of other cryptographic algorithms can be constructed from a block cipher. Clearly, it is crucial to have a high level of confidence in the security of such a building block. However, it is not known how to construct a block cipher which can be proven secure, and so instead, the security is evaluated by trying to mount every known attack against the cipher. A deep understanding of the different attack techniques is therefore essential in order to get an accurate assessment of a block cipher's strength.

This thesis explores one such attack technique: *linear cryptanalysis*. Being a central technique in the cryptanalyst's tool kit, every new block cipher has to demonstrate resistance against this attack. Nevertheless, our understanding of this statistical attack is not complete, especially so for advanced variants where the stochastic variables we need to analyse are quite complex. Therefore, the following work is part of an effort to build advanced tools and models with the aim of providing more accurate analysis of advanced linear attacks.

The first part of this thesis introduces block ciphers and notions of their security, followed by an introduction to linear cryptanalysis. The second part contains four publications that advance the field of linear cryptanalysis in several ways. They present new tools that help the cryptanalyst discover new linear attacks, and facilitated by these tools, new statistical models are presented. These models aim to remove many simplifying assumptions that have previously been made when evaluating linear attacks. We present new attacks on the block ciphers DES and PRESENT using these models, and the statistical behaviour of linear attacks is examined for a number of other block ciphers. It is shown that the type of probability distribution involved in a linear attack can vary wildly between ciphers, demonstrating that when we evaluate the effectiveness of such an attack, great care must be taken. Thus, while the work of this thesis does advance the frontier of linear cryptanalysis, it also shows that there is much unknown land yet to discover.

Resumé

Den digitale tidsalder har uden tvivl haft stor indflydelse verden over, og har medvirket, at information er en eftertragtet handelsvare. Takket være computerens allestedsnærværelse er indsamling og udveksling af information lettere og mere hyppig end nogensinde før. Inden længe vil næsten alle elektroniske apparater indeholde en computer, og de vil ydermere alle kommunikere med hinanden. Resultatet er, at store mængder information bliver indsamlet om os alle og sendt verden rundt – information som de fleste sandsynligvis ville foretrække var privat. Den moderne tidsalder har således forårsaget en stigende interesse i – og fokus på vigtigheden af – *kryptologi* som led i sikring af vores indbyrdes kommunikation.

Et vigtigt forskningsområde inden for kryptologi er konstruktionen af *sikre block ciphers* – et såkaldt kryptografisk primitiv hvis vigtighed stammer fra det faktum, at de kan bruges til at konstruere et væld af andre kryptografiske algoritmer. Det er tydeligt, at det er afgørende at have stor tillid til sådan en byggeklods. Der er imidlertid ingen måde, hvorpå vi kan konstruere en block cipher, som kan bevises sikker, og sikkerheden evalueres derfor ved at forsøge at angribe cipheren med alle kendte midler. Det er derfor vigtigt at have en dyb forståelse af de forskellige angrebsteknikker for nøjagtigt at kunne vurdere en block ciphers styrke.

Denne afhandling fokuserer på én sådan angrebsteknik: *lineær kryptoanalyse*. Da dette er et primært redskab i kryptoanalytikerens værktøjskasse, er det påkrævet at enhver ny block cipher kan modstå et sådan angreb. På trods af dette er vores forståelse af denne type statistiske angreb ikke fuldkommen, især ikke når det gælder avancerede varianter, hvor de stokastiske variable, der er involveret, er meget komplekse. Denne afhandling er derfor en del af et forsøg på at udvikle avancerede værktøjer og modeller for at opnå mere præcis analyse af avancerede lineære angreb.

Den første del af denne afhandling introducerer block ciphers og relaterede sikkerhedsbegreber, efterfulgt af en introduktion til lineær kryptoanalyse. Den anden del indeholder fire udgivelser, der fremmer lineær kryptoanalyse på forskellige måder. De præsenterer nye værktøjer, der hjælper kryptoanalytikeren til at finde nye lineære angreb, og ved hjælp af disse værktøjer gives også nye statistiske modeller. Disse modeller har til formål at fjerne mange forenklende antagelser, der tidligere har været anvendt. Ved brug af disse modeller præsenterer vi nye angreb på to block ciphers, DES og PRESENT, og vi undersøger lineære angreb på flere andre ciphers med fokus på deres statistiske egenskaber. Vi demonstrerer, at den type sandsynlighedsfordeling, der indgår i et lineært angreb, kan variere voldsomt mellem ciphers, og at man derfor skal være påpasselig når man analysere disse angreb. Resultaterne præsenteret i denne afhandling flytter således grænsen for vores viden om lineær kryptoanalyse, men de viser også, at der endnu er meget vi ikke forstår.

Acknowledgements

Doing a Ph.D. – and I think most people who have done it will agree – is not a trivial task. It takes time, effort, dedication, and perhaps a measure of stubbornness. More importantly, it takes a network of amazing people to inspire you, help you, encourage you, and lead you in the right direction.

A heartfelt thank you goes to my supervisor, Andrey Bogdanov, for taking me under his wing. He has taught me many things about research, academia, and otherwise, and has guided my career at DTU since long before my Ph.D. studies. I commend him for always having an open door and time to discuss whatever I needed to get off my chest. To my co-supervisor and head of the research group during most of my time at DTU, Lars R. Knudsen, I also extend my gratitude for providing an excellent work environment and a sense of belonging.

Likewise, I am grateful for my current and former co-workers at the Section for Cyber Security: Andrey Bogdanov, Christian D. Jensen, Andreas Kidmose, Lars R. Knudsen, Stefan Kölbl, Weizhi Meng, Christian Probst, Christian Rechberger, Arnab Roy, Rishikesh Sahay, Shizra Sultan, Tyge Tiessen, Elmar Tischhauser, and Tine Topsøe, as well as our always helpful secretary Ann-Cathrin Dunker. A special thanks to Stefan and Tyge, without who my coffee consumption would have been lower, my productivity higher, and my time as a Ph.D. student much more dreary. In the same vein, thank you to Thomas Perrett, Martin Merker, Kasper Lyngsie, Liang Zhong, Sophie Birot, and Eva Rotenberg for many delightful hours of shared coffee, beer, and helpful discussion.

Thank you to my co-authors and collaborators, Andrey Bogdanov, Mathias Hall-Andersen, Martin Jepsen, Stefan Kölbl, Matthieu Rivain, Elmar Tischhauser, and Junwei Wang, as well as anyone else from the crypto community that I have had the pleasure of sharing coffee or beer with during conferences and summer schools.

To my amazing friends outside of DTU, Nikolaj, Jacob, Nina, and especially the guys and girls of KoCtG (you know who you are): the last three years have been immeasurably better because of you, and for that I am eternally grateful. A special thanks to Nina for helping me proofread this manuscript and always having a helpful comment regarding graphic design.

Last, but in no way least, a giant thank you to my parents: for their support, love, hospitality, and seemingly endless supply of amazing wine and food. Truly, non of this would have happened without you.

— Nærum, 15th October, 2018

Contents

Part I

Block Ciphers and Linear Cryptanalysis

1 Introduction

Cryptography was born out of a desire for humans to communicate securely with each other, but the field has evolved greatly since Julius Caesar used simple encryption to communicate with his generals. Today, the facets of cryptography are numerous, and many of them are ubiquitous in most people's lives, often times without them even realising. Indeed, while most people have learned to feel a sense of security when the green lock icon shows up in their browser during online purchases, few of them grasp the large number of moving cryptographic parts required to establish a TLS connection, even though as much as 75% of all web pages visited are protected by this protocol [53].

Certainly, humanity's reliance on secure communications will only increase with time. The expanding interest in the Internet of Things definitely illustrates this. An ever growing number of "smart devices" are connected to the Internet in order to automate, coordinate, and optimise our lives. Phones, cars, watches, fridges, alarm clocks, even lamps, already are or will soon communicate with each other and with the rest of the world. In fact, it was estimated that already in 2008 there were more devices connected to the Internet than people [54]. Clearly, all the information that these devices record and transmit has to be protected, lest we completely forfeit any dream of privacy.

But modern cryptography is used for so much more than just secure communication between Alice and Bob. Encryption is also used for secure storage of data at rest. Digital signatures make it possible for a receiver of a message to verify the identify of the sender. Advanced algorithms for fully homomorphic encryption allow for example medical institutions to derive statistics from encrypted medical data, ensuring patient privacy and confidentiality. Similarly, protocols for secure multiparty computation allow several people to compute functions of private data, without revealing this data to each other. Cryptographic proof-of-work algorithms have been widely used in the recent development of various block chain protocols. The list of applications for cryptography is long and keeps growing.

While the motivation for developing cryptographic algorithms and protocols is clear, achieving these goals is a non-trivial task. Ideally, we would like to have some sort of guarantee that the algorithms we use are secure, but proving such security can be extremely difficult for large complicated systems. Thus, it is common wisdom in the field to start with a few simple and secure building blocks, and then build more advanced systems from these components. The core idea here is that if we trust the building blocks to be secure, we can often prove that the bigger system also is. We call such building blocks *cryptographic primitives*.

One such primitive is the *block cipher*. This type of algorithm provides very basic encryption functionality, yet many other cryptographic systems can be built from such a function. Thus, design and analysis of block ciphers are core topics of cryptographic research. Interestingly, we do not know how to construct block ciphers that can be proven secure. Instead, a trial by fire approach is used, in which the cipher is subjected to every known attack in the cryptanalyst's arsenal. If the cipher manages to survive this scrutiny, it is assumed to be sufficiently secure for use as a building block.

For the above approach to work, we need to develop a deep understanding of each different attack technique. Therefore, this thesis thoroughly investigates one particular technique, namely *linear cryptanalysis*. Chapter 2 gives an introduction to block ciphers and briefly presents a range of different attacks against these primitives. Chapter 3 introduces the fundamentals of linear cryptanalysis as well as some advanced variants of this attack. Finally, Chapter 4 gives a summary of the four publications presented in Part II of this thesis. These four publications aim to further our understanding of linear cryptanalysis. They do so by both presenting new tools that aid with analysis of block ciphers in this regard, as well as proposing new models for analysis and evaluation of linear attacks.

2 Block Ciphers and Their Cryptanalysis

In this chapter we consider a fundamental cryptographic primitive, the *block cipher*. After defining what a block cipher is, we describe various aspects of the security of such a cipher and give some examples of its applications. We then consider some general ways to construct a practical block cipher. We finish by explaining some of the more prominent cryptanalysis techniques used against block ciphers.

2.1 The Block Cipher

Informally, a block cipher is a function which, given a fixed length *key*, transforms (*encrypts*) an input string of fixed length (the *plaintext*) to an output string of the same length (the *ciphertext*), such that for each choice of the key, the transformation is a permutation [69]. As a consequence, the transformation is invertible (*decryption*). A formal definition follows.

Definition 2.1 (Block Cipher [81, Chapter 7])**.** Let $\mathbb{F}_2^n$ be the space of vectors of length n over the field of two elements, $\mathbb{F}_2$, and likewise let $\mathbb{F}_2^\kappa$ be the space of vectors of length κ over $\mathbb{F}_2$. A *block cipher* is a function

$$\mathcal{E}(x,k) : \mathbb{F}_2^n \times \mathbb{F}_2^\kappa \to \mathbb{F}_2^n,$$

such that for every choice of the key $k \in \mathbb{F}_2^\kappa$, $\mathcal{E}_k(x) := \mathcal{E}(x,k)$ is a permutation on $\mathbb{F}_2^n$. Moreover, for each k, we denote the inverse of $\mathcal{E}_k(x)$ by $\mathcal{E}_k^{-1}(x)$.

The concept is illustrated in Figure 2.1. Typical values of n used in practice are 64 and 128, while κ is usually 128 or 256. Ultimately, we would like to use a block cipher as a building block for other cryptographic primitives. However, not every construction that fits Definition 2.1 is particularly interesting or useful for cryptographic purposes, or even efficiently computable, limiting practical use. Indeed, we are mainly interested in the design and analysis of practical and *secure* block ciphers.

Inherent Properties In order to understand what we expect from a secure block cipher, and what constitutes a valid attack on such a cipher, we first point out some inherent properties which are consequences of Definition 2.1.

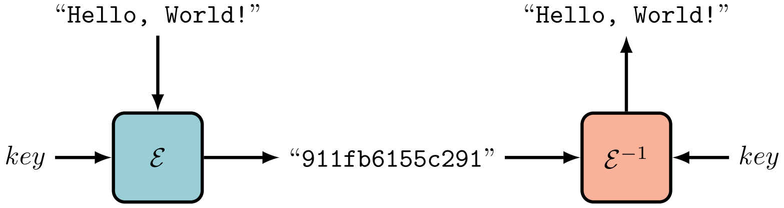


Figure 2.1: Illustration of a block cipher. A plaintext block of thirteen characters is encrypted to a ciphertext block of the same length. The same key is used for encryption and decryption.

Assume that we are given *black box* access to an instance of a block cipher, that is, we can query $\mathcal{E}_k$, for some unknown k, with inputs and get the corresponding output. Then, given a ciphertext y it is always possible to find an input x such that $\mathcal{E}_k(x) = y$ by querying the black box at most 2^n times. Indeed, it is possible to get a complete description of $\mathcal{E}_k$ by storing the answer to all 2^n queries, although this requires 2^n storage as well. Similarly, if we are given x and $y = \mathcal{E}_k(x)$, we can find some k such that this relation holds by using at most 2^κ queries. Note that we will likely need more than one plaintext-ciphertext pair to find a unique value of k [95].

We call the above approach to inverting $\mathcal{E}_k$, or finding the key k, *brute force search*. We consider a block cipher *insecure* if some method exists whereby we can e.g. find the key faster than the equivalent brute force search. We will go into more details about these methods, called *attacks*, in the following. First, however, it will be useful to introduce the notion of an *ideal cipher*.

Definition 2.2 (Ideal Block Cipher)**.** An *ideal block cipher* is a block cipher such that for each key, the permutation is drawn uniformly at random from the space of all permutations on $\mathbb{F}_2^n$.

Constructing such a cipher is clearly infeasible even for small values of n, but we would like a secure block cipher to look like an ideal cipher from the perspective of the adversary.

General Attack Goals Informally, we expect that if the key k is drawn uniformly at random, the resulting permutation $\mathcal{E}_k$ will look like a randomly drawn permutation to an adversary who knows the description of the cipher, *but does not know the key* (a concept commonly known as Kerckhoffs's principle [66]). More specifically, we give an adversary black box access to the block cipher, using one or more unknown keys, and the security of the cipher is then judged by whether a set of general attacks can be mounted against it [69]:

- **Deduction**: Given y, the adversary tries to find x such that $\mathcal{E}_k(x) = y$, or vice versa, in time less than 2^n, e.g. by finding an algorithm $\mathcal{E}'$ which is functionally equivalent to $\mathcal{E}_k$ or $\mathcal{E}_k^{-1}$.

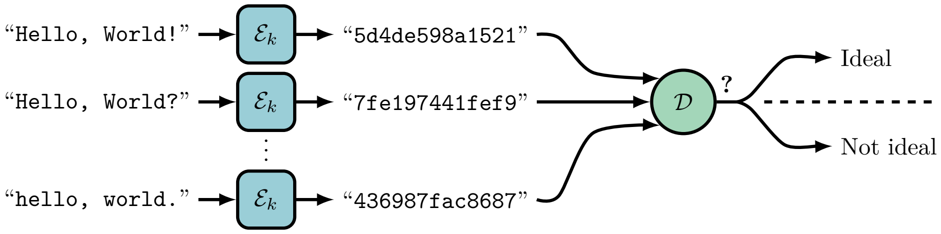


Figure 2.2: Illustration of a distinguishing attack. The distinguisher tries to determine whether $\mathcal{E}$ is an ideal cipher or not.

- **Key recovery**: The adversary tries to recover the encryption key k in time less than 2^κ.
- **Distinguishing**: Given black box access to either an ideal cipher or a concrete block cipher instance (and/or its inverse), the adversary tries to determine which of these two she is interacting with.

Since we assume that the only secret part of the block cipher is the key, key recovery implies deduction. The distinguishing attack is illustrated in Figure 2.2. While distinguishing might not immediately appear very useful in practice, for many practical cipher designs a good distinguisher often leads to a key recovery attack. Indeed, the works presented in Part II of this thesis are all concerned with finding such distinguishers.

Attack Settings Finally, it is natural to define different types of attack settings by what kind of information the adversary has available when performing the mentioned attacks, i.e. how she is allowed to interact with the black box. The typical types are as follows [69]:

- **Ciphertext only**: The adversary can query the black box for the encryption of randomly drawn, unknown, plaintexts.
- **Known plaintext**: The adversary can query the black box for randomly drawn plaintext-ciphertext pairs.
- **Chosen plaintext (ciphertext)**: The adversary can query the black box for the encryption (decryption) of plaintexts (ciphertexts) of her choosing.

Other attack settings, such as adaptively chosen text, related key [11], and weak key attacks [84], have also been discussed in the literature, but we will not consider these in the following.

It is not known how to construct a block cipher whose security can be reduced to known hard problems, in the way that popular asymmetric encryption algorithms can. For typical block cipher designs we therefore do not know how to prove resistance to all the general attacks mentioned here. Thus, a block cipher's security is commonly demonstrated by showing resistance to all known attacks. If no attack can be found, we have some confidence that the block cipher is secure, and under this assumption, other constructions that use the block cipher as a building block can be proven secure. We discuss a number of prominent attack techniques in Section 2.3, but we first mention some general ways in which we can construct block ciphers, as well as uses of these as part of other cryptographic primitives.

2.2 Block Cipher Constructions

In practice, we would like a block cipher to not only be secure but also efficient. Thus, it is common for block cipher designs to consist of relatively simple and fast components which are then applied repeatedly in order to increase complexity. This idea gives rise to the concept of an *iterative block cipher*.

Definition 2.3 (Iterative Block Cipher [47])**.** Consider a block cipher as in Definition 2.1. Let k_i, $i = 1, \dots, r$, be a set of *round keys* derived from the key k. Let $f_i^{k_i}$ be key-dependent permutations on $\mathbb{F}_2^n$. We call $f_i^{k_i}$ the i^{th} *round function*. If the block cipher can be written as

$$\mathcal{E}_k = f_r^{k_r} \circ \ldots \circ f_1^{k_1},$$

we call it an *iterative block cipher*.

One common type of iterative block cipher is the class of Feistel ciphers. For these ciphers, the i^{th} round function consists of splitting the input into two halves, say (x_L, x_R), and then outputting $(x_R, x_L \oplus g_i^{k_i}(x_R))$, where $g_i^{k_i}$ is a key-dependent function [69]. Another prominent type of iterative block cipher is the so-called *key-alternating block cipher* [47]. For this type of construction, the round function consists of applying a key-independent permutation to the input followed by a bitwise XOR with a round key. A common way to construct this key-independent permutation is according to the *substitution-permutation network* (SPN) approach.

Definition 2.4 (Substitution-Permutation Network [40, 41])**.** Consider an iterative block cipher $\mathcal{E}_k$ as in Definition 2.3. Let $s : \mathbb{F}_2^b \to \mathbb{F}_2^b$ be a permutation such that b divides n, and let $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ denote the parallel application of s to the $\frac{n}{b}$ b-bit chunks of the input. Let $P : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an affine function. If the i^{th} round function of $\mathcal{E}_k$ is given by

$$f_i^{k_i}(x) = P(S(x)) \oplus k_i,$$

and the first round function is preceded by the addition of a *whitening key* k_0, then $\mathcal{E}_k$ is called a substitution-permutation network.

The concept is illustrated in Figure 2.3. The core idea of this design strategy is that the b-bit s permutations, called S-boxes, provide strong local *confusion* which

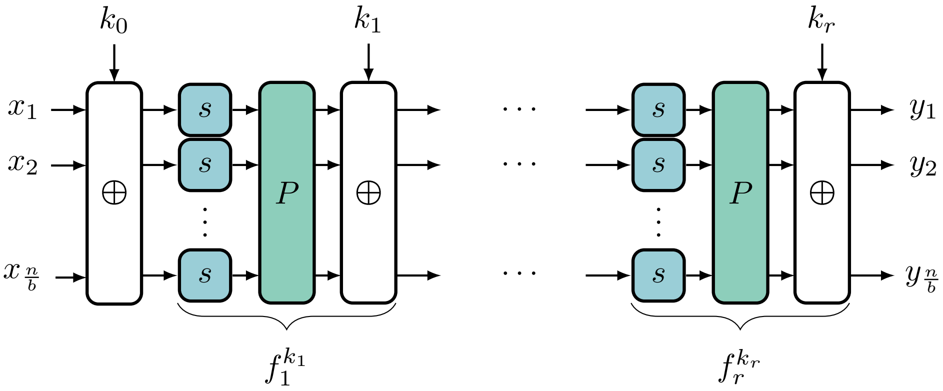


Figure 2.3: Illustration of a substitution-permutation network. The input to the block cipher is transformed by parallel S-boxes, mixed with an affine function P, and finally a round key is added. The processes is repeated r times.

is then *diffused* throughout the entire state by P [94]. Since the S-boxes can be expressed as small look-up tables, and the affine function P is often relatively simple, this approach can lead to very efficient block ciphers.

Block ciphers as building blocks From Definition 2.1 it is clear that the usefulness of a block cipher in isolation is limited: we can essentially only encrypt and decrypt a small amount of information. Thus, block ciphers are almost never used by themselves, but instead as components in other primitives. A basic example is using a block cipher in a *mode of operation* which allows for encryption/decryption of arbitrary length messages. Examples of basic modes of operation are the *cipher block chaining* (CBC) [52] and *counter* (CTR) [48] modes. Interestingly, the latter can be viewed as using a block cipher to construct a *secure pseudorandom number generator*, and then using this to construct a *stream cipher*.

In most cases, encryption alone is not enough, and we also require some form of *authentication*, which a block cipher alone does not supply. However, a *message authentication code* can be created from a block cipher using e.g. the CBC-MAC [1] or PMAC [19] constructions. If we want to combine encryption and authentication into one algorithm – a so-called *authenticated encryption* scheme – a block cipher can be used in e.g. *Galois/Counter mode* (GCM) [51], *Offset Codebook Mode* (OCB) [91], or *Counter with CBC-MAC mode* (CCM) [102]. Alternatively, several of the recent entries to the CAESAR competition for authenticated encryption schemes use a secure block cipher as their central component [2, 61, 83, 104].

Another important cryptographic primitive, namely a *cryptographic hash function*, can also be constructed from a block cipher. This can for example be done by first turning the block cipher into a one-way function using the Davies-Meyer construction [103], and then using this one-way function as the compression function of a Merkle-Damgård hash [82]. Cryptographic hash functions themselves appear in almost every aspect of cryptography.

Lastly, we note that an area closely related to that of block ciphers has garnered increased attention in recent years, namely that of *permutation based cryptography*. Instead of using a keyed block cipher to build other primitives, this area instead focuses on building primitives from a *fixed* cryptographic permutation. However, many of the security rationales used for block ciphers carry over to this setting. While several specific primitives based on permutations have been proposed, e.g. the hash function Grøstl [56] and the stream cipher Salsa [6], there has also been research into more general constructions. These are for example the *sponge* [10], *duplex* [9], and *Farfalle* [8] constructions, which can be viewed as different modes of operation for permutations. These modes are very versatile, and can be used to construct stream ciphers, hash functions, message authentication codes, authenticated encryption schemes, and more. In light of this, efficient cryptographic permutations have been designed, e.g. Gimli [7] and Xoodoo [43].

2.3 Attacks on Block Ciphers

The wide range of different use cases for block ciphers demonstrate immense their usefulness. More importantly, it emphasises the importance of having a high confidence in the security of any block cipher we may use as part of a bigger construction, in order to have any confidence in the security of said construction. It is little wonder then that the design and analysis of secure block ciphers is a highly active research area, and many different cryptanalysis techniques have been developed in order to attack a wide range of block cipher designs. In the following, we briefly describe some prominent attack techniques which can be used against block ciphers. This is by no means an exhaustive list of techniques, but any new block cipher design should at least demonstrate resistance to the following attacks.

1) Algebraic Attacks Any block cipher can be represented as a set of multivariate equations in the plaintext and key bits; by choosing some plaintexts, we obtain equations in the unknown key bits. Solving a general system of multivariate equations is hard, but if a cipher can be described by a simple equation system, solving this system immediately leads to a key recovery attack [95]. A sufficiently sparse set of quadratic equations can for example be solved with existing techniques such as linearisation or Gröbner basis methods [74]. Other more dedicated algorithms have also been proposed, such as the XL and XSL algorithms [38, 39].

A different way to exploit the algebraic structure of a block cipher is the *cube attack* [50]. In this attack, non-linear terms of the equations are eliminated by summing the equations for different values of the plaintext, resulting in linear equations in the key bits. If the number of plaintexts required is not too large, and we can generate enough such equations, the key can be recovered with simple Gaussian elimination. Alternatively, the cube attack can be viewed as a higher-order differential attack – these are described in more detail later in this section.

2) Meet-in-the-middle Attack This type of attack applies to block ciphers where the key can be split into two independent parts k_1 and k_2, such that the block cipher can be written as $\mathcal{E}(x, k_1 \| k_2) = b(f(x, k_1), k_2)$ [49]. In this case, a time-memory trade-off can be made in which a table containing $f(x, k_1)$ for a fixed x and all values of k_1 is stored. Then, during the attack, $y = \mathcal{E}_k(x)$ is obtained, k_2 is guessed, and k_1 is found by computing $b^{-1}(y, k_2)$ and finding this value in the table. Many extensions to this basic technique have been explored in the literature, such as MITM with partial matching [32], 3-subset MITM [27], MITM with splice and cut techniques [3, 101], and the biclique attack [24].

3) Integral Cryptanalysis This type of attack (also known as the square, saturation, or multiset attack) [18, 44, 71, 77] utilises sets of plaintexts where e.g. one byte varies over all possible values, while the rest of the plaintext is fixed. Clearly, the sum of the texts in such a set is zero, also called balanced. An integral attack can be mounted if it is possible to predict that (part of) the corresponding encrypted set will also be balanced.

An interesting generalisation of the integral attack is the *division property* attack [98]. In this attack, more general conditions than balancedness are used as a distinguisher, e.g. whether the sum of any polynomial expression of the plaintexts of at most degree k is even. This generalisation yields better results for some block ciphers where the basic integral attack is not very effective.

4) Invariant Subspace Attack As the name suggests, this attack tries to find a subspace of $\mathbb{F}_2^n$ such that for some keys the ciphertext corresponding to a plaintext in this subspace is also part of the subspace [76]. Clearly, for such a key, the cipher can immediately be distinguished from an ideal cipher. For some ciphers, this property has been shown to also facilitate efficient key recovery. While the original version of the attack uses affine subspaces, a version using non-linear invariants has also been proposed [99]. A related idea can also be found in the yoyo attack [92] in which texts are chosen in such a way that specific differences between them are independent of the key, providing a distinguisher.

5) Differential Cryptanalysis While the attacks mentioned so far are essentially deterministic, differential cryptanalysis exploits probabilistic behaviour of the block cipher. This technique uses pairs of plaintexts which have a specific difference, and then considers the probability that the corresponding pair of ciphertexts also has some given difference. If we can find such a *differential* which occurs with sufficiently high probability, this can be used as a distinguisher. Such a distinguisher can usually be used as part of a key recovery attack [16].

A multitude of extensions have been proposed to the simple differential attack. *Truncated differentials* [68] relax the requirement of specific input and output differences, and instead only partially define these, e.g. we allow any type of difference in the first two bytes of the input and output, but all other bytes must have no difference. In this way, we essentially consider many differentials simultaneously, hopefully increasing the total probability. Another generalisation is *higher order differential cryptanalysis* [68, 72]; the normal differentials can be viewed as first order discrete derivatives, and so it is natural to take higher order derivatives. Taking the d^{th} order derivative reduces the degree of the function by at least d, and so this might facilitate easier cryptanalysis. Examples of second order differential attacks are the *boomerang* attack [100] and its extension the *rectangle* attack [14]. Finally, we note that differentials that have exactly zero probability of occurring can also be used in a so-called *impossible differential attack* [12, 67].

6) Linear Cryptanalysis Another prominent type of probabilistic attack is *linear cryptanalysis* [78, 80]. In this attack, we try to find a linear expression in the bits of the plaintext and a (potentially different) linear expression in the bits of the ciphertext which correlate strongly with each other. As for differential cryptanalysis, such a connection between plaintext and ciphertext can be used as a distinguisher, and ultimately as part of a key recovery attack.

The rest of this thesis will be concerned with linear cryptanalysis. Chapter 3 will cover the basics of the topic and discuss various extensions to the simple attack. Part II contains several publications that advance the field in various ways. The contributions of these works are summarised in Chapter 4.

3 Linear Cryptanalysis

This chapter will give an introduction to linear cryptanalysis. Section 3.1 introduces the basic ideas and notation. Section 3.2 presents useful tools for analysing linear approximations of practical block ciphers, while Section 3.3 explains how linear distinguishing and key recovery attacks work. Section 3.4 explains various ways to use more than one approximation for an attack, and finally some other extensions are discussed in Section 3.5.

3.1 Fundamentals of Linear Cryptanalysis

Consider a block cipher as defined in Definition 2.1, and recall the distinguishing attack described in Section 2.1. The motivation behind this attack is that if we can tell a given block cipher apart from a completely randomly drawn permutation, the cipher must exhibit some non-random behaviour which indicates a flaw in the design – something which can often be used in e.g. a key recovery attack. For linear cryptanalysis [78, 80], this non-randomness is indicated by linear expressions in the plaintext and ciphertext bits that are biased towards 0 or 1.

In order to formalise the above idea, we first introduce the concept of a *linear approximation*. The approximation essentially defines the linear expression of bits we will use in the attack.

Definition 3.1 (Linear approximation [78])**.** *For a block cipher as given in Definition 2.1, a linear approximation is a tuple $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. We call α the input mask and β the output mask.*

To go from the two masks to a linear function, we use an inner product. Let α be as in Definition 3.1 and x be an element of $\mathbb{F}_2^n$. Let $x[i]$ be the i^{th} component (bit) of x. We define the canonical inner product on $\mathbb{F}_2^n$ as

$$\langle \alpha, x \rangle = \sum_{i=1}^{n} \alpha[i] \cdot x[i], \quad \text{for } \alpha, x \in \mathbb{F}_2^n,$$

i.e. the sum of the bitwise products of α and x. For a fixed x, $\langle \alpha, x \rangle$ is a linear boolean function of x which simply expresses the sum of the bits of x as indicated by α. If x is drawn randomly, the probability that $\langle \alpha, x \rangle = 0$ is $\frac{1}{2}$. But if x is not completely random, this probability can be smaller or larger than $\frac{1}{2}$. For a block cipher, we therefore associate a *linear correlation* to an approximation as follows.

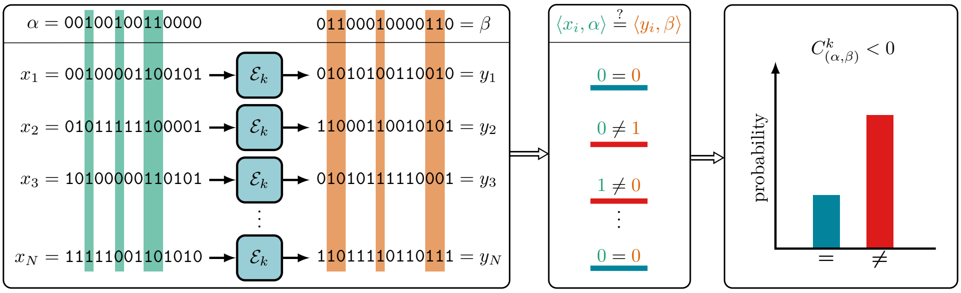


Figure 3.1: Illustration of a linear approximation (α, β) and its linear correlation. In the left box, it is shown how α and β indicate bits of the plaintexts and ciphertexts, respectively. In the middle box, the parity of the indicated plaintext bits are compared to the parity of the indicated ciphertext bits. In the right box, the histogram shows that the plaintext parity is more likely to be *unequal* to the ciphertext parity, resulting in a negative correlation.

Definition 3.2 (Linear Correlation [42])**.** The *linear correlation* of an approximation (α, β) of a block cipher $\mathcal{E}$ is given by

$$C^k_{(\alpha,\beta)} = 2 \cdot \Pr_{x \in \mathbb{F}_2^n} \left(\langle x, \alpha \rangle \oplus \langle \mathcal{E}_k(x), \beta \rangle = 0 \right) - 1,$$

for a fixed key $k \in \mathbb{F}_2^\kappa$.

The concept is illustrated in Figure 3.1. We say that an approximation is *non-trivial* if both α and β are non-zero. For any non-trivial approximation, since each key k is likely to correspond to a different permutation on $\mathbb{F}_2^n$, the linear correlation will in general take on different values over the key space. Indeed, we are often not interested in $C^k_{(\alpha,\beta)}$ for any fixed k, but the distribution of $C^k_{(\alpha,\beta)}$ over the space of all keys. This distribution highly depends on the block cipher. However, the following result can be shown for the ideal cipher.

Theorem 3.3 ([46, 88])**.** *For any non-trivial approximation (α, β) of an ideal block cipher, the discrete probability distribution of the linear correlation over keys is given by*

$$\Pr\left(C^k_{(\alpha,\beta)} = 2^{2-n} x \right) = \frac{\binom{2^{n-1}}{2^{n-2}+x}^2}{\binom{2^n}{2^{n-1}}},$$

which can be approximated by a normal distribution with zero mean and variance 2^{-n}, i.e. $C^k_{(\alpha,\beta)} \sim \mathcal{N}(0, 2^{-n})$.

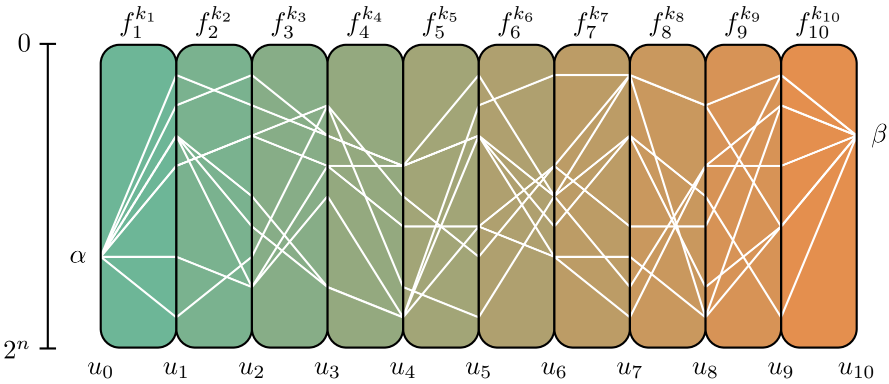


Figure 3.2: An illustration of the linear hull of a linear approximation (α, β). Many different linear trails connect α and β through the round functions $f_i^{k_i}$. Each intermediate linear mask u_i is a value in $\mathbb{F}_2^n$.

Thus, if we want to build a distinguisher from a linear approximation, we need to find an approximation whose correlation distribution over keys is sufficiently different from $\mathcal{N}(0, 2^{-n})$. Alas, for a concrete cipher design, it is less clear how one determines the distribution of $C_{(\alpha,\beta)}^k$, or even the linear correlation for any fixed key. We explore this topic next.

3.2 Linear Trails and the Linear Hull

In general, determining the linear correlation of a block cipher is a hard problem. If we wanted to measure the correlation directly, even for a single key, we would have to query the cipher about 2^n times. However, for the specific block cipher constructions presented in Section 2.2, there exist results that allow us to at least approximate the linear correlation.

For an iterative block cipher, the functions $f_i^{k_i}$ are usually relatively simple, allowing us to directly calculate the correlation of any approximation of one of these functions. Now, in order to calculate the correlation over the whole function $\mathcal{E}_k$, we first introduce to notion of a linear trail.

Definition 3.4 (Linear Trail [42])**.** Given a linear approximation (α, β) of an iterative block cipher, a *linear trail* is an $r + 1$ tuple

$$U = (u_0, \ldots, u_r) \quad \text{with } u_0 = \alpha,\, u_r = \beta.$$

Let C_i^k be the correlation of the approximation (u_{i-1}, u_i) of the i^{th} round function

$f_i^{k_i}$. We define the *correlation contribution* of the trail U as

$$C_U^k = \prod_{i=1}^{r} C_i^k.$$

We call the collection of all trails of an approximation the *linear hull* [86], and the concept is illustrated in Figure 3.2. The definition of a linear trail is primarily useful due to the following result, which states that the linear hull fully determines the correlation of an approximation.

Theorem 3.5 ([40, 42])**.** *Given a linear approximation (α, β) of an iterative block cipher, the linear correlation is equal to the sum of correlation contributions for all linear trails of (α, β):*

$$C_{(\alpha,\beta)}^k = \sum_{\substack{u_0=\alpha \\ u_r=\beta}} C_U^k. \tag{3.1}$$

While we could use Theorem 3.5 to calculate the linear correlation, applying it in practice is quite challenging, as the number of trails in the linear hull is extremely large even for moderate values of n and r. Additionally, the terms of the sum in Equation 3.1 are not necessarily independent, complicating any analysis of the distribution of $C_{(\alpha,\beta)}^k$ over the key space. The study of techniques for approximating Equation 3.1 is essential to linear cryptanalysis, and a large part of research on this topic is dedicated to this problem. Indeed, it is central to all the works presented in Part II of this thesis.

3.3 Linear Distinguishers and Key Recovery

Let us for a moment assume that we know the distribution of $C_{(\alpha,\beta)}^k$ for some non-trivial approximation (α, β) and some block cipher $\mathcal{E}$. In the following, we describe how to build a known plaintext distinguisher from this knowledge, and how to use such a distinguisher as part of a key recovery attack.

We first introduce the notion of the *undersampled* correlation: an adversary is often not interested in obtaining the full codebook in order to measure $C_{(\alpha,\beta)}^k$ exactly, as this naturally increases the computational complexity of the attack. Thus, she will obtain from $\mathcal{E}_k$ some set $\mathcal{T}$ of $N < 2^n$ randomly drawn plaintext-ciphertext pairs and calculate an undersampled correlation value:

$$C_{(\alpha,\beta)}^{k,\mathcal{T}} = 2 \cdot \Pr_{(x,y)\in\mathcal{T}} (\langle x, \alpha \rangle \oplus \langle y, \beta \rangle = 0) - 1, \tag{3.2}$$

Note that $C_{(\alpha,\beta)}^{k,\mathcal{T}}$ is a random variable over both the key space and the text space. The distribution of $C_{(\alpha,\beta)}^{k,\mathcal{T}}$ has been studied extensively, and in most cases it is possible to derive this distribution from the distribution of $C_{(\alpha,\beta)}^k$ over keys [21, 85]. Therefore, we assume this to be known to the adversary. As an example, the undersampled distribution of an ideal cipher is $\mathcal{N}(0, 2^{-n} + \frac{1}{N})$.

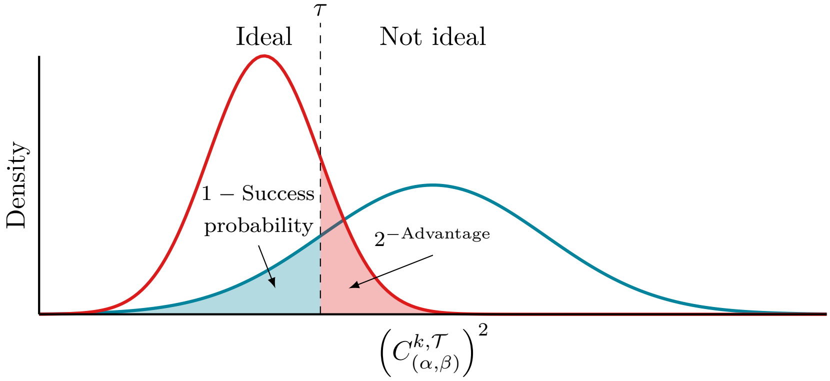


Figure 3.3: An illustration of a statistical distinguisher based on the linear correlation of a linear approximation. A value $(C^{k,\mathcal{T}}_{(\alpha,\beta)})^2$ drawn from the undersampled squared correlation distribution over keys and texts is compared to the threshold τ. Based on this comparison, it is decided whether $(C^{k,\mathcal{T}}_{(\alpha,\beta)})^2$ was drawn from the ideal distribution or not.

Linear Distinguishing Recall that the goal of a distinguishing attack is for the adversary to determine whether she is interacting with an ideal or a non-ideal block cipher. For linear cryptanalysis, this boils down to determining whether a correlation value was drawn from the distribution $\mathcal{N}(0, 2^{-n})$ or the distribution of $C^k_{(\alpha,\beta)}$ (or their undersampled equivalents). To simplify the exposition, we will instead use the squared correlation, in which case the ideal distribution is $2^{-n}\chi^2$, and assume that $\mathrm{E}(2^{-n}\chi^2) \leq \mathrm{E}((C^k_{(\alpha,\beta)})^2)$, where E denotes the mean of the distributions. The following is a simple way to perform a linear distinguishing attack [64].

- Fix a *threshold value* τ.
- Obtain a set $\mathcal{T}$ of N random plaintext-ciphertext pairs from the block cipher, and calculate the undersampled linear correlation $C^{k,\mathcal{T}}_{(\alpha,\beta)}$ as in Equation 3.2.
- If $(C^{k,\mathcal{T}}_{(\alpha,\beta)})^2 < \tau$, assume that the block cipher is ideal. If $(C^{k,\mathcal{T}}_{(\alpha,\beta)})^2 \geq \tau$ assume otherwise.

For a statistical distinguishing attack like the above, we are primarily interested in two things: how often we correctly classify a non-ideal cipher as such (true positive), and how often we erroneously classify an ideal cipher as non-ideal (false positive). Clearly, this depends on τ and the distribution of $C^{k,\mathcal{T}}_{(\alpha,\beta)}$. We express these rates, and thus the effectiveness of the distinguisher, using the notions of success probability and advantage.

Definition 3.6 (Success Probability and Advantage [93])**.** For a linear distinguisher as described above, we define the *success probability* as

$$p_S = \Pr\left((C_{(\alpha,\beta)}^{k,\mathcal{T}})^2 \geq \tau \mid \mathcal{E} \text{ is not ideal}\right),$$

and the *advantage* as

$$a = -\log_2\left(\Pr\left((C_{(\alpha,\beta)}^{k,\mathcal{T}})^2 \geq \tau \mid \mathcal{E} \text{ is ideal}\right)\right).$$

These concepts are illustrated in Figure 3.3. Typically, an attacker will fix the success probability and calculate the corresponding threshold value and advantage. While the motivation for the definition of the success probability is clear in the distinguishing setting, the advantage primarily plays a role when we want to use the distinguisher as part of a key recovery attack.

Linear Key Recovery Consider an iterative block cipher with r rounds as given in Definition 2.3. We now define a reduced version of the cipher with $r-1$ rounds, i.e.

$$\mathcal{E}'_k = f_{r-1}^{k_{r-1}} \circ \ldots \circ f_1^{k_1}.$$

Let $g_r^{k_r}$ denote the inverse of $f_r^{k_r}$ and let (α, β) be a linear approximation of $\mathcal{E}'_k$. By guessing the last round key and applying the distinguishing attack to $\mathcal{E}'_k$, we obtain a key recovery attack [78]. In more detail, the attack works by using the distinguisher to filter out bad key guesses as follows.

- Fix a *threshold value* τ.
- Obtain a set $\mathcal{T}$ of N random plaintext-ciphertext pairs from the block cipher $\mathcal{E}_k$.
- For each guess of k_r, apply $g_r^{k_r}$ to each ciphertext in order to obtain sets $\mathcal{T}_{k_r}$ of potential plaintext-ciphertext pairs of the cipher $\mathcal{E}'_k$.
- For each set of plaintext-ciphertext pairs, calculate $C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}}$ as in Equation 3.2.
- If $(C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}})^2 < \tau$, discard the corresponding guess of k_r. If $(C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}})^2 \geq \tau$ save the key guess.

In practice, it is usually sufficient to only partially guess k_r in order to calculate $C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}}$. Moreover, it is often possible to calculate the encryption key k if one or more of the round keys are known. Once we have a number of candidates for k, the correct key can be identified e.g. through trial encryption.

This type of key recovery attack relies on the following hypothesis: if we make a wrong guess of the last round key, the resulting ciphertext will look random. This hypothesis formalises the idea that if we decrypt the last round with a wrong key,

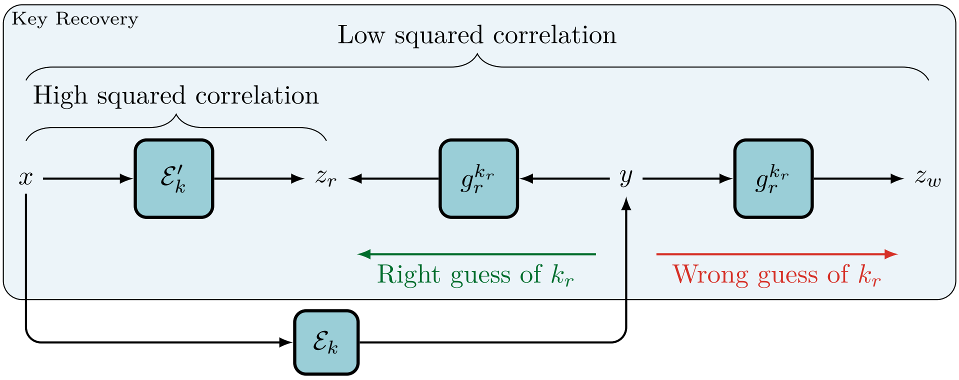


Figure 3.4: Illustration of a key recovery attack against an iterative block cipher $\mathcal{E}_k$ using linear cryptanalysis. Plaintext-ciphertext pairs (x, y) are obtained and the last round key k_r is guessed. For a correct guess, we obtain a set of pairs (x, z_r), whereas for a wrong guess we obtain a set of pairs (x, z_w). A high resulting squared linear correlation indicates a correct guess.

we are essentially adding one round to the cipher. In this case, we are considering an approximation over $r + 1$ rounds, instead of $r - 1$ rounds, which should have a much weaker correlation. The idea is illustrated in Figure 3.4.

Hypothesis 1 (Wrong-Key Randomisation [29, 57, 63])**.** *Consider a key recovery attack as described above. If the last round key k_r is incorrectly guessed, then $C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}}$ will be distributed as for an ideal cipher, namely $\mathcal{N}(0, 2^{-n} + \frac{1}{N})$.*

Under this hypothesis, the advantage relates to the number of candidates we get for the encryption key k. Assume that we guess $\bar{\kappa}$ bits during the key recovery attack. By definition of the advantage we expect $2^{\bar{\kappa}} \cdot 2^{-a}$ key guesses to survive the filtering described in the attack above. If we assume that guessing the remaining $\kappa - \bar{\kappa}$ bits allows us to determine a candidate for k, then the number of candidates we get is

$$2^{\bar{\kappa}} \cdot 2^{-a} \cdot 2^{\kappa - \bar{\kappa}} = 2^{\kappa - a}.$$

Thus, the attack effectively reduces the size of the key space by a bits [93]. Since, for a given threshold value, the advantage is entirely determined by the distribution of $C_{(\alpha,\beta)}^{k,\mathcal{T}}$, determining this distribution, or at least obtaining a good estimate of it, is essential to linear cryptanalysis.

Note that the above exposition is only one way of performing and analysing a linear key recovery attack. One alternative way of performing the attack is to rank each key candidate by the magnitude of $(C_{(\alpha,\beta)}^{k,\mathcal{T}_{k_r}})^2$, and then search the list of candidates from highest to lowest correlation. This approach was originally taken in [79] and analysed in [64, 93], amongst others. Alternatively, if the linear approximation has a single trail whose correlation contribution is much larger than that of any other trail, the sign of $C^{k,\mathcal{T}_{k_r}}_{(\alpha,\beta)}$ can also be used to deduce some bits of the key [78].

3.4 Using Multiple Approximations

Many extensions to the basic linear cryptanalysis described above have been proposed, but we will focus on those using several approximations simultaneously. Attacks that exploit several approximations at the same time are usually split into two categories: *multiple linear attacks* and *multidimensional linear attacks*. In short, the difference between these two types lies mainly in the assumptions made about the statistical behaviour of the linear correlations, and the type of sets of approximations they use.

3.4.1 Multiple Linear Cryptanalysis

The idea of using multiple linear approximations simultaneously to improve linear attacks was first proposed in [62] and was later extended in [17]. These works propose using a set of linear approximations

$$\mathcal{A} = \{(\alpha_1, \beta_1), \ldots, (\alpha_M, \beta_M)\},$$

and its corresponding vector of linear correlations

$$C^k_{\mathcal{A}} = (C^k_{(\alpha_1,\beta_1)}, \ldots, C^k_{(\alpha_M,\beta_M)}).$$

The goal is then essentially to distinguish the M-variate distribution of $C^k_{\mathcal{A}}$ over keys from the corresponding M-variate distribution for an ideal block cipher. While [17] did describe how the location of a correlation measurement $(C^{k,\mathcal{T}}_{(\alpha_1,\beta_1)}, \ldots, C^{k,\mathcal{T}}_{(\alpha_M,\beta_M)})$ in an M-dimensional space can be used to reveal some information about the key, it is more common to calculate some univariate distribution from the distribution of $C^k_{\mathcal{A}}$. Indeed, [17] introduced the notion of *capacity* as a measure of the "combined correlation" of the M approximations, defined as the sum of squared correlations:

$$\mathcal{C}^k = \sum_{i=1}^{M} (C^k_{(\alpha_i,\beta_i)})^2.$$

Note that it is straightforward to generalise the attack description of Section 3.3 to the case of multiple approximations by simply replacing $(C^{k,\mathcal{T}}_{(\alpha,\beta)})^2$ with a measurement of the capacity. However, in general it is highly non-trivial to determine the distribution of $C^k_{\mathcal{A}}$ over keys, and thus also the distribution of $\mathcal{C}^k$, making it difficult to evaluate the effectiveness of such an attack. Therefore, both [62] and [17] assume that the approximations in $\mathcal{A}$ are statistically independent, facilitating their analysis. For this reason, the term *multiple linear cryptanalysis* is usually associated with this independence assumption.

3.4.2 Multidimensional Linear Cryptanalysis

In order to eliminate the assumption of statistical independence made for multiple linear cryptanalysis, an alternative approach was proposed in [35, 59]. This approach, called *multidimensional linear cryptanalysis*, builds on the earlier work [4]. Instead of considering linear approximations directly, multidimensional linear cryptanalysis considers the value of $x \| \mathcal{E}_k(x) \in \mathbb{F}_2^{2n}$, where $\|$ denotes concatenation, restricted to some subspace of $\mathbb{F}_2^{2n}$. Specifically, let A be a $d_A \times n$ matrix whose rows are linearly independent and have Hamming weight 1, and B a $d_B \times n$ matrix with identical constraints. Then the function

$$h_{(A,B)}(x \| \mathcal{E}_k(x)) = A \cdot x \| B \cdot \mathcal{E}_k(x),$$

maps $x \| \mathcal{E}_k(x)$ to a $d = d_A + d_B$ dimensional subspace of $\mathbb{F}_2^{2n}$ by selecting d_A components of x and d_B components of $\mathcal{E}_k(x)$. We now consider the probability that $h_{(A,B)}(x \| \mathcal{E}_k(x))$ takes on a specific value in $\mathbb{F}_2^d$, that is, we define a d-variate probability distribution $\boldsymbol{\eta}^k = (\eta_0^k, \ldots, \eta_{2^d-1}^k)$ by

$$\eta_i^k = \Pr_{x \in \mathbb{F}_2^n} \left(h_{(A,B)}(x \| \mathcal{E}_k(x)) = i \right) \text{ for } i \in \mathbb{F}_2^d.$$

We say that (A, B) is a d-dimensional linear approximation, and the rows of A and B are called basis approximations. It can be shown that this multidimensional approximation is equivalent to the set of $2^d - 1$ non-zero, one-dimensional linear approximations spanned by the basis approximations [59]. In particular, the capacity of these $2^d - 1$ approximations can be calculated as

$$\mathcal{C}^k = 2^d \sum_{i=1}^{2^d-1} (\eta_i^k - 2^{-d})^2.$$

Since the η_i^k are independent, with the restriction that they sum to 1 for any k, this potentially allows us to determine the distribution of $\mathcal{C}^k$ over keys without assuming independence of the involved approximations. Indeed, since its introduction, multidimensional linear cryptanalysis has given rise to a number of attacks on block ciphers [34, 36].

3.5 Other Extensions

Some other flavours of linear cryptanalysis have been proposed. As an analogue to impossible differentials, *zero-correlation linear cryptanalysis* [28] uses linear approximations that have correlation exactly zero for all keys. While the basic variant requires a high data complexity in order to measure the correlation of such an approximation exactly, a variant that uses multiple zero-correlation approximations is able to decrease the amount of plaintext-ciphertext pairs needed [30].

A related-key variant of linear cryptanalysis, the *key difference invariant bias attack* [23], has also been developed. This variant uses approximations which have the same correlation value between two keys that exhibit a specific difference.

While most work on linear cryptanalysis assumes that plaintexts are drawn randomly with replacement from $\mathbb{F}_2^n$, some works have considered settings where the plaintexts a drawn without replacement [21, 26]. Additionally, a recent publication suggests filtering the plaintexts in order to achieve a higher correlation [15].

A number of different generalisations of linear cryptanalysis have been considered. In particular, several ways of replacing the linear expressions in the input and output bits with non-linear expressions in these bits have been proposed [37, 57, 58, 70, 96]. In a similar vein, the idea of using expressions over groups or fields other than $\mathbb{F}_2$ has also been explored [5, 45, 65, 90].

Combining linear approximations with differentials in the so-called differential-linear attack has also proven useful in some cases [13, 73]. Finally, several connections to other attacks have been made, such as differential cryptanalysis [22, 31], integral cryptanalysis [97], and statistical saturation attacks [75].

4 Contributions of Publications

Part II of this thesis presents four papers that further the field of linear cryptanalysis. In the following, we give an overview of the contributions of these works. As explained in Chapter 3, determining the distribution of the linear correlation of an approximation is essential in order to estimate the effectiveness of a linear attack. In principle, Theorem 3.5 solves this problem, however applying it is infeasible in practically every interesting scenario. Thus, various simplifying assumptions have been made throughout the history of linear cryptanalysis.

As a starting point, it was often assumed that a linear approximation had a *dominating trail*, meaning a trail with a much larger correlation contribution than all other trails. Thus, Equation 3.1 could be estimated just from this trail. While this assumption simplifies analysis, it is not strictly true in practice, and it has been shown that the effect of multiple trails can be very strong for some ciphers [75, 89]. Other common assumptions in early linear cryptanalysis were that the linear correlation was virtually the same for all values of the key, and that the round keys were independent. These assumptions greatly simplify analysis, especially in the case of multiple/multidimensional linear cryptanalysis. In recent years, much work has been done on removing these types of simplifying assumptions [20, 21, 29, 60, 87]. The papers presented in Part II are part of this effort.

Building Tools The starting point of our work is essentially the *signal/noise decomposition* proposed in [29] and used in e.g. [20]. In this model, we assume that we know a set $\mathcal{S}$ (the *signal*) of linear trails of the approximation that have a large correlation contribution. We then define the *signal correlation* as

$$C_{\mathcal{S}}^k = \sum_{U \in \mathcal{S}} C_U^k.$$

The distribution of $C_{(\alpha,\beta)}^k$ is then approximated by assuming that the remaining trails behave like *noise*. That is, we make the approximation

$$C_{(\alpha,\beta)}^k \approx C_{\mathcal{S}}^k + \mathcal{N}(0, 2^{-n}).$$

Ideally, we want to know as large a set of signal trails as possible. This motivates the work done in the paper *Generating Graphs Packed with Paths* (Publication 3), where we present a new algorithm for linear trail search. While many other algorithms with this purpose have been presented in the literature, their complexity is almost always linear in the number of trails, which quickly becomes a problem if a good signal set is very large. In contrast, our algorithm is specifically designed to avoid this problem, which we demonstrate by finding as much a 2^{112} trails for the block cipher PUFFIN [33]. In addition, we also present an efficient algorithm for sampling from the signal distribution, directly facilitating the use of the signal/noise decomposition model. This algorithm also allows for sampling of correlations for multiple approximations simultaneously, without any assumptions on the round keys.

Refining Models The aforementioned algorithms make it easier to take a more computational approach to linear cryptanalysis. We can now draw observations from the distribution of $C^k_{(\alpha,\beta)}$ or from the distribution of its multivariate equivalent

$$\boldsymbol{C}^k_{\mathcal{A}} = (C^k_{(\alpha_1,\beta_1)}, \dots, C^k_{(\alpha_M,\beta_M)}),$$

without making any assumptions about statistical independence of approximations, trails, or round keys. In *Multivariate Profiling of Hulls for Linear Cryptanalysis* (Publication 2) we build a model for multiple linear cryptanalysis in this framework, called the *multivariate profiling model*. The big advantage of this model is that it makes no a priori assumptions about the shape or dependence structure of the signal distribution. In principle, it is therefore able to express any distribution $\boldsymbol{C}^k_{\mathcal{A}}$ might have, with the single limitation that the approximations are linearly independent. We analyse the block cipher PRESENT [25] in this new model, and demonstrate that the key-schedule of the cipher does have an effect on the shape of the multivariate distribution of $\boldsymbol{C}^k_{\mathcal{A}}$. Then, we present a new attack on 27 out of 31 rounds of the cipher.

Exploring Correlations While the linear approximations of PRESENT are somewhat well behaved, meaning that they are approximately jointly normally distributed, we revisit linear cryptanalysis of DES [55] in *Linear Cryptanalysis of DES with Asymmetries* (Publication 1) and find that the situation here is more complicated. Indeed, we show that the signal distribution can be expressed as a multivariate normal mixture, leading to a special case of the multivariate profiling model. More surprisingly, we find sets of approximations for which the correlation distributions are not symmetric around zero, as one would expect if assuming statistical independence of the approximations. We propose using a likelihood-ratio approach in order to fully exploit these asymmetries during an attack, and as a result we present a multiple linear attack on full DES which improves both time and data complexity of previous attacks.

Inspired by the above observations, the paper *On Linear Correlation Distributions: More Instructive Examples* (Publication 4) takes a closer look at how the shape of correlation distributions impacts our ability to attack a cipher. We compare the advantage obtained using multiple approximations under various standard independence assumptions against the advantage obtained using the multivariate profiling model. Interestingly, we find that for ciphers that fit in the normal mixture model, the exact configuration of the mixture components has a significant impact on the advantage. Specifically, we find both cases where the advantage in the profiling model is significantly higher than when using independence assumptions, and cases where it is significantly lower. We also find one case, the block cipher RECTANGLE [105], for which the correlation distribution is highly non-normal, severely decreasing the advantage.

In conclusion, the papers presented in the following provide new tools and models for accurately assessing the effectiveness of linear cryptanalysis using multiple approximations. They furthermore demonstrate that the cryptanalyst should take care when creating a new attack, as the behaviour of linear correlations seems to be highly dependent on the cipher. Indeed, there is still much work to be done before we fully understand these attacks.

Future Challenges While the publications presented in the following are a good step in the direction of a deeper understanding of linear cryptanalysis, there are also many open questions left to answer. Chief amongst these is perhaps the problem of an appropriate wrong-key randomisation hypothesis in the multiple/multidimensional case. While it is clear that the marginal correlation distributions of $C_{\mathcal{A}}^{K}$ over keys are normal, following Theorem 3.3, it is an open problem exactly how the joint distribution looks. Indeed, it is currently unknown whether two linearly independent approximations of an ideal cipher are also statistically independent. The question of what the dependence structure of two linearly dependent approximations looks like seems even harder to tackle. Solving this problem would effectively bridge the gap between multiple and multidimensional linear cryptanalysis, allowing for the use of completely arbitrary sets of approximations.

In a similar vein, it is unclear what exactly causes the sometimes large deviation from normality of the multivariate correlation distributions observed for some ciphers, as demonstrated in e.g. Publication 4. In particular, it would be interesting to explore how different design decisions influence these distributions. A first target for this type of research could be to examine the effect of the of key schedule on multiple linear cryptanalysis more closely. A more complex task would be to examine exactly how the choice of linear layer in an SPN cipher affects the clustering of linear trails and ultimately the shape of the joint correlation distribution. Related to such work, new attack techniques that more directly exploit the shape of the correlation distributions could be investigated.

Lastly, seeing that Theorem 3.5 in large part enables the work we have done on correlation distributions for SPN ciphers, it would be interesting to develop a similar result for differential cryptanalysis. For that type of attack, it is largely unknown exactly how the key influences the statistical behaviour of differentials. Thus, it would be interesting to attempt to reduce the number of assumptions in this setting as well, and then reevaluate old attacks.

Bibliography

[1] ISO/IEC 9797-1:2011. “Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher”. In: (1999).

[2] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda. “COLM v1”. In: *Submission to the CAESAR competition* (2016).

[3] Kazumaro Aoki and Yu Sasaki. “Preimage Attacks on One-Block MD4, 63-Step MD5 and More”. In: *Selected Areas in Cryptography, SAC 2008*. 2008, pp. 103–119.

[4] Thomas Baignères, Pascal Junod, and Serge Vaudenay. “How Far Can We Go Beyond Linear Cryptanalysis?” In: *Advances in Cryptology - ASIACRYPT 2004*. 2004, pp. 432–450.

[5] Thomas Baignères, Jacques Stern, and Serge Vaudenay. “Linear Cryptanalysis of Non Binary Ciphers”. In: *Selected Areas in Cryptography, SAC 2007*. 2007, pp. 184–211.

[6] Daniel J. Bernstein. “The Salsa20 Family of Stream Ciphers”. In: *New Stream Cipher Designs - The eSTREAM Finalists*. 2008, pp. 84–97.

[7] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. “Gimli : A Cross-Platform Permutation”. In: *Cryptographic Hardware and Embedded Systems - CHES 2017*. 2017, pp. 299–320.

[8] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. “Farfalle: Parallel Permutation-Based Cryptography”. In: *IACR Trans. Symmetric Cryptol.* 2017.4 (2017), pp. 1–38.

[9] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. “Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications”. In: *Selected Areas in Cryptography, SAC 2011*. 2011, pp. 320–337.

[10] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. “Sponge Functions”. In: *ECRYPT Hash Workshop*. Vol. 2007. 9. 2007.

[11] Eli Biham. “New Types of Cryptanalytic Attacks Using Related Keys”. In: *Journal of Cryptology* 7.4 (1994), pp. 229–246.

[12] Eli Biham, Alex Biryukov, and Adi Shamir. "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials". In: *Advances in Cryptology - EUROCRYPT '99*. 1999, pp. 12–23.

[13] Eli Biham, Orr Dunkelman, and Nathan Keller. "Enhancing Differential-Linear Cryptanalysis". In: *Advances in Cryptology - ASIACRYPT 2002*. 2002, pp. 254–266.

[14] Eli Biham, Orr Dunkelman, and Nathan Keller. "The Rectangle Attack - Rectangling the Serpent". In: *Advances in Cryptology - EUROCRYPT 2001*. 2001, pp. 340–357.

[15] Eli Biham and Stav Perle. "Conditional Linear Cryptanalysis – Cryptanalysis of DES with Less Than 2^{42} Complexity". In: *IACR Transactions on Symmetric Cryptology* 2018.3 (2018), pp. 215–264.

[16] Eli Biham and Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In: *Advances in Cryptology - CRYPTO '90*. 1990, pp. 2–21.

[17] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. "On Multiple Linear Approximations". In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 1–22.

[18] Alex Biryukov and Adi Shamir. "Structural Cryptanalysis of SASAS". In: *Advances in Cryptology - EUROCRYPT 2001*. 2001, pp. 394–405.

[19] John Black and Phillip Rogaway. "A Block-Cipher Mode of Operation for Parallelizable Message Authentication". In: *Advances in Cryptology - EUROCRYPT 2002*. 2002, pp. 384–397.

[20] Céline Blondeau and Kaisa Nyberg. "Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis". In: *IACR Transactions on Symmetric Cryptology* 2016.2 (2016), pp. 162–191.

[21] Céline Blondeau and Kaisa Nyberg. "Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and its Impact to Data Complexity". In: *Designs, Codes and Cryptography* 82.1-2 (2017), pp. 319–349.

[22] Céline Blondeau and Kaisa Nyberg. "New Links between Differential and Linear Cryptanalysis". In: *Advances in Cryptology - EUROCRYPT 2013*. 2013, pp. 388–404.

[23] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao. "Key Difference Invariant Bias in Block Ciphers". In: *Advances in Cryptology - ASIACRYPT 2013*. 2013, pp. 357–376.

[24] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. "Biclique Cryptanalysis of the Full AES". In: *Advances in Cryptology - ASIACRYPT 2011*. 2011, pp. 344–371.

[25] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. “PRESENT: An Ultra-Lightweight Block Cipher”. In: *Cryptographic Hardware and Embedded Systems - CHES 2007*. 2007, pp. 450–466.

[26] Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. “Integral and Multidimensional Linear Distinguishers with Correlation Zero”. In: *Advances in Cryptology - ASIACRYPT 2012*. 2012, pp. 244–261.

[27] Andrey Bogdanov and Christian Rechberger. “A 3-Subset Meet-in-the-Middle Attack: Cryptanalysis of the Lightweight Block Cipher KTANTAN”. In: *Selected Areas in Cryptography, SAC 2010*. 2010, pp. 229–240.

[28] Andrey Bogdanov and Vincent Rijmen. “Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers”. In: *Designs, Codes and Cryptography* 70.3 (2014), pp. 369–383.

[29] Andrey Bogdanov and Elmar Tischhauser. “On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui’s Algorithm 2”. In: *Fast Software Encryption, FSE 2013*. 2013, pp. 19–38.

[30] Andrey Bogdanov and Meiqin Wang. “Zero Correlation Linear Cryptanalysis with Reduced Data Complexity”. In: *Fast Software Encryption, FSE 2012*. 2012, pp. 29–48.

[31] Florent Chabaud and Serge Vaudenay. “Links Between Differential and Linear Cryptanalysis”. In: *Advances in Cryptology - EUROCRYPT ’94*. 1994, pp. 356–365.

[32] David Chaum and Jan-Hendrik Evertse. “Crytanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers”. In: *Advances in Cryptology - CRYPTO ’85*. 1985, pp. 192–211.

[33] Huiju Cheng, Howard M. Heys, and Cheng Wang. “PUFFIN: A Novel Compact Block Cipher Targeted to Embedded Digital Systems”. In: *11th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2008*. 2008, pp. 383–390.

[34] Joo Yeon Cho. “Linear Cryptanalysis of Reduced-Round PRESENT”. In: *Topics in Cryptology - CT-RSA 2010*. 2010, pp. 302–317.

[35] Joo Yeon Cho, Miia Hermelin, and Kaisa Nyberg. “A New Technique for Multidimensional Linear Cryptanalysis with Applications on Reduced Round Serpent”. In: *Information Security and Cryptology - ICISC 2008*. 2008, pp. 383–398.

[36] Joo Yeon Cho and Kaisa Nyberg. “Improved Linear Cryptanalysis of SMS4 Block Cipher”. In: *Symmetric Key Encryption Workshop*. 2011, pp. 1–14.

[37] Nicolas Courtois. “Feistel Schemes and Bi-Linear Cryptanalysis”. In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 23–40.

[38] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. “Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations”. In: *Advances in Cryptology - EUROCRYPT 2000.* 2000, pp. 392–407.

[39] Nicolas Courtois and Josef Pieprzyk. “Cryptanalysis of Block Ciphers with Overdefined Systems of Equations”. In: *Advances in Cryptology - ASIACRYPT 2002.* 2002, pp. 267–287.

[40] Joan Daemen. *Cipher and Hash Function Design, Strategies Based on Linear and Differential Cryptanalysis, PhD Thesis.* http://jda.noekeon.org/. K.U.Leuven, 1995.

[41] Joan Daemen, René Govaerts, and Joos Vandewalle. “A New Approach to Block Cipher Design”. In: *Fast Software Encryption, 1993.* 1993, pp. 18–32.

[42] Joan Daemen, René Govaerts, and Joos Vandewalle. “Correlation Matrices”. In: *Fast Software Encryption, 1994.* 1994, pp. 275–285.

[43] Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. “Xoodoo Cookbook”. In: *IACR Cryptology ePrint Archive* 2018 (2018), p. 767.

[44] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. “The Block Cipher Square”. In: *Fast Software Encryption, FSE ’97.* 1997, pp. 149–165.

[45] Joan Daemen and Vincent Rijmen. “Correlation Analysis in GF(2^n)”. In: *Advanced Linear Cryptanalysis of Block and Stream Ciphers.* 2011, pp. 115–131.

[46] Joan Daemen and Vincent Rijmen. “Probability Distributions of Correlation and Differentials in Block Ciphers”. In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 221–242.

[47] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2.

[48] Whitfield Diffie and Martin E Hellman. “Privacy and Authentication: An Introduction to Cryptography”. In: *Proceedings of the IEEE* 67.3 (1979), pp. 397–427.

[49] Whitfield Diffie and Martin E. Hellman. “Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard”. In: *IEEE Computer* 10.6 (1977), pp. 74–84.

[50] Itai Dinur and Adi Shamir. “Cube Attacks on Tweakable Black Box Polynomials”. In: *Advances in Cryptology - EUROCRYPT 2009.* 2009, pp. 278–299.

[51] Morris J Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC.* Tech. rep. 2007.

[52] William F Ehrsam, Carl HW Meyer, John L Smith, and Walter L Tuchman. *Message Verification and Transmission Error Detection by Block Chaining.* US Patent 4,074,066. 1978.

[53] Let's Encrypt. *Percentage of Web Pages Loaded by Firefox Using HTTPS.* `https://letsencrypt.org/stats/`. Retrived on 21/09/2018.

[54] Dave Evans. "The Internet of Things: How the Next Evolution of the Internet is Changing Everything". In: *CISCO white paper* 1.2011 (2011), pp. 1–11.

[55] PUB FIPS. "46-3: Data Encryption Standard (DES)". In: *National Institute of Standards and Technology* 25.10 (1999), pp. 1–22.

[56] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. "Grøstl - A SHA-3 Candidate". In: *Symmetric Cryptography, 11.01. - 16.01.2009.* 2009.

[57] Carlo Harpes, Gerhard G. Kramer, and James L. Massey. "A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-Up Lemma". In: *Advances in Cryptology - EUROCRYPT '95.* 1995, pp. 24–38.

[58] Carlo Harpes and James L. Massey. "Partitioning Cryptanalysis". In: *Fast Software Encryption, FSE '97.* 1997, pp. 13–27.

[59] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. "Multidimensional Linear Cryptanalysis of Reduced Round Serpent". In: *Information Security and Privacy, ACISP 2008.* 2008, pp. 203–215.

[60] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. "Capacity and Data Complexity in Multidimensional Linear Attack". In: *Advances in Cryptology - CRYPTO 2015.* 2015, pp. 141–160.

[61] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, and E Kobayashi. *CLOC and SILC v3.* 2016.

[62] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. "Linear Cryptanalysis Using Multiple Approximations". In: *Advances in Cryptology - CRYPTO '94.* 1994, pp. 26–39.

[63] Pascal Junod. "On the Complexity of Matsui's Attack". In: *Selected Areas in Cryptography, SAC 2001.* 2001, pp. 199–211.

[64] Pascal Junod and Serge Vaudenay. "Optimal Key Ranking Procedures in a Statistical Cryptanalysis". In: *Fast Software Encryption, FSE 2003.* 2003, pp. 235–246.

[65] John Kelsey, Bruce Schneier, and David A. Wagner. "Mod n Cryptanalysis, with Applications Against RC5P and M6". In: *Fast Software Encryption, FSE '99.* 1999, pp. 139–155.

[66] Auguste Kerckhoffs. "La Cryptographie Militaire". In: *Journal des Sciences Militaires* IX (1883).

[67] Lars Knudsen. “DEAL – A 128-bit Block Cipher”. In: *complexity* 258.2 (1998), p. 216.

[68] Lars R. Knudsen. “Truncated and Higher Order Differentials”. In: *Fast Software Encryption, 1994*. 1994, pp. 196–211.

[69] Lars R. Knudsen and Matthew Robshaw. *The Block Cipher Companion*. Information Security and Cryptography. Springer, 2011.

[70] Lars R. Knudsen and Matthew J. B. Robshaw. “Non-Linear Approximations in Linear Cryptanalysis”. In: *Advances in Cryptology - EUROCRYPT ’96*. 1996, pp. 224–236.

[71] Lars R. Knudsen and David A. Wagner. “Integral Cryptanalysis”. In: *Fast Software Encryption, FSE 2002*. 2002, pp. 112–127.

[72] Xuejia Lai. “Higher Order Derivatives and Differential Cryptanalysis”. In: *Communications and Cryptography*. Springer, 1994, pp. 227–233.

[73] Susan K. Langford and Martin E. Hellman. “Differential-Linear Cryptanalysis”. In: *Advances in Cryptology - CRYPTO ’94*. 1994, pp. 17–25.

[74] Daniel Lazard. “Gröbner-Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations”. In: *Computer Algebra, EUROCAL ’83*. 1983, pp. 146–156.

[75] Gregor Leander. “On Linear Hulls, Statistical Saturation Attacks, PRESENT and a Cryptanalysis of PUFFIN”. In: *Advances in Cryptology - EUROCRYPT 2011*. 2011, pp. 303–322.

[76] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. “A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack”. In: *Advances in Cryptology - CRYPTO 2011*. 2011, pp. 206–221.

[77] Stefan Lucks. “The Saturation Attack - A Bait for Twofish”. In: *Fast Software Encryption, FSE 2001*. 2001, pp. 1–15.

[78] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology - EUROCRYPT ’93*. 1993, pp. 386–397.

[79] Mitsuru Matsui. “The First Experimental Cryptanalysis of the Data Encryption Standard”. In: *Advances in Cryptology - CRYPTO ’94*. 1994, pp. 1–11.

[80] Mitsuru Matsui and Atsuhiro Yamagishi. “A New Method for Known Plaintext Attack of FEAL Cipher”. In: *Advances in Cryptology - EUROCRYPT ’92*. 1992, pp. 81–91.

[81] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN: 0-8493-8523-7.

[82] Ralph Merkle. “Secrecy, Authentication, and Public Key Systems”. In: *Ph. D. Thesis, Stanford University* (1979).

[83] Kazuhiko Minematsu. “AES-OTR v3. 1”. In: *NEC Corporation, Japan. Submission to CAESAR* (2016).

[84] Judy H. Moore and Gustavus J. Simmons. “Cycle Structures of the DES with Weak and Semi-Weak Keys”. In: *Advances in Cryptology - CRYPTO ’86.* 1986, pp. 9–32.

[85] S. Murphy. “The Independence of Linear Approximations in Symmetric Cryptanalysis”. In: *IEEE Transactions on Information Theory* 52.12 (2006), pp. 5510–5518.

[86] Kaisa Nyberg. “Linear Approximation of Block Ciphers”. In: *Advances in Cryptology - EUROCRYPT ’94.* 1994, pp. 439–444.

[87] Kaisa Nyberg. “Statistical and Linear Independence of Binary Random Variables”. In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 432.

[88] Luke O’Connor. “Properties of Linear Approximation Tables”. In: *Fast Software Encryption 1994.* 1994, pp. 131–136.

[89] Kenji Ohkuma. “Weak Keys of Reduced-Round PRESENT for Linear Cryptanalysis”. In: *Selected Areas in Cryptography, SAC 2009.* 2009, pp. 249–265.

[90] Matthew G Parker. “Generalised S-box Nonlinearity”. In: *NESSIE Public Document-NES/DOC/UIB/WP5/020/A* (2003).

[91] Phillip Rogaway, Mihir Bellare, and John Black. “OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption”. In: *ACM Transactions on Information and System Security* 6.3 (2003), pp. 365–403.

[92] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseth. “Yoyo Tricks with AES”. In: *Advances in Cryptology - ASIACRYPT 2017.* 2017, pp. 217–243.

[93] Ali Aydin Selçuk. “On Probability of Success in Linear and Differential Cryptanalysis”. In: *Journal of Cryptology* 21.1 (2008), pp. 131–147.

[94] Claude E Shannon. *A Mathematical Theory of Cryptography.* https://www.iacr.org/museum/shannon45.html. 1945.

[95] Claude E Shannon. “Communication Theory of Secrecy Systems”. In: *Bell System Technical Journal* 28.4 (1949), pp. 656–715.

[96] Takeshi Shimoyama and Toshinobu Kaneko. “Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES”. In: *Advances in Cryptology - CRYPTO ’98.* 1998, pp. 200–211.

[97] Bing Sun, Zhiqiang Liu, Vincent Rijmen, Ruilin Li, Lei Cheng, Qingju Wang, Hoda AlKhzaimi, and Chao Li. “Links Among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis”. In: *Advances in Cryptology - CRYPTO 2015.* 2015, pp. 95–115.

[98] Yosuke Todo. “Structural Evaluation by Generalized Integral Property”. In: *Advances in Cryptology - EUROCRYPT 2015.* 2015, pp. 287–314.

[99] Yosuke Todo, Gregor Leander, and Yu Sasaki. “Nonlinear Invariant Attack - Practical Attack on Full SCREAM, iSCREAM, and Midori64”. In: *Advances in Cryptology - ASIACRYPT 2016*. 2016, pp. 3–33.

[100] David A. Wagner. “The Boomerang Attack”. In: *Fast Software Encryption, FSE ’99*. 1999, pp. 156–170.

[101] Lei Wei, Christian Rechberger, Jian Guo, Hongjun Wu, Huaxiong Wang, and San Ling. “Improved Meet-in-the-Middle Cryptanalysis of KTANTAN”. In: *Information Security and Privacy, ACISP 2011*. 2011, pp. 433–438.

[102] D Whiting, R Housley, and N Ferguson. “RFC 3610, Counter with CBC-MAC (CCM)”. In: *Internet Engineering Task Force* (2003).

[103] Robert S. Winternitz. “A Secure One-Way Hash Function Built from DES”. In: *Proceedings of the 1984 IEEE Symposium on Security and Privacy*. 1984, pp. 88–90.

[104] Hongjun Wu and Tao Huang. “The JAMBU Lightweight Authentication Encryption Mode (v2. 1)”. In: *Submitted to the CAESAR competition (September 2016)* (2016).

[105] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. “RECTANGLE: A Bit-Slice Lightweight Block Cipher Suitable for Multiple Platforms”. In: *SCIENCE CHINA Information Sciences* 58.12 (2015), pp. 1–15.

Part II

Publications

Publication 1

Linear Cryptanalysis of DES with Asymmetries

Publication Information

Andrey Bogdanov and Philip S. Vejre. "Linear Cryptanalysis of DES with Asymmetries". In: *Advances in Cryptology - ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 187–216. ISBN: 978-3-319-70693-1

Contribution

- Main author.

Remarks

This publication has been slightly edited to fit the format.

Linear Cryptanalysis of DES with Asymmetries

Andrey Bogdanov and Philip S. Vejre

Technical University of Denmark

Abstract. Linear cryptanalysis of DES, proposed by Matsui in 1993, has had a seminal impact on symmetric-key cryptography, having seen massive research efforts over the past two decades. It has spawned many variants, including multidimensional and zero-correlation linear cryptanalysis. These variants can claim best attacks on several ciphers, including PRESENT, Serpent, and CLEFIA. For DES, none of these variants have improved upon Matsui's original linear cryptanalysis, which has been the best known-plaintext key-recovery attack on the cipher ever since. In a revisit, Junod concluded that when using 2^{43} known plaintexts, this attack has a complexity of 2^{41} DES evaluations. His analysis relies on the standard assumptions of right-key equivalence and wrong-key randomisation.

In this paper, we first investigate the validity of these fundamental assumptions when applied to DES. For the right key, we observe that strong linear approximations of DES have more than just one dominant trail and, thus, that the right keys are in fact *inequivalent* with respect to linear correlation. We therefore develop a new right-key model using Gaussian mixtures for approximations with several dominant trails. For the wrong key, we observe that the correlation of a strong approximation after the partial decryption with a wrong key still shows much *non-randomness.* To remedy this, we propose a novel wrong-key model that expresses the wrong-key linear correlation using a version of DES with more rounds. We extend the two models to the general case of multiple approximations, propose a likelihood-ratio classifier based on this generalisation, and show that it performs better than the classical Bayesian classifier.

On the practical side, we find that the distributions of right-key correlations for multiple linear approximations of DES exhibit exploitable *asymmetries.* In particular, not all sign combinations in the correlation values are possible. This results in our improved multiple linear attack on DES using 4 linear approximations at a time. The lowest computational complexity of $2^{38.86}$ DES evaluations is achieved when using $2^{42.78}$ known plaintexts. Alternatively, using 2^{41} plaintexts results in a computational complexity of $2^{49.75}$ DES evaluations. We perform practical experiments to confirm our model. To our knowledge, this is the best attack on DES.

1 Introduction

Accepted as a standard in 1976 by the National Bureau of Standards (later NIST), DES can now celebrate its fortieth birthday. Being a highly influential cipher, it has inspired much cryptanalysis. Triple-DES is still massively deployed in conservative industries such as banking. Moreover, it is used to secure about 3% of Internet traffic [26].

The first attack on the full DES came in 1992, where Biham and Shamir demonstrated that *differential cryptanalysis* enabled a key recovery using 2^{47} *chosen plaintexts* in time 2^{37} [1]. The year after, in 1993, Matsui introduced a new cryptanalytic technique, *linear cryptanalysis*, which DES proved especially susceptible to. While the first iteration of the attack required 2^{47} *known plaintexts* [20], Matsui soon improved his attack to only require 2^{43} known texts, taking 2^{43} time to recover the key. This complexity estimate was lowered to 2^{41} by Junod in 2001 [16]. In [17], Knudsen and Mathiassen lower the complexity to 2^{42} plaintexts, however this attack uses *chosen plaintexts*.

In this paper we present the first successful attack on full DES using multiple linear approximations. By developing new models for the correlation distributions, and by exploiting asymmetries in the right-key distribution, we obtain an improved key-recovery attack. Using $2^{42.78}$ known plaintexts, the attack recovers the key in time equal to $2^{38.86}$ DES encryptions.

1.1 Previous Work and Problems

Linear cryptanalysis has proven to be widely applicable, and has spawned many variants and generalisations. Amongst them are differential-linear cryptanalysis [18], multiple linear cryptanalysis [2, 15], multidimensional linear cryptanalysis [13, 14], zero-correlation linear cryptanalysis [4, 5], multivariate linear cryptanalysis [7], etc. These techniques have successfully been applied to a wide range of ciphers, including Serpent [14, 22], PRESENT [7, 8], Camellia and CLEFIA [3], and CAST-256 [27].

Matsui first introduced the concept of a *linear approximation* of a block cipher in [20]. If we denote the encryption of a plaintext $\mathcal{P}$ using key K by $\mathcal{C} = E_K(\mathcal{P})$, then a linear approximation of this cipher is a pair of masks, (α, β), which indicate some bits of the plaintext and ciphertext. The idea is to find α and β such that the sum of plaintext bits indicated by α is strongly correlated to the sum of ciphertext bits indicated by β. A measure of the strength of a linear approximation is the *linear correlation*, defined by

$$C_K(\alpha, \beta) = 2 \cdot \Pr(\langle \alpha, x \rangle \oplus \langle \beta, E_K(x) \rangle = 0) - 1,$$

where $\langle \cdot, \cdot \rangle$ is the canonical inner product. Matsui showed how an approximation with linear correlation that deviates significantly from zero can be used to attack the cipher, and found such approximations for DES. The attack procedure was formalised as Algorithm 2, in which an attacker obtains plaintext-ciphertext pairs over r rounds of a cipher. The attacker then guesses the outer round keys in order to encrypt/decrypt the outer rounds, and compute the correlation over $r-2$ rounds.

Standard assumptions for linear cryptanalysis on DES

In [16] Junod revisited Matsui's attack, and concluded that Matsui's original complexity was slightly overestimated. Junod instead estimated that the attack could be performed in time 2^{41} using the same number of known plaintexts. Central to both Matui's and Junod's analysis are two assumptions.

Assumption A (Right-Key Equivalence)**.** *For a linear approximation (α, β), the magnitude of the correlation, $|C_K(\alpha, \beta)|$, does not deviate significantly from its expected value over all keys, that is, $|C_K(\alpha, \beta)| = \mathrm{E}(|C_K(\alpha, \beta)|)$.*

Problem 1: Insufficient Right-Key Distribution: The assumption of right-key equivalence is usually the result of assuming that the magnitude of the linear correlation is determined by a single dominant trail. This further implies that the linear correlation only takes on two values over the key space. However, in [23], Nyberg first introduced the concept of a *linear hull*, i.e. the collection of all trails of a linear approximation, and showed that Assumption A is not true in general. In [6], Bogdanov and Tischhauser gave a refined version of Assumption A, which takes a larger part of the hull into account. However, to the best of our knowledge, no thorough exploration of the right-key distribution for DES has been conducted, and it is unclear how accurate Assumption A is in this context.

Assumption B (Wrong-Key Randomisation)**.** *In the context of Algorithm 2, the correlation of a linear approximation (α, β) is equal to 0 for all wrong guesses of the outer round keys.*

Problem 2: Unrealistic Wrong-Key Distribution: The assumption of wrong-key randomisation implies that if an attacker guesses the wrong outer round keys in Algorithm 2, the resulting texts pairs behave in a completely random way, i.e. the linear correlation will be equal to zero. A refined version of this assumption was given by Bogdanov and Tischhauser in [6], where the wrong-key distribution was given as the Gaussian distribution $\mathcal{N}(0, 2^{-n})$, where n is the block size. This distribution matches that of an ideal permutation. Neither of these assumptions have been verified for DES. Indeed, DES exhibits very strong linear approximations, and it is not clear if a wrong key guess is sufficient to make the linear correlation close to that of an ideal permutation.

Linear cryptanalysis of DES with multiple approximations

While several models for using multiple approximations for linear cryptanalysis have been proposed, see e.g. [2, 7, 13, 14, 15, 25], the application to DES has been very limited. In [15], Kaliski and Robshaw specifically note that their approach is limited when applied to DES. In [25], Semaev presents an alternative approach, but does not obtain better results than Matsui's original attack.

The most promising attempt was given in [2] by Biryukov et al. Under Assumption A, when using M approximations, the key space can be partitioned into at most 2^M key classes based on the signs of the M linear correlations. This allowed Biyukov et al. to describe the correlation of each key class as an M-variate normal distribution $\mathcal{N}_M(\boldsymbol{\mu}_i, 1/N \cdot \mathbf{I})$, where $\mathbf{I}$ is an $M \times M$ identity matrix, and the mean vector is given by

$$\boldsymbol{\mu}_i = (s_{i,1}|C_K(T_1)|, \ldots, s_{i,M}|C_K(T_M)|)^\top,$$

where $s_{i,j} \in \{-1, 1\}$ describes the sign combination of the i'th key class. Based on this, they developed a Bayesian classifier, in order to decide between a correct or incorrect guess of the round keys in Algorithm 2.

Problem 3: Applying Multiple Linear Cryptanalysis to DES: While Biryukov et al. demonstrate that their method of using multiple approximations can potentially reduce the complexity of Matsui's attack, they also note that the structure of DES makes it difficult to arbitrarily use a large number of approximations. As such, they did not present a new attack on DES. Similar observations were made by Kaliski and Robshaw in [15]. To the best of our knowledge, no other variants of linear cryptanalysis which uses multiple approximations have been able to outperform Matsui's original attack.

1.2 Our Contributions

More Accurate Right-Key Model for DES.

In Section 3 we consider Problem 1, i.e. the fundamental problem of the DES right-key distribution. We enumerated over 1000 trails for the linear approximation used by Matsui, and calculated the resulting correlation distribution for 1 million keys. We demonstrate in Section 3.2 that while this distribution does have two modes symmetric around zero, each mode does not consist of a single value, as predicted by Assumption A. Indeed, it is not even the case that each mode takes on a simple Gaussian distribution. As such, one cannot consider different keys to have equivalent behaviour.

We therefore develop a new model for the right-key distribution in Section 3.3. This model is given below, and expresses the distribution as a mixture of Gaussian components. An example of this model applied to DES is shown in Figure 1.

Model A (Right-Key Equivalence for One Approximation). *Consider a linear approximation (α, β) of r rounds of DES. The distribution of the linear correlation $C_K(\alpha, \beta)$ over the key space is approximately given by a Gaussian mixture for some weights λ_i and components $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, \ldots, \ell$.*

Applying this model to the approximations used by Matsui, we show that it is able to accurately describe the observed distribution. Moreover, it is interesting to note

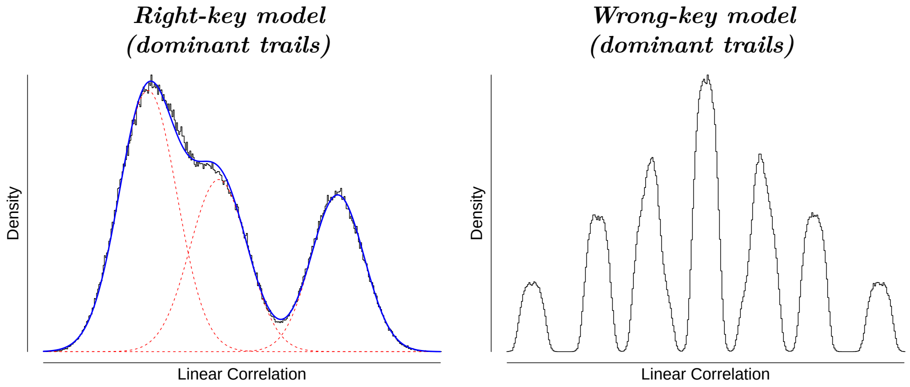


Figure 1: Our new models for the distributions of linear correlation over the key space for DES. The distributions are expressed as Gaussian mixtures. The model shows a deviation from the standard assumptions of right-key equivalence and wrong-key randomisation.

that the component associated with the dominant trail *only accounts for 30% of the correlation, contrasting Assumption A*. We furthermore apply the mixture model to describe the full correlation distribution observed during an attack. We note that when the number of texts used in the attack is small, the right-key distribution originally given by Matsui is a good approximation. However, we stress that the cryptanalyst should carefully examine the right-key distribution when this is not the case.

New Wrong-Key Model for DES.

In Section 4 we consider Problem 2. In order to obtain a wrong-key model that more accurately describes the case of a wrong key guess in Algorithm 2, we propose the following new approach.

Model B (Non-Random Wrong-Key Distribution)**.** *Consider an Algorithm 2 style attack on r rounds of DES using a linear approximation (α, β) over $r-2$ rounds. Let R_K be the keyed round function of DES, and let $E_K^\star$ denote the r-round encryption function. For a wrong guess of the outer round keys, the correlation will be distributed as for the cipher*

$$E'_K(x) = R_{K_a}^{-1}(E_K^\star(R_{K_b}^{-1}(x))), \tag{1}$$

where K_a and K_b are chosen uniformly at random.

This model accurately matches the situation of guessing the wrong outer round keys in an Algorithm 2 attack. We enumerated over 900 trails for the linear approximation used by Matsui for the cipher E', and calculated the resulting correlation distribution for 1 million keys. The result is shown in Figure 1. While the distribution has mean zero, the shape of the distribution does not match Assumption B, nor that of the revised version by Bogdanov and Tischhauser, as its variance is much larger than 2^{-n}. As is the case for the right-key distribution, the wrong-key distribution is also not a simple Gaussian, but rather some Gaussian mixture. Again, for low data complexities, we demonstrate that a Gaussian model is sufficient to describe the wrong-key distribution observed during an attack, but advise caution when the data complexity is close to full codebook.

Multiple Linear Cryptanalysis with Asymmetries.

In Sections 5 and 6 we remedy Problem 3. We develop a classifier for M approximations based on the likelihood-ratio of the right-key and wrong-key distributions developed in Section 3 and Section 4. This classifier is given by

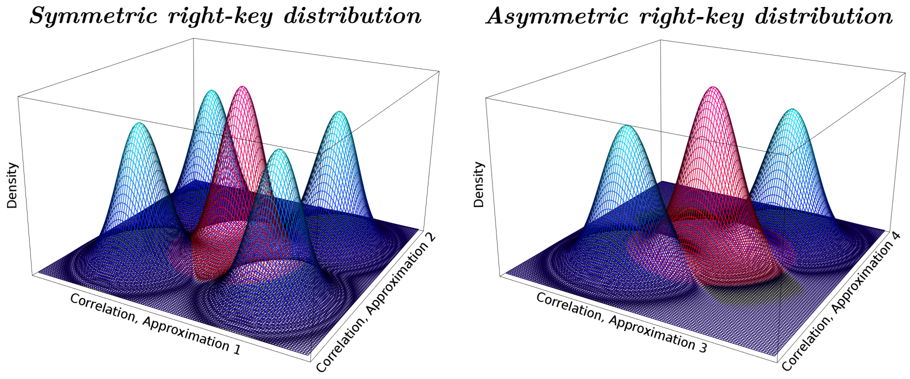
$$\Lambda(\mathbf{x}) = \frac{\sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \mathbf{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I})},$$

where ϕ is the probability density function (PDF) of the Gaussian distribution. The wrong-key distribution is a simple M-variate Gaussian. The right-key distribution is a mixture of at most 2^M, M-variate components based on the signs of the M correlations. In contracts to the work in [2], we do not partition the key space, but express the correlation distribution over the entire key space. Also in contrast to this work, our classifier directly takes the wrong-key distribution into account. We demonstrate how this improves the classifier.

We make the interesting observation that if the right-key distribution is asymmetric, that is, if the number of components is less than 2^M, we obtain a stronger classifier. This situation is demonstrated in Figure 2. From this example, one can get an intuitive understanding of how an asymmetric distribution makes it easier to distinguish between right-key and wrong-key. We therefore propose the term *symmetry factor*, namely the ratio between number of components and 2^M, and conjecture that a lower symmetry factor will result in a stronger attack.

First Successful Multiple Linear Cryptanalysis of DES.

By using the asymmetric classifier in Section 6, we give the first attack on full DES using multiple linear approximations which improves Matsui's original attack. We use two sets of four linear approximations. Using $2^{42.78}$ known plaintexts, the attack recovers the key in time equal to $2^{38.86}$ encryptions, with a success probability of 85%. This is 4.4 times faster than Junod's estimate of Matsui's attack, and uses $2^{40.2}$ fewer texts. We confirm these results by measuring the actual correlation distributions using this number of texts for 1300 random keys, and computing the resulting advantage of our classifier. We find that the model fits our practical results very well. Alternatively, we can lower the data complexity to 2^{41}, and recover the



Symmetric right-key distribution

Asymmetric right-key distribution

Figure 2: An illustration of the difference between a symmetric and an asymmetric joint distribution of linear correlation for two approximations over the key space. The right-key distribution is blue, while the wrong-key distribution is red.

Technique	**Data complexity**	**Time complexity**	**Success probability**	**Attack scenario**	**Source**
Differential	$2^{47.00}$	$2^{37.00}$	58%	CP	[1]
Linear	$2^{43.00}$	$2^{43.00}$	85%	KP	[21]
Linear	$2^{43.00}$	$2^{41.00}$	85%	KP	[16]
Multiple Linear	$2^{42.78}$	$2^{38.86}$	85%	KP	Sec. 6
Multiple Linear	$2^{41.00}$	$2^{49.76}$	80%	KP	Sec. 6

Table 1: Comparison of key-recovery attacks on full DES. Kown plaintext and chosen plaintext attacks are referred to as KP and CP, respectively.

key in time $2^{49.76}$, with a success probability of 80%. Our attack is compared to previous attacks on full DES in Table 1.

2 Linear Cryptanalysis of DES

In 1993, Matsui introduced the concept of linear cryptanalysis and applied it to derive a key-recovery attack on the full 16-round DES [20, 21]. In this section, we briefly outline the attack. We then give an overview of the assumptions Matsui made in his analysis, and show the resulting complexity of the attack. Moreover, we show a variant of linear cryptanalysis due to Biryukov, de Cannière, and Quisquater [2], which will be important for the remaining part of this work.

2.1 Basics of Linear Cryptanalysis

We consider a block cipher with block length n and key length κ. We denote the encryption of plaintext $\mathcal{P} \in \mathbb{F}_2^n$ under key $K \in \mathbb{F}_2^\kappa$ by $E_K(\mathcal{P})$. The idea of linear cryptanalysis is to find a *linear approximation* $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ such that the magnitude of its *linear correlation*, defined by

$$C_K(\alpha, \beta) = 2 \cdot \Pr(\langle \alpha, x \rangle \oplus \langle \beta, E_K(x) \rangle = 0) - 1,$$

is large. Here, $\langle \cdot, \cdot \rangle$ denotes the canonical inner product on $\mathbb{F}_2^n$. Thus, the correlation is a measure of how often the parity bit $\langle \alpha, \mathcal{P} \rangle$ of the plaintext is equal to the parity bit $\langle \beta, \mathcal{C} \rangle$ of the ciphertext. We expect a strong cipher to only have approximations with linear correlation close to 0, and hence a correlation value that deviates significantly from 0 indicates a weakness of the cipher.

For Feistel ciphers, such as DES, the linear correlation of an approximation (α, β) can be calculated by considering so called *linear trails* of the cipher. We define a single-round linear trail of DES as the triple $(u, t, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^m \times \mathbb{F}_2^n$, where m is the size of a single round key. The linear correlation of this single-round trail is then defined as

$$C_{K_r}(u, t, v) = 2 \cdot \Pr(\langle u, x \rangle \oplus \langle v, R_{K_r}(x) \rangle = \langle t, K_r \rangle) - 1,$$

where R_{K_r} is the DES round-function using the r'th round key K_r. We now define a linear trail T over r rounds as a collection of single-round trails (u_i, t_i, u_{i+1}), $i = 0, \ldots, r-1$, as well as the *correlation contribution* of the trail T as [10, 12]

$$C_K(T) = \prod_{i=0}^{r-1} C_{K_i}(u_i, t_i, u_{i+1}).$$

We will also make use of the concept of an associated *key trail* $\bar{T}$ of a trail T. The key trail is defined as the concatenation of the t_i, $i = 0, \ldots, r-1$.

Daemen and Rijmen demonstrated that the correlation contribution of a trail can be written as [10, 12]

$$C_K(T) = (-1)^{s_T \oplus \langle \bar{T}, \bar{K} \rangle} |C_K(T)|, \tag{2}$$

where s_T is a sign bit specific to the trail T, and $\bar{K}$ denotes the concatenation of the round keys K_i. Moreover, under the assumption of independent round keys, $|C_K(T)|$ is independent of the key. Thus, the correlation contribution of a trail T has a fixed magnitude for all keys, but the sign is determined by the round key bits indicated by the key trail $\bar{T}$. Finally, Daemen and Rijmen give the correlation over all r rounds for some approximation (α, β) as [10, 12]

$$C_K(\alpha, \beta) = \sum_{u_0=\alpha, u_r=\beta} C_K(T) = \sum_{u_0=\alpha, u_r=\beta} (-1)^{s_T \oplus \langle \bar{T}, \bar{K} \rangle} |C_K(T)|, \tag{3}$$

i.e. the sum of the correlation contributions of all trails from α to β.

2.2 Matsui's approach

Matsui's key observation was that DES exhibits linear trails where the correlation contribution deviates significantly from zero. Consider the full 16-round DES, let $\mathcal{P}$ be the plaintext, and let $\mathcal{C}$ be the ciphertext. Let $[i_0, \ldots, i_\ell]$ denote an element in $\mathbb{F}_2^n$ whose i_j'th components are 1, $j = 0, \ldots, \ell$, while all other components are 0. Then, over 14 rounds of DES, the approximations

$$\gamma_1 = ([7, 18, 24], [7, 18, 24, 29, 47]) \quad \text{and} \quad \delta_3 = ([15, 39, 50, 56, 61], [39, 50, 56]),$$

both have trails with correlation contribution $C_K(T) = \pm 2^{-19.75}$ [21]. From Equation 2 we can determine one bit of information if we know the sign of $C_K(T)$, namely the parity $\langle \bar{T}, \bar{K} \rangle$ of the round key bits indicated by the key trail $\bar{T}$. Let k_f denote the key-bits of round key K_0 required to partially encrypt a plaintext $\mathcal{P}$ one round and calculate $\langle \alpha, R_{K_0}(\mathcal{P}) \rangle$, and let k_b denote the key-bits of round key K_{r-1} required to partially decrypt the ciphertext $\mathcal{C}$ one round and calculate $\langle \beta, R^{-1}_{K_{r-1}}(\mathcal{C}) \rangle$. Matsui developed the following general approach in order to determine $|k_f| + |k_b| + 1$ key bits, formalised as Algorithm 2.

Algorithm 2

1. Obtain N plaintext-ciphertext pairs.
2. For each guess of the key-bits (k_f, k_b), partially encrypt, respectively decrypt, each plaintext-ciphertext pair $(\mathcal{P}, \mathcal{C})$ and calculate the number of times L_i the input parity $\langle \alpha, R_{R_0}(\mathcal{P}) \rangle$ is equal to the output partiy $\langle \beta, R^{-1}_{R_{r-1}}(\mathcal{C}) \rangle$ for the i'th guess, $i = 1, \ldots, 2^{|k_f| + |k_b|}$.
3. For each counter L_i, if $L_i > N/2$, guess that the sign bit $\langle \bar{T}, \bar{K} \rangle = s_T$, otherwise guess that $\langle \bar{T}, \bar{K} \rangle = s_T \oplus 1$.
4. For any counter L_i with $|T_i - N/2| > \Gamma$, for a predetermined value Γ, guess the remaining $\kappa - (|k_f| + |k_b| + 1)$ bits of the master key K, and determine the correct value of K through trial encryption.

For his attack on DES, Matsui performed Algorithm 2 once for γ_1 and once for δ_3, determining 26 bits before guessing the remaining 30 bits of K. In his analysis of the success rate and complexity of the attack, Matsui assumed that the linear correlation of the approximations γ_1 and δ_3 were only determined by a single trail T. The idea is that the correlation contribution of T is much larger than that of all other trails – a so called *dominant trail*. We will call the associated key trail $\bar{T}$ of such a trail a *dominant key trail*. In the presence of such a dominant trail, $C_K(\alpha, \beta)$ only takes on two values over the key space. This can be seen from Equation 3, as the case of a dominant trail implies that this sum only has one term. Under this assumption, Matsui concluded that when using 2^{43} texts, there is an 85% probability of recovering the key at a time complexity of 2^{43} DES encryptions. In a later analysis of Matsui's attack [16], Junod concluded that the actual computational complexity is closer to 2^{41} DES encryptions.

2.3 Biryukov et al. – Multiple Approximations

A natural extension of Matsui's linear cryptanalysis is to attempt to use multiple linear approximations simultaneously. The first attempt at developing such a framework was by Kaliski and Robshaw in [15]. This work has the limitation that all linear approximations must have the same dominant key trail, and the approximations were assumed to be statistically independent. Moreover, as Kaliski and Robshaw note, the application of this method to DES is very limited.

Another approach was undertaken by Biryukov et al. in [2]. Here, the approximations can in principle be picked arbitrarily, but the framework still requires the assumption of one dominant trail for each approximation, and independence between approximations. Due to these restrictions, the foundations of multidimensional linear cryptanalysis was developed in e.g. [13, 14]. While this approach has been applied with great success to a large range of ciphers, no results have been shown on DES. Thus, Matsui's single linear cryptanalysis still provides the best results on this cipher.

Let us briefly reconsider the method by Biryukov et al., assuming the use of M linear approximations. The idea is to partition the key space into at most 2^M classes based on the parity of the $\langle \bar{T}_i, \bar{K} \rangle$, where $\bar{T}_i$ is the dominant key trail of the i'th approximation. An Algorithm 2 type attack is then performed: For each guess of the key-bits (k_f, k_b), the vector $(L_{i,1}, \ldots, L_{i,M})$ is calculated, and the likelihood of that vector belonging to each of the key classes is computed. The right guess of (k_f, k_b) should yield one class with high likelihood, and the class then indicates at most M parity bits, $\langle \bar{T}_i, \bar{K} \rangle$. Central to the analysis of [2] are the following two assumptions:

Assumption 1 (Right-Key Equivalence)**.** *For a linear approximation (α, β), the magnitude of the correlation, $|C_K(\alpha, \beta)|$, does not deviate significantly from its expected value over all keys, that is, $|C_K(\alpha, \beta)| = \mathrm{E}(|C_K(\alpha, \beta)|)$.*

Assumption 2 (Wrong-Key Randomisation)**.** *For Algorithm 2, the correlation of a linear approximation (α, β) is 0 for all wrong guesses of (k_f, k_b).*

The assumption of right-key equivalence implies that the linear approximation has one dominant trail, say T, and consequently the distribution of the correlation over the key space only takes on two values, namely $\pm|C_K(T)|$. Thus, the natural partitioning of the key space for M approximations is the partitioning induced by the sign of the correlations, i.e. the vector $((-1)^{\langle \bar{T}_1, \bar{K} \rangle}, \ldots, (-1)^{\langle \bar{T}_M, \bar{K} \rangle})$. In practice however, the correlations are calculated from the counters $L_{i,j}$. The joint distribution of the resulting measured correlations, for some specific key class, is given in [2] as an M-variate normal distribution, described in the following model.

Model 1 (Right-Key Partitioning for Multiple Approximations [2])**.** *Consider a set of linear approximations $(\alpha_1, \beta_1), \ldots, (\alpha_M, \beta_M)$ of r rounds of DES. Then, the key space can be partitioned into at most 2^M key classes based on the signs of the correlations. The undersampled distribution of the linear correlation vector, using N texts and restricted to the i'th key class, denoted by $C_i^N(\boldsymbol{\alpha}, \boldsymbol{\beta})$, is an M-variate* *normal distribution*

$$C_i^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{N}_M(\boldsymbol{\mu}_i, 1/N \cdot \mathbf{I}).$$

The mean vector of the i'th key class is given by $\boldsymbol{\mu}_i[j] = s_{i,j}|C_K(T_i)|$*, where* $s_{i,j} \in \{-1, 1\}$ *describes the sign combination of the i'th key class,* $j = 1, \dots, M$*.*

Based on this model, a Bayesian classifier is constructed. We refer to Section 5 for the details. While the approach presented by Biryukov et al. seems promising, it has yet to result in an improved attack on DES. To the best of our knowledge, no other variants of linear cryptanalysis which uses multiple approximations have been able to outperform Matsui's original attack. Moreover, while updated versions of Assumption 1 and Assumption 2 have been applied to other ciphers, no such work exists for DES. In the following, we address these concerns. We consider the right-key distribution in Section 3, and the wrong-key distribution in Section 4. Using the results obtained in these sections, we develop an improved linear attack on DES in Sections 5 and 6.

3 Right-Key Correlation for DES: Key Inequivalence

In this section, we consider the correlation distribution of DES approximations over the key space. In Section 3.1, we consider current models for this distribution, as well as the undersampled distribution. In Section 3.2, we enumerate a large number of trails for DES, and show that, contrary to Assumption 1, the absolute value of the correlation does vary significantly as the key changes. In fact, the correlation distribution has a complicated structure. In Section 3.3, we develop a new model for this correlation based on Gaussian mixtures, which is able to accurately describe this structure. Moreover, we extend the model to describe the full undersampled correlation distribution over keys for multiple approximations.

3.1 The Correlation Distribution of a Single Approximation

As mentioned, most linear cryptanalysis of DES assumes that each linear approximation has one dominant trail, determining the magnitude of the absolute correlation. This idea is effectively expressed by Assumption 1. Consider, for example, one of the approximations used by Matsui, γ_1. This approximation has a primary trail T_A over 14 rounds of DES with correlation contribution $C_K(T_A) = \pm 2^{-19.75}$. In [23], Nyberg first introduced the concept of a linear hull, i.e. the collection of all trails of a linear approximation, and showed that Assumption 1 is not true in general. For γ_1, the trail with second largest correlation contribution, T', has contribution $C_K(T') = \pm 2^{-25.86}$. While the contribution from this trail is not large enough to change the sign of the linear correlation $C_K(\gamma_1)$, or increase/decrease the magnitude of the correlation much, it does not match the model given in Assumption 1. When including the second trail, the correlation distribution does not take on only two distinct values, but four.

Signal/noise decomposition.

In order to refine Assumption 1, Bogdanov and Tischhauser considered a *signal/noise decomposition* of the hull in [6]. Consider a situation in which d dominant trails of an approximation (α, β) are known. We call this collection of trails the *signal*, and define the *signal correlation* as the sum of their correlation contributions

$$C'_K(\alpha, \beta) = \sum_{i=1}^{d} (-1)^{s_{T_i} \oplus \langle \bar{T}_i, \bar{K} \rangle} |C_K(T_i)|.$$

The remaining part of the hull is unknown, and is modelled as *noise*, with the distribution $\mathcal{N}(0, 2^{-n})$. Then, the refined right-key equivalence assumption of [6] states that the correlation of (α, β) is given by the sum of the signal correlation and the noise:

$$C_K(\alpha, \beta) = C'_K(\alpha, \beta) + \mathcal{N}(0, 2^{-n}).$$

Since the approximations we will typically consider in the context of DES have quite high correlation, the addition of the noise term will not make a significant difference. However, we include it for completeness.

Undersampling.

The cryptanalyst is most often not interested in having to obtain the full codebook to exactly measure the linear correlation $C_K(\alpha, \beta)$. Therefore, the undersampled distribution is of great interest. Let

$$C_K^N(\alpha, \beta) = \frac{2}{N} \#\{x_i, i = 1, \ldots, N | \langle \alpha, x_i \rangle \oplus \langle \beta, E_K(x_i) \rangle = 0\} - 1$$

be the empirical value of $C_K(\alpha, \beta)$ measured using N text pairs. Here, we assume that x_i is drawn uniformly at random with replacement from $\mathbb{F}_2^n$. Matsui first considered the distribution of $C_K^N(\alpha, \beta)$ over the key space under Assumption 1. In this case, Matsui used the Gaussian distribution $C_K^N(\alpha, \beta) \sim \mathcal{N}(C_K(\alpha, \beta), 1/N)$. While no proof is given in [20], one can show this result via a Gaussian approximation to the binomial distribution, assuming that $|C_K(\alpha, \beta)|$ is small.

3.2 Exploring the Signal Distribution of DES

On the basis of the signal/noise model, we now turn our attention to the signal distribution of DES approximations. By computing the signal correlation C'_K for a large number of trails, we are able to get a good idea of the actual distribution of the correlation C_K. We first describe how the signal trails were enumerated.

Our trail enumeration algorithm.

We implemented a bounded breadth-first search in order to enumerate trails of DES approximations over 14 rounds. The algorithm consists of two search phases and a matching phase. Consider an approximation (α, β). The first search phase searches for trails in the forward direction, from round one to round seven. The search starts with α as an input mask to the first round, and then finds t and v such that the single round trails (α, t, v) has non-zero correlation. This process is then repeated for each trail with v as input mask to the second round, etc. The second search phase is similar, but searches backwards from β.

The searches are bounded in two ways. First, we only consider trails that activate at most three S-Boxes in each round. Second, we limit the number of trails which are kept in each round. This is done in such a way that only the trails with largest absolute correlation contribution are kept. This ensures a locally optimal choice, although no such guarantee can be made globally. The number of trails kept is determined by the *branching factor* B, such that in the i'th round of the search, $i \cdot B$ trails are kept.

After the two search phases, each trail found in the forward direction is matched to any trail in the backwards direction which shares the same mask in the middle. In this way, we obtain a number of trails of (α, β) over 14 rounds. Globally optimal trails will have a good chance of being enumerated if the branching factor B is chosen sufficiently large. In the following, we set $B = 1$ million, which means that we can find at most 7 million trails in each search direction. Note that the number of trails eventually discovered by the algorithm highly depends on the number of rounds and the approximation under consideration. We performed the enumeration for the eight approximations given in Table 2 using 20 Intel Xeon Processor E5-2680 cores. The enumeration took about 8 CPU hours.

Computing the Signal Distribution.

Using the algorithm described above, we enumerated 1126 trails of the approximation γ_1 over 14 rounds, and calculated the signal correlation

$$C'_K(\gamma_1) = \sum_{i=1}^{1126} (-1)^{s_{T_i} \oplus \langle \bar{T}_i, \bar{K} \rangle} |C_K(T_i)|,$$

for 1 million randomly drawn keys. The trails we found have an absolute correlation contribution between $2^{-43.61}$ and $2^{-19.75}$, and include the dominant trail used by Matsui in [21]. The resulting distribution can be seen in Figure 3.

The left part of the figure shows the full distribution over the key space. At this scale, the distribution resembles the one described in Section 2; there are two very prominent modes symmetric around zero, with peaks around $\pm 2^{-19.75}$, corresponding to the correlation contribution of the dominant trail. However, the right part of the plot, showing the positive half of the distribution, largely contradicts Assumption 1

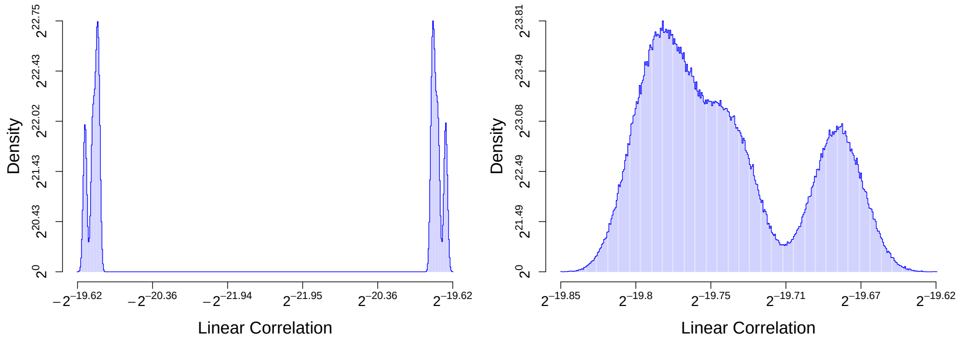


Figure 3: The signal distribution of linear correlation for the approximation γ_1 over 14 rounds of DES. The signal correlation was calculated using 1126 trails and 1 million randomly drawn keys. The trails had an absolute correlation contribution between $2^{-43.61}$ and $2^{-19.75}$. The left plot shows the two main modes, symmetric around zero. The right plot shows only the positive half of the distribution.

of key equivalence. While the mean of the distribution is $2^{-19.75}$, it also has a non-negligible standard deviation of $2^{-24.71}$. Moreover, the distribution is not Gaussian. The correlations cluster around three values, namely $2^{-19.79}$, $2^{-19.75}$, and $2^{-19.68}$. Interestingly, the probability density is larger around the cluster with the lowest correlation value.

Under the signal/noise model, adding the noise distribution $\mathcal{N}(0, 2^{-n})$ gives us a good estimate of the actual distribution of the correlation $C_K(\gamma_1)$. However, due to the large variance of the signal distribution, the effect of the noise term is negligible in this case. Thus, the distribution in Figure 3 should be quite close to the actual distribution. This poses a fundamental problem, as none of the analysis of linear cryptanalysis applied to DES accounts for this type of distribution. Indeed, it is not clear how the distribution of the undersampled correlation, C_K^N, looks, which is essential to know when determining the complexity of linear attacks.

3.3 A New Mixture Model for Single and Multiple Approximations

To relieve the problems discussed in Section 3.2, we now propose a model for the correlation distribution based on *Gaussian mixtures*. Consider a distribution in which each sample is drawn from one of ℓ Gaussian distributions. Each Gaussian is called a *component*. The probability of the sample being drawn from the i'th component is λ_i, usually called the *weights*, with $\sum \lambda_i = 1$. The probability density function

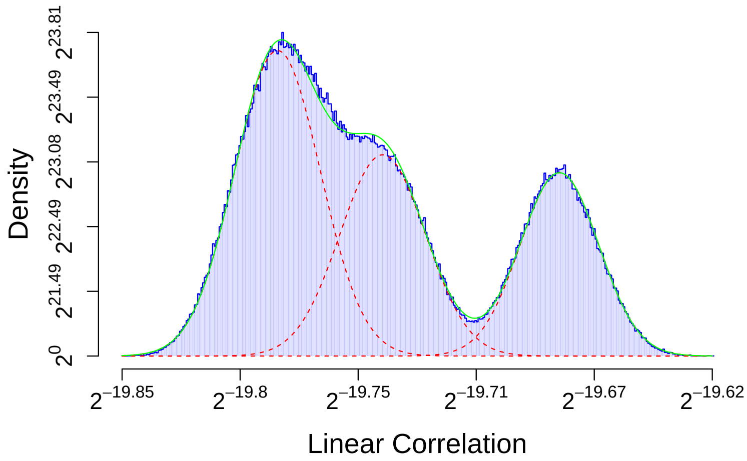


Figure 4: A Gaussian mixture fitted to the correlation distribution of the linear approximation γ_1 over 14 rounds of DES. The individual components are shown in red, the mixture density is shown in green, and the measured distribution is shown in blue. Under this model, only 30% of the distribution is attributed to the Gaussian component associated with the dominant trail.

(PDF) of such a distribution is given by

$$f(x) = \sum_{i=1}^{\ell} \lambda_i \phi(x; \mu_i, \sigma_i^2),$$

where $\phi(x; \mu_i, \sigma_i^2)$ is the PDF of the i'th Gaussian distribution, having mean μ_i and variance σ_i^2 [19]. We will denote the distribution itself by $\mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell)$. We then propose the following model.

Model 2 (Right-Key Inequivalence for One Approximation)**.** *Consider a linear approximation (α, β) of r rounds of DES. The distribution of the linear correlation $C_K(\alpha, \beta)$ over the key space is approximately given by a Gaussian mixture for some weights λ_i and components $\mathcal{N}(\mu_i, \sigma_i^2)$, $i = 1, \ldots, \ell$. That is,*

$$C_K(\alpha, \beta) \sim \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell).$$

We note that the signal/noise decomposition easily applies to this model. If we determine that the signal correlation follows a Gaussian mixture, i.e. $C'_K(\alpha, \beta) \sim$

$\mathcal{M}(\lambda_i', \mu_i', \sigma_i^{2\prime}, \ell')$ for some appropriate parameters, then we can approximate the actual correlation distribution by adding the noise distribution:

$$C_K(\alpha, \beta) \sim \mathcal{M}(\lambda_i', \mu_i', \sigma_i^{2\prime}, \ell') + \mathcal{N}(0, 2^{-n}).$$

We apply Model 2 to the distribution obtained in Section 3.2. The result of fitting a Gaussian mixture model with three components to the positive part of the signal distribution is shown in Figure 4. We first note that the mixture model fits the measured signal distribution quite well. The parameters are

$$\begin{aligned} \lambda_1 &= 0.45, & \mu_1 &= 2^{-19.79}, & \sigma_1^2 &= 2^{-52.40}, \\ \lambda_2 &= 0.30, & \mu_2 &= 2^{-19.75}, & \sigma_2^2 &= 2^{-52.37}, \\ \lambda_3 &= 0.25, & \mu_3 &= 2^{-19.68}, & \sigma_3^2 &= 2^{-52.68}. \end{aligned}$$

The second mixture component has mean equal to the correlation contribution of the dominant trail, but this component only contributes to 30% of the full distribution. In fact, the main part of the contribution, 45%, can be attributed to the first component, which has a slightly lower mean. This demonstrates that considering only the contribution of the dominant trail can be misleading, even when the remaining trails have a far lower correlation contribution. In general, one should consider as large a part of the hull as possible. Nevertheless, for attacks with relatively low data complexity, the actual distribution can easily be hidden, as we shall see next.

The undersampled mixture.

In Section 3.2, we recalled that under the assumption of a dominant trail, the distribution of the undersampled correlation C_K^N is given by the Gaussian $\mathcal{N}(C_K, 1/N)$. We state the following equivalent result in the setting of Model 2 and give an outline of the proof.

Theorem 1 (Undersampled distribution)**.** *Assuming Model 2, the undersampled correlation distribution of an approximation* (α, β) *obtained using* N *random text pairs is given by*

$$C_K^N(\alpha, \beta) \sim \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, \ell) + \mathcal{N}(0, 1/N).$$

Proof. For any fixed key k, C_k^N is distributed as $\mathrm{Bin}(N, C_k)$ over the random text sample, which can be approximated by $\mathcal{N}(C_k, 1/N)$ if C_k is small. That is, $C_K^N \mid K = k \sim \mathcal{N}(C_k, 1/N)$. The PDF of the compound distribution C_K^N, i.e. without the conditioning on K, is given by

$$p_{C_K^N}(y) = \int \phi(y; x, 1/N) \cdot \sum_{i=1}^{\ell} \lambda_i \phi(x; \mu_i, \sigma_i^2) dx,$$

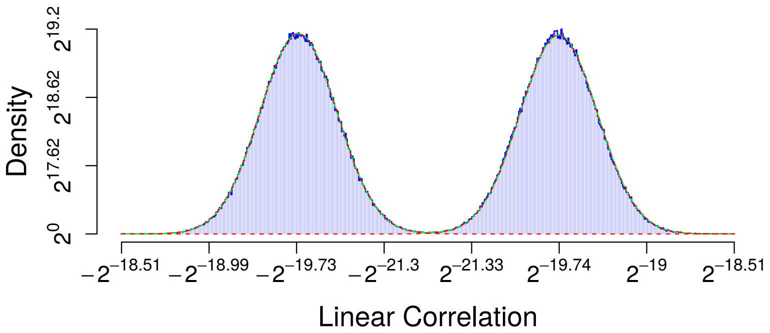


Figure 5: The distribution of the undersampled linear correlation of γ_1, $C'_K + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N)$, over 14 rounds of DES, with $N = 2^{43}$. C'_K was measured using 1126 trails over 1 million randomly drawn keys. A Gaussian mixture with two components have been fitted to the distribution. The components are shown in red, while the full distribution is shown in green.

which can be shown to be equal to

$$p_{C_K^N}(y) = \sum_{i=1}^{\ell} \lambda_i \phi(y; \mu_i, \sigma_i^2 + 1/N).$$

This is a Gaussian mixture where each component can be written as $\mathcal{N}(\mu_i, \sigma_i^2) + \mathcal{N}(0, 1/N)$. But since we add the second distribution with probability one, the same distribution can be obtained by first drawing from the original mixture, and then adding the distribution $\mathcal{N}(0, 1/N)$, finishing the proof. □

If the number of texts N is relatively large, the model can be somewhat simplified. If we wanted to apply Model 2 and Theorem 1 directly to the case of γ_1, we would model the measured correlation as

$$C_K^N(\gamma_1) = \mathcal{M}(\lambda_i, \mu_i, \sigma_i^2, 6) + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N), \tag{4}$$

using six components for the Gaussian mixture. However, the details of the mixture are easily lost at high levels of undersampling, as can be seen in Figure 5. Here, we have shown the distribution

$$C'_K(\gamma_1) + \mathcal{N}(0, 2^{-n}) + \mathcal{N}(0, 1/N),$$

where $N = 2^{43}$. The resulting distribution can be described as a Gaussian mixture with two components, instead of six. Each component has variance roughly equal to $1/N$, and the means are $\pm 2^{-19.75}$, i.e. the correlation contribution of the dominant trail. This agrees with the models used by e.g. Matsui and Biryukov, et al., but we stress that this is only true when N is relatively small compared to the linear correlation. In particular, for ciphers with strong dominant trails, $1/N$ needs to be larger than the variance of the positive/negative part of the distributions. For values of N close to the full codebook, this is not true (unless the approximation is extremely weak), and the distribution of C_K cannot be ignored. However, this simplification will help greatly when we consider the joint distribution of multiple approximations in the next subsection.

The Gaussian mixture of multiple approximations.

Model 2 and the results of Section 3.3 can be generalised to consider the case of multiple linear approximations. Let $C_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$ denote the vector of correlations of M linear approximations, $(C_K(\alpha_1, \beta_1), \dots, C_K(\alpha_M, \beta_M))^\top$. In the following, we will restrict ourselves to the case where the signal distributions, $C'_K(\alpha_i, \beta_i)$, each have two distinct modes: one positive and one negative. This allows us to split the joint signal distribution, $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$, into at most 2^M components determined by the signs of $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$. In the case of relatively low values of N, we propose the following model.

Model 3 (Right-Key Mixture for Multiple Approximations)**.** *Consider a set of linear approximations $(\alpha_1, \beta_1), \dots, (\alpha_M, \beta_M)$ of r rounds of DES. The undersampled distribution of the linear correlation vector over the key space, $C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta})$, is approximately given by an M-variate Gaussian mixture, namely*

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{M}_M(1/\ell, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + 1/N \cdot \mathbf{I}, \ell),$$

where $\ell \le 2^M$. Moreover, the parameters of the mixture components are given by

$$\boldsymbol{\mu}_i = \mathcal{E}(C_K(\boldsymbol{\alpha}, \boldsymbol{\beta}) | s_{i,j} \cdot C_K(\alpha_i, \beta_i) > 0, j = 1, \dots, M),$$
$$\boldsymbol{\Sigma}_i = \mathrm{Cov}(C_K(\boldsymbol{\alpha}, \boldsymbol{\beta}) | s_{i,j} \cdot C_K(\alpha_i, \beta_i) > 0, j = 1, \dots, M),$$

where $s_{i,j} \in \{-1, 1\}$ describes the sign combination of the i'th component.

As for the case of a single approximation, the signal/noise decomposition applies to this model, resulting in an undersampled distribution of the form

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{M}_M(1/\ell, \boldsymbol{\mu}'_i, \boldsymbol{\Sigma}'_i + (2^{-n} + 1/N)\mathbf{I}, \ell).$$

The signal parameters, $\boldsymbol{\mu}'_i$ and $\boldsymbol{\Sigma}'_i$, can be estimated by enumerating an appropriate number of trails and then calculating $C'_K(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for a large number of keys.

This model bears some resemblance to the one given by Biryukov et al. in [2]. While both models use the signs of the correlation vector to split the distribution into several Gaussians, our model captures the entire key space in one distribution, whereas the model in [2] partitions the key space into at most 2^M parts which are considered separately. Additionally, we do not make any assumption about the independence of the linear approximations. As such, $\boldsymbol{\Sigma}_i$ need not be diagonal matrices, and not all 2^M sign combinations need to be present. While the possibility of $\ell < 2^M$ is briefly mentioned in [2], all experiments were done such that $\ell = 2^M$. As we shall see in Section 5, the case of $\ell < 2^M$ allows for stronger attacks. Moreover, an improved attack on full DES was not presented in [2] . We apply our model to obtain a key-recovery attack on full DES in Section 6. First, however, we turn our attention to the wrong-key distribution.

4 Wrong-Key Correlation for DES: Non-Random Behaviour

In this section, we consider the correlation distribution of DES approximations in the case of a wrong key guess in Algorithm 2. This distribution is essential, as the effectiveness of the algorithm is determined by how different the right-key and wrong-key distributions are. In Section 4.1, we consider the current models for the wrong-key distribution. In Section 4.2, we develop a new model for the wrong-key distribution of DES, and show that the distribution obtained under this model deviates significantly from that considered in Section 4.1. Nevertheless, as for the right-key in Section 3, we show that the deviation has little impact when the number of texts used in the attack is relatively small.

4.1 The Current Ideal Wrong-Key Distribution

The assumption of wrong-key randomisation, Assumption 2, used by Matsui in [21] and by Biryukov et al. in [2], predicts that a wrong guess of the outer round keys in Algorithm 2 should result in an approximation with correlation zero. This is motivated by the idea that if we encrypt/decrypt using the wrong key, we are doing something equivalent to encrypting two extra rounds. This should result in a linear correlation much closer to zero, as we are essentially considering the correlation over $r + 4$ rounds instead of r rounds. However, as shown by Daemen and Rijmen in [11], even a linear approximation of an ideal permutation will approximately have the correlation distribution

$$C_K(\alpha, \beta) \sim \mathcal{N}(0, 2^{-n}),$$

where n is the blocksize. Since we intuitively cannot do "worse" than an ideal cipher, the correlation of a wrong guess should follow this distribution. This consideration led Bogdanov and Tischhauser to present an updated wrong-key randomisation hypothesis in [6], in which the wrong key correlation follows this ideal Gaussian distribution. However, if we consider the case of DES where, even over 14 rounds, strong linear approximations exist, the wrong-key correlation might not be close to the ideal distribution. We consider this problem next.

4.2 A New Non-Random Wrong-Key Distribution

Consider the scenario in which an attacker obtains a plaintext-ciphertext pair computed over r rounds of a cipher, and attempts to encrypt the plaintext one round, and decrypt the ciphertext one round, in order to calculate the correlation of an approximation over $r-2$ rounds. If the attacker uses the wrong round keys for the encryption/decryption, she essentially obtains a plaintext/ciphertext pair of some related cipher with $r+2$ rounds. Motivated by this, we propose the following wrong-key model for linear cryptanalysis on DES.

Model 4 (Non-Random Wrong-Key Distribution)**.** *Consider an Algorithm 2 style attack on r rounds of DES using a linear approximation (α, β) over r − 2 rounds. Let R_K be the keyed round function of DES, and let $E_K^\star$ denote the r-round encryption function. For a wrong guess of the outer round keys, the correlation will be distributed as for the cipher*

$$E'_K(x) = R_{K_a}^{-1}(E_K^\star(R_{K_b}^{-1}(x))), \tag{5}$$

where K_a and K_b are chosen uniformly at random.

For DES, where encryption and decryption are similar, this can reasonably be simplified to $E'_K(x) = E_K^{r+2}$, where the outer round keys are randomly chosen.

In light of this, we considered the approximation γ_1 over 18 rounds of DES, with randomly chosen outer round keys. Using the algorithm described in Section 3.2, with $B = 1$ million, we enumerated 954 trails of this approximation. Using 20 Intel Xeon Processor E5-2680 cores, the enumeration took about 15 CPU hours. We then calculated the resulting signal correlation for 1 million keys. The trails had an absolute correlation contribution between $2^{-45.84}$ and $2^{-28.75}$. The distribution is shown in Figure 6. We note that the result is similar for the other approximations given in Table 2.

As was the case for the right-key distribution, this wrong-key distribution appears to be approximately a Gaussian mixture. More importantly, while the distribution is symmetric around zero, the variance is much larger than that of an ideal permutation: $2^{-56.08}$ compared to 2^{-64}. This shows that, while the added four rounds make the correlation weaker, the assumption of a resulting ideal distribution is optimistic. For attacks that use a data complexity close to the full codebook, this assumption could result in a overestimate of success probability or an underestimate of attack complexity. Moreover, if the cryptanalyst only appends/prepends one round to the approximation, this effect could be significant.

The undersampled distribution.

While the distribution in Figure 6 is far from ideal, the actual distribution of the correlation matters little if the level of undersampling is significant. If we apply signal/noise decomposition and Theorem 1 to our estimate of the wrong-key distribution, with the number of texts $N = 2^{43}$, we obtain the result shown in Figure 7.

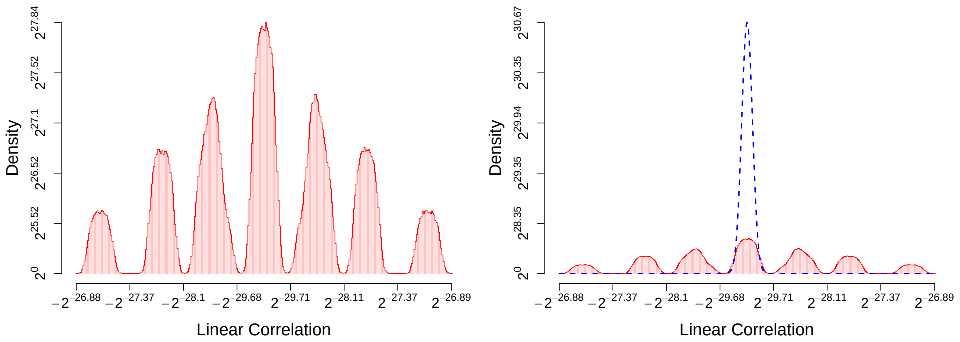


Figure 6: The distribution of linear correlation for the approximation γ_1 over 18 rounds of DES with randomly chosen outer round keys. The correlation was calculated using 954 trails and 1 million randomly drawn keys. The distribution is close to zero, but the variance is $2^{-56.08}$. To the right, the distribution is compared to that of an ideal permutation, i.e the Gaussian $\mathcal{N}(0, 2^{-64})$.

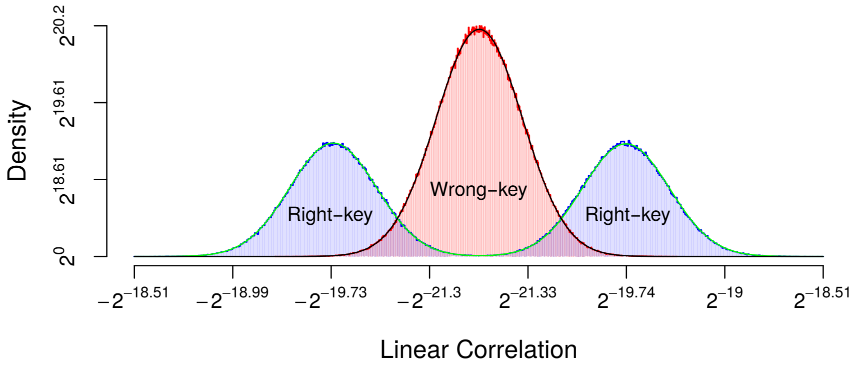


Figure 7: Undersampled right-key (blue) and wrong-key (red) distributions for the approximation γ_1 with $N = 2^{43}$. The signal distributions were measured using 1 million randomly drawn keys. A Gaussian mixture has been fitted to the right-key distribution (green), while a single Gaussian distribution was fitted to the wrong-key distribution (black).

We see here that it is sufficient to use a single Gaussian distribution to approximate the undersampled wrong-key correlation distribution. If this distribution is similar for other approximations, it will be sufficient to model the joint wrong-key correlation distribution of M approximations as an M-variate Gaussian distribution. Thus, if $\boldsymbol{\Sigma}_W$ is the covariance matrix of the signal correlation of the M approximations over

E'_K, then the undersampled wrong-key distribution will approximately be given by

$$C_K^N(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I}),$$

if $1/N$ is sufficiently large.

Using Model 3 for the right-key and Model 4 for the wrong-key distribution, we develop a classifier that uses both these distributions in the following section.

5 Classifying Keys using Asymmetric Distributions

In Section 3, we developed a model for the linear correlation distribution of a correct key-guess in Algorithm 2, namely a multivariate Gaussian mixture model. In Section 4, we similarly developed a simple multivariate Gaussian model for the linear correlation distribution of a wrong key-guess. Using these two distributions, we now develop a classifier based on the likelihood-ratio, which can be used in Algorithm 2 to decide between potential right and wrong key guesses. We first present the classifier given in [2] in Section 5.1. We then introduce our new classifier in Section 5.2, and compare the performance of the two in Section 5.3.

In the following, we will consider the two sets of four linear approximations over 14 rounds of DES given in Table 2. While it is difficult to visualise the joint distribution of more than three approximations, Figure 8 shows the pairwise joint distributions of the approximations γ_1, γ_2, γ_3, and γ_4, as well as the marginal distributions, for $N = 2^{43}$. Note that the joint distributions of γ_1 and γ_3, as well as that of γ_2 and γ_4, only have two components. We will explore this phenomenon in Section 5.4, and show that such distributions can improve our classifier.

5.1 The Bayesian Classifier of Biryukov et al.

Consider an Algorithm 2 style attack using M linear approximations. Let $\mathcal{K}_R$ denote the space of correct guesses of the key-bits (k_f, k_b), and let $\mathcal{K}_W$ denote the space of wrong guesses. We have to classify each key-guess as either an incorrect guess or a potential correct guess, based on the measured linear correlation vector $\mathbf{x}$. Let $f_R(\mathbf{x}) = \Pr(\mathbf{x} \mid (k_f, k_b) \in \mathcal{K}_R)$ be the PDF of the right-key correlation distribution. We define the Bayesian classifier, BC, as the following decision rule

$$BC(\mathbf{x}) = \begin{cases} \text{If } B(\mathbf{x}) > \Gamma, \text{ decide that } (k_f, k_b) \in \mathcal{K}_R, \\ \text{otherwise, decide that } (k_f, k_b) \in \mathcal{K}_W, \end{cases}$$

where $B(\mathbf{x}) = f_R(\mathbf{x})$. Under Model 3, $B(\mathbf{x})$ is given as the Gaussian mixture

$$B(\mathbf{x}) = \sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I}).$$

This exact classifier is not described in [2], but it is essentially identical to the one developed there. The difference is that in [2], each component of f_R is considered

Linear approximation	Dominant key trail	$\lvert C_K(T_\cdot)\rvert$	$s_{T_\cdot}$
$\gamma_1 = ([7, 18, 24], [7, 18, 24, 29, 47])$	$\bar{T}_A$	$2^{-19.75}$	1
$\gamma_2 = ([7, 18, 24], [7, 18, 24, 29, 44, 48])$	$\bar{T}_B$	$2^{-20.48}$	1
$\gamma_3 = ([7, 18, 24, 29], [7, 18, 24, 47])$	$\bar{T}_A$	$2^{-20.75}$	0
$\gamma_4 = ([7, 18, 24, 29], [7, 18, 24, 44, 48])$	$\bar{T}_B$	$2^{-20.07}$	1
$\delta_1 = ([15, 39, 50, 56], [39, 50, 56, 61])$	$\bar{T}_C$	$2^{-20.75}$	0
$\delta_2 = ([12, 16, 39, 50, 56], [39, 50, 56, 61])$	$\bar{T}_D$	$2^{-20.07}$	1
$\delta_3 = ([15, 39, 50, 56, 61], [39, 50, 56])$	$\bar{T}_C$	$2^{-19.75}$	1
$\delta_4 = ([12, 16, 39, 50, 56, 61], [39, 50, 56])$	$\bar{T}_D$	$2^{-20.48}$	1

Key trail	Non-zero key mask bits	Key trail	Non-zero key mask bits
$\bar{T}_A$	$\{t_1^{22}, t_2^{44}, t_3^{22}, t_5^{22}, t_6^{44}, t_7^{22}, t_9^{22}, t_{10}^{44}, t_{11}^{22}, t_{13}^{22}\}$	$\bar{T}_B$	$\bar{T}_A \backslash t_{13}^{22} \cup \{t_{13}^{19}, t_{13}^{23}\}$
$\bar{T}_C$	$\{t_0^{22}, t_2^{22}, t_3^{44}, t_4^{22}, t_6^{22}, t_7^{44}, t_8^{22}, t_{10}^{22}, t_{11}^{44}, t_{12}^{22}\}$	$\bar{T}_D$	$\bar{T}_C \backslash t_0^{22} \cup \{t_0^{19}, t_0^{23}\}$

Table 2: The top table specifies two sets of four linear approximations over 14 rounds of DES, and gives the correlation contribution of their dominant trail, as well as the sign bit of that trail. The bottom table specifies the set of non-zero bits of the associated dominant key trails, where t_i^j is the j'th bit of t_i.

separately, and so ℓ scores are produced for each key guess. The classifier BC should be functionally equivalent to this approach, but this representation allows for easy comparison to the likelihood-ratio classifier we propose next.

5.2 Our Likelihood Classifier

We now propose a new classifier based in the likelihood-ratio. As opposed to the Bayesian classifier, the likelihood classifier directly takes the wrong-key distribution into account. To this end, let $f_W(\mathbf{x}) = \Pr(\mathbf{x} \mid (k_f, k_b) \in \mathcal{K}_R)$ be the PDF of the wrong-key correlation distribution. Then the *likelihood-ratio* is defined as $\Lambda(\mathbf{x}) = f_R(\mathbf{x})/f_W(\mathbf{x})$. For the right-key and wrong-key distributions described in Sections 3 and 4, this is equal to

$$\Lambda(\mathbf{x}) = \frac{\sum_{i=1}^{\ell} \lambda_i \phi_M(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_W + (2^{-n} + 1/N)\mathbf{I})},$$

where $\mathbf{x}$ is an observed value of correlations for M approximations. A large value of $\Lambda(\mathbf{x})$ will then indicate a likely correct key guess, while a low value will indicate a wrong key guess. Thus, we define the *likelihood classifier* LC as the following decision rule

$$LC(\mathbf{x}) = \begin{cases} \text{If } \Lambda(\mathbf{x}) > \Gamma, \text{ decide that } (k_f, k_b) \in \mathcal{K}_R, \\ \text{ otherwise, decide that } (k_f, k_b) \in \mathcal{K}_W. \end{cases}$$

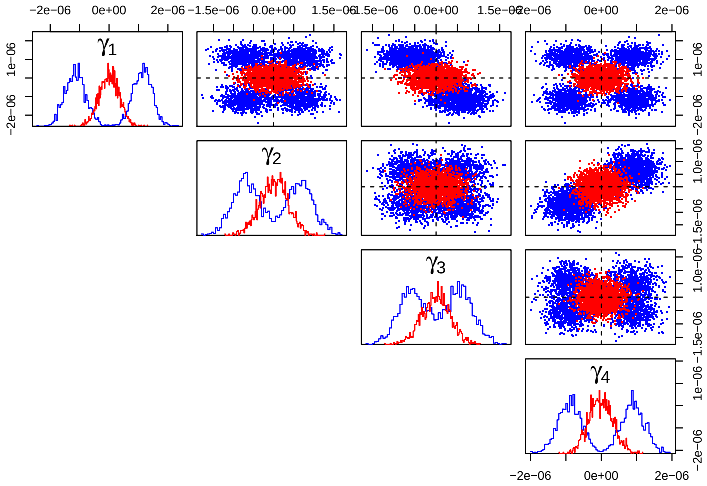


Figure 8: Histograms and pairwise distributions of the undersampled correlations of approximations $\gamma_1, \ldots, \gamma_4$ given in Table 2. The right-key distributions are shown in blue, the wrong-key distributions are shown in red. The number of texts is $N = 2^{43}$. Note that since γ_1 and γ_3 have the same dominant key trail, their joint distribution only has two components. Likewise for γ_2 and γ_4.

In light of this definition, two important concepts are the success probability and advantage of the classifier. Formally, we define the success probability and advantage, respectively, as

$$P_S = 1 - \Pr(\Lambda(\mathbf{x}) < \Gamma \mid (k_f, k_b) \in \mathcal{K}_R), \tag{6}$$

$$a = -\log_2(\Pr(\Lambda(\mathbf{x}) \geq \Gamma \mid (k_f, k_b) \in \mathcal{K}_W)), \tag{7}$$

in accordance with the usual definition [24]. We usually choose Γ such that we achieve a certain success probability. Under our proposed model, the involved probabilities cannot be explicitly stated. Thus, we must rely on simulations to calculate these values. Since simulating values from a Gaussian distribution is easy, this is not a problem. Using this approach, we now compare the performance of the likelihood classifier and the Bayesian classifier.

5.3 Decision Boundaries

The likelihood classifier *LC* divides the M-dimensional cube $[-1, 1]^M$ into two regions separated by the *decision boundary*, namely where $\Lambda(\mathbf{x}) = \Gamma$. On one side of the decision boundary, observations are classified as belonging to the right-key distribution, while observations from the other side are classified as belonging to the wrong-key distribution. By visualising this decision boundary, we can get a better understanding of the classifier.

In the following, we consider the eight approximations given in Table 2, over 14 rounds of DES. We enumerated between 1100 and 1400 trails for each approximation and calculated the signal correlations for 1 million random keys, in order to estimate $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$. The same was done over E'_K, where between 950 and 1100 trails were enumerated, in order to estimate $\boldsymbol{\Sigma}_W$. For each data point, we added noise drawn from $\mathcal{N}_M(\mathbf{0}, (2^{-n} + 1/N)\mathbf{I})$, according to the signal/noise decomposition and Theorem 1. This allows us to simulate $\Lambda(\mathbf{x})$ and $B(\mathbf{x})$ for varying values of N and calculate the resulting decision boundary and advantage.

Consider the pair of approximations γ_1 and δ_1 and let $N = 2^{43}$. We simulate $\Lambda(\mathbf{x})$ and $B(\mathbf{x})$ for each data point as described above, and then fix a threshold value for each classifier such that $P_S = 0.90$, cf. Equation 6. The resulting decision boundaries, as well as the related probability distributions, are shown in Figure 9. In this case, the likelihood classifier obtains an advantage of 5.5 bits, while the Bayesian classifier only has an advantage of 3.1 bits. By considering the decision boundary, it is clear why this is the case. Since the Bayesian classifier only uses information about the right-key distribution, it simply creates a decision boundary around each component of the mixture which is large enough to obtain the desired success probability. In view of the information that is available to the classifier, this makes sense, since observations close to the mean of component have a larger chance of being a correct key guess. Because of this, the parts of the right-key distribution which is farthest away from the wrong-key distribution is also discarded as unlikely candidates. This in turn requires the decision boundary to be wider than actually needed, and the advantage is therefore quite low due to an increased number of false positives.

The likelihood classifier on the other hand does use information about the wrong-key distribution. The decision boundary is created such that there is a good boundary between each component and the wrong-key distribution. Any observation that is sufficiently far away from the wrong-key distribution is deemed a likely correct key guess, no matter how extreme the observation is in the right-key distribution. Thus, extreme points in the right-key distribution are not "wasted", allowing for a tight decision boundary around the wrong-key distribution, yielding a larger advantage.

For the approximations used here, all sign combinations of the correlation vector are possible. In terms of the mixture model, the number of components is $\ell = 2^M$. We now turn our attention to the case where $\ell < 2^M$.

5.4 Observations on the Asymmetric Distribution

As shown in Section 3.2, the sign of the signal correlation $C'_K(\gamma_1)$ for a given key is determined by the parity $\langle \bar{T}_A, \bar{K} \rangle$, where $\bar{T}_A$ is the dominant key trail. Consider the two approximations γ_1 and γ_3 given in Table 2. Both approximations have the same dominant key trail, and since their sign bits s_T are different, the sign of their

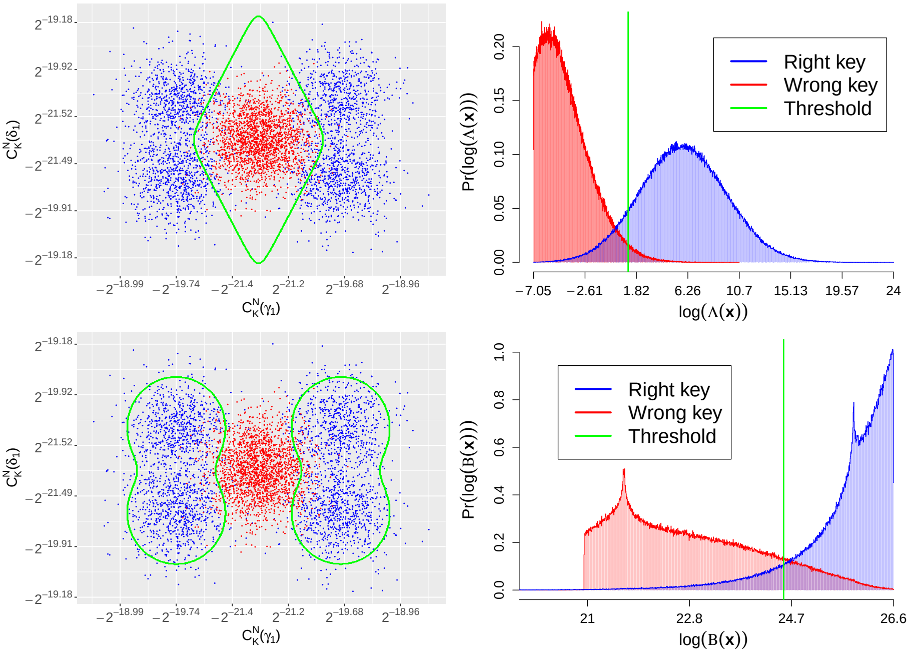


Figure 9: Left: The joint distribution of $C_K^N(\gamma_1)$ and $C_K^N(\delta_1)$, with $N = 2^{43}$, are shown for both a right key guess (blue) and a wrong key guess (red). The decision boundaries for a success probability of 90% are drawn for the likelihood-ratio classifier (top) and the Bayesian classifier (bottom). Right: The corresponding probability distributions of $\Lambda(\mathbf{x})$ (top) and $B(\mathbf{x})$ (bottom) as well as the threshold value. The likelihood ratio classifier obtains an advantage of 5.5 bits, while the Bayesian classifier obtains an advantage of 3.1 bits.

correlation will therefore always be opposite. In the terminology of Section 3.3, the number of components ℓ of the Gaussian mixture is strictly less than 2^M. We will call such a distribution *asymmetric*. On the other hand, the two approximations γ_1 and δ_1 have different dominant key-trails, and therefore all four sign combinations of their correlations are possible. In this case, $\ell = 2^M$, and we call such a distribution *symmetric*.

For γ_1 and δ_1, the decision boundary for the likelihood classifier was shown in Figure 9. For γ_1 and γ_3, the decision boundary is shown in Figure 10. Here, the "missing" components in the first and third quadrant are clearly shown, while the wrong-key distribution is still symmetric around zero. We note that, all else being equal, the classifier on the asymmetric distribution achieves an increased advantage of 0.7 bits. Moreover, the comparison here is fair, since the strength of δ_1 is the same as that of γ_3. The reason for this increase is apparent when we compare the two

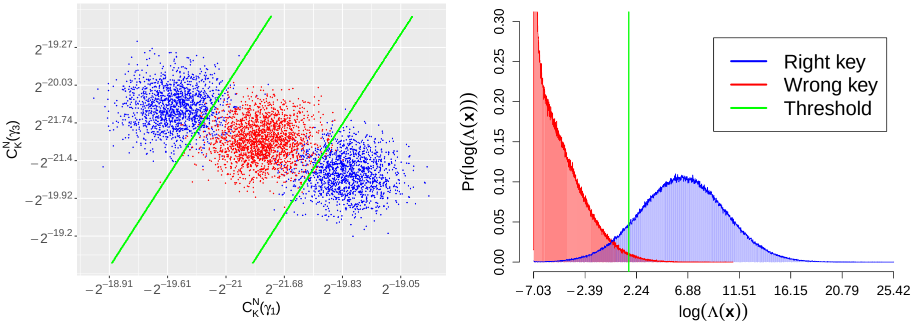


Figure 10: Left: The joint distribution of $C_K^N(\gamma_1)$ and $C_K^N(\gamma_3)$, with $N = 2^{43}$, are shown for a right key guess (blue) and a wrong key guess(red). The decision boundaries for a success probability of 90% are drawn for the likelihood-ratio classifier. Right: The probability distributions of $\Lambda(\mathbf{x})$ as well as the threshold value. The classifier obtains an advantage of 6.2 bits.

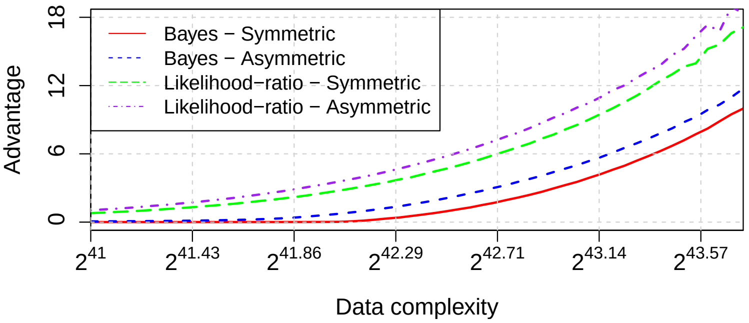


Figure 11: A comparison of the advantage obtained by using the Bayesian classifier and the likelihood ratio classifier on both symmetric and asymmetric correlation distributions. The symmetric distribution uses the set of approximations $\{\gamma_1, \gamma_2, \delta_1, \delta_2\}$ while the asymmetric distribution uses the set $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$.

decision boundaries. For the asymmetric distribution, the decision boundary is such that even extreme points in the wrong-key distribution towards the first and third quadrant are easily classified as wrong key guesses. This decreases the number of false positives, increasing the advantage.

This improvement in the classifier for asymmetric distributions generally extends to higher dimensions, where the effect can be even more pronounced. Indeed, for larger M, ℓ can be much smaller than 2^M. In the example above, we had $\ell = 2$ while $2^M = 4$. Consider now the set of approximations $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$. A shown in Table 2, these approximations only have two distinct dominant key trails, implying that the set has an asymmetric distribution with $\ell = 4 < 2^M = 16$. Figure 11 compares the advantage of this set of approximations to the set $\{\gamma_1, \gamma_2, \delta_1, \delta_2\}$, which has a symmetric distribution, i.e. $\ell = 2^M = 16$. In general, we observe that the classifiers are stronger for the asymmetric distribution, with an increase in advantage of 1.4 bits for $N = 2^{43}$. Additionally, the better performance of the likelihood classifier is quite clear, consistently obtaining a larger advantage over the Bayesian classifier. For $N = 2^{43}$, the likelihood classifier has an advantage 4.9 bits higher than the Bayesian classifier on both the symmetric and asymmetric distribution. Due to these observations, we propose the term *symmetry factor* for these types of distributions, defined as $\ell/2^M$. A distribution with symmetry factor one is a symmetric distribution, while a symmetry factor less than one indicates an asymmetric distribution. We conjecture that, all else being equal, a lower symmetry factor will result in a stronger classifier.

6 Improved Attack on DES

Using the results from the previous sections, we now mount a key-recovery attack on DES using eight linear approximations. We will use two sets of four linear approximations, $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ and $\{\delta_1, \delta_2, \delta_3, \delta_4\}$ over 14 rounds, as given in Table 2. The attack is mostly identical to Matsui's Algorithm 2. As such, we obtain N plaintext-ciphertext pairs over 16 rounds, guess the key-bits required to partially encrypt/decrypt the texts and compute the linear correlations, and then use the likelihood classifier to categorise each guess as a likely wrong or right key guess. For each guess, we further gain some parity bits of the key based on the signs of the correlations.

6.1 Attack Description

Table 3 shows the key- and text-bits relevant to the attack. For both sets of approximations, we need to know 29 bits of the plaintext/ciphertext, designated $t_{f,\cdot}$ / $t_{b,\cdot}$, and we will guess 24 bits of the first/last round key, designated $k_{f,\cdot}/k_{b,\cdot}$. Moreover, the signs of $C_K^N(\gamma_1)$, $C_K^N(\gamma_4)$, $C_K^N(\delta_3)$, and $C_K^N(\delta_2)$, will allow us to deduce the parity bits p_A, p_B, p_C, and p_D. Thus, the attacker will learn a total of 52 bits of the master key, and will have to guess the remaining 4 bits. In the following, we assume that the distribution parameters $\boldsymbol{\mu}_{i,\cdot}$, $\boldsymbol{\Sigma}_{i,\cdot}$, and $\boldsymbol{\Sigma}_{W,\cdot}$ have been determined before the attack, as described in Section 3.3. Moreover, we assume that $\lambda_i = 1/\ell$ for all i. The attack is then given as follows:

- **Distillation**
  1. Obtain N plaintext-ciphertext pairs.
  2. Create two vectors $\mathbf{t}_\gamma$ and $\mathbf{t}_\delta$ of size 2^{29} each. $\mathbf{t}_\gamma[i]$ (similarly $\mathbf{t}_\delta$) is equal to the number of text pairs such that the bits $(t_{f,\gamma}, t_{b,\gamma})$ are equal to i.

Forward key bits guessed				#bits
$k_{f,\gamma}$	$\{K_0^{18},\ldots,K_0^{23}\}$	$k_{f,\delta}$	$\{K_0^{24},\ldots,K_0^{35},$ $K_0^{42},\ldots,K_0^{47}\}$	6+18
Backward key bits guessed				#bits
$k_{b,\gamma}$	$\{K_{15}^{24},\ldots,K_{15}^{35},$ $K_{15}^{42},\ldots,K_{15}^{47}\}$	$k_{b,\delta}$	$\{K_{15}^{18},\ldots,K_{15}^{23}\}$	18+6
Plaintext bits stored				#bits
$t_{f,\gamma}$	$\{\mathcal{P}^{11},\ldots,\mathcal{P}^{16},\mathcal{P}^{39},\mathcal{P}^{50},\mathcal{P}^{56}\}$			9
$t_{f,\delta}$	$\{\mathcal{P}^{0},\mathcal{P}^{7},\mathcal{P}^{15},\ldots,\mathcal{P}^{24},\mathcal{P}^{27},\ldots,\mathcal{P}^{31},\mathcal{P}^{44},\mathcal{P}^{47},\mathcal{P}^{48}\}$			20
Ciphertext bits stored				#bits
$t_{b,\gamma}$	$\{\mathcal{C}^{0},\mathcal{C}^{7},\mathcal{C}^{15},\ldots,\mathcal{C}^{24},\mathcal{C}^{27},\ldots,\mathcal{C}^{31},\mathcal{C}^{44},\mathcal{C}^{47},\mathcal{C}^{48}\}$			20
$t_{b,\delta}$	$\{\mathcal{C}^{11},\ldots,\mathcal{C}^{16},\mathcal{C}^{39},\mathcal{C}^{50},\mathcal{C}^{56}\}$			9
Parity bits obtained from signs				
p_A	$K_1^{22}\oplus K_2^{44}\oplus K_3^{22}\oplus K_5^{22}\oplus K_6^{44}\oplus K_7^{22}\oplus K_9^{22}\oplus K_{10}^{44}\oplus K_{11}^{22}\oplus K_{13}^{22}$			
p_B	$p_A\oplus K_{13}^{22}\oplus K_{13}^{19}\oplus K_{13}^{23}$			
p_C	$K_0^{22}\oplus K_2^{22}\oplus K_3^{44}\oplus K_4^{22}\oplus K_6^{22}\oplus K_7^{44}\oplus K_8^{22}\oplus K_{10}^{22}\oplus K_{11}^{44}\oplus K_{12}^{22}$			
p_D	$p_C\oplus K_0^{22}\oplus K_0^{19}\oplus K_0^{23}$			

Table 3: This table specifies the key/text bits involved in the attack, as well as the parity key bits derived. X^i denotes the i'th bit of X.

- **Analysis**

  1. For each guess of $(k_{f,\gamma}, k_{b,\gamma})$, calculate the vector

     $$\mathbf{c}_\gamma = (C_K^N(\gamma_1), C_K^N(\gamma_2), C_K^N(\gamma_3), C_K^N(\gamma_4))^\top,$$

     by partially encrypting/decrypting the data in $\mathbf{t}_\gamma$. Do similarly for the δ-approximations to calculate $\mathbf{c}_\delta$.

  2. Calculate

     $$\Lambda(\mathbf{c}_\gamma) = \frac{\frac{1}{4}\sum_{i=1}^{4}\phi_M(\mathbf{c}_\gamma; \boldsymbol{\mu}_{i,\gamma}, \boldsymbol{\Sigma}_{i,\gamma} + (2^{-n} + 1/N)\mathbf{I})}{\phi_M(\mathbf{c}_\gamma; \mathbf{0}, \boldsymbol{\Sigma}_{W,\gamma} + (2^{-n} + 1/N)\mathbf{I})},$$

     for each guess of $(k_{f,\gamma}, k_{b,\gamma})$. If $\Lambda(\mathbf{c}_\gamma) \leq \Gamma_\gamma$, discard the key guess. Likewise, calculate $\Lambda(\mathbf{c}_\delta)$ for each guess of $(k_{f,\delta}, k_{b,\delta})$. If $\Lambda(\mathbf{c}_\delta) \leq \Gamma_\delta$, discard the key guess.

  3. For each surviving key guess, determine the four bits p_A, p_B, p_C, p_D based on the signs of $\mathbf{c}_\gamma$ and $\mathbf{c}_\delta$.

- **Search**

  1. For each remaining guess of $(k_{f,\gamma}, k_{b_\gamma}, k_{f,\delta}, k_{b,\delta})$, guess the last 4 bits of the master key, and verify the guess by trial encryption.

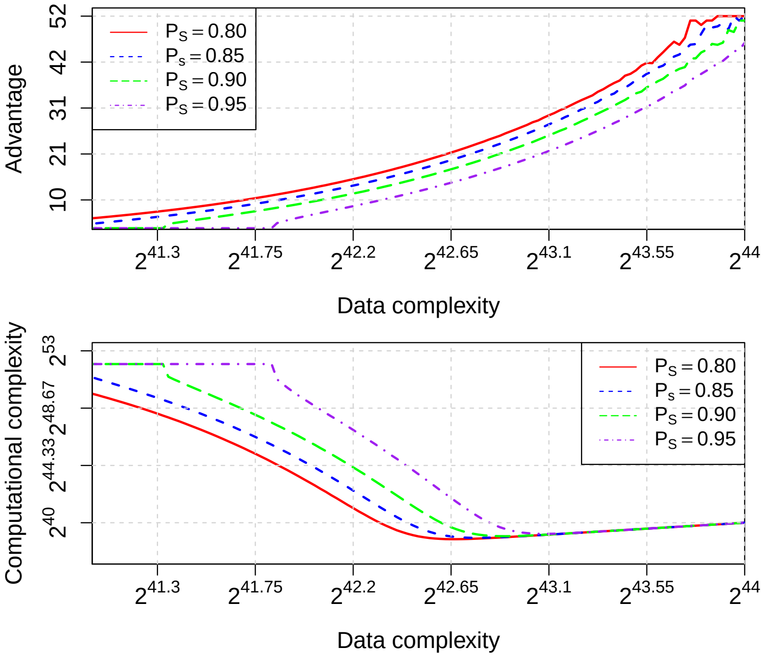


Figure 13: Top: Combined advantage of the two likelihood classifiers using approximations in Table 2. The success probabilities include the probability of guessing the four parity bits correctly. Bottom: The computational complexity of our key-recovery attack on DES. Each curve has a clear minimum where the trade-off between the data complexity and the strength of the classifiers is optimal.

6.2 Attack Complexity

In the following, we assume that one computational unit is the time it takes to perform one round of DES. The computational complexity of the distillation phase is $\mathcal{O}(N)$, while the memory complexity is $\mathcal{O}(2 \cdot 2^{29})$. For the analysis phase, each C_K^N can be calculated for all key guesses in time $\mathcal{O}((|k_{f,\cdot}| + |k_{b,\cdot}|)2^{|k_{f,\cdot}|+|k_{b,\cdot}|+1.6})$ using the FFT method presented in [9]. In total, step 1 of the analysis phase can be completed in time $\mathcal{O}(2 \cdot 4 \cdot 24 \cdot 2^{25.6}) \approx \mathcal{O}(2^{33.18})$. Step 2 requires the calculation of $\ell + 1$ terms for each key-guess of the type $(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$, to calculate the normal probabilities. Each term can be computed in time $\mathcal{O}(2M^3)$. Thus, step 2 takes a total of $\mathcal{O}(2 \cdot 2^{24} \cdot 5 \cdot 4^3) \approx \mathcal{O}(2^{33.32})$ time. Step 3 takes $\mathcal{O}(2 \cdot 2^{24-a_\gamma} + 2 \cdot 2^{24-a_\delta})$ time, where a_γ and a_δ is the advantage of the classifiers in step 2. The analysis step requires $\mathcal{O}(2^{24-a_\gamma} + 2^{24-a_\delta})$ memory to store the surviving key guesses. The search phase requires $\mathcal{O}(16 \cdot 2^{48-(a_\gamma+a_\delta)} \cdot 2^{56-52}) = \mathcal{O}(16 \cdot 2^{56-(a_\gamma+a_\delta+4)})$ time and negligible memory. Dividing everything by 16 to get the total number of full DES encryptions, the computational complexity is approximately

$$\mathcal{O}(N \cdot 2^{-4} + 2^{29.18} + 2^{29.32} + 2^{21-a_\gamma} + 2^{21-a_\delta} + 2^{52-(a_\gamma+a_\delta)}).$$

Thus, the attack complexity depends on the advantage of the two classifiers, which in turn depends on the choice of Γ_γ and Γ_δ. Note that step 3 of the analysis phase is not guaranteed to succeed, so the threshold values must be chosen such that the overall success probability of the attack is P_S. Namely, if P_γ and P_δ is the success probabilities of the two classifiers, and Q_γ and Q_δ is the success probabilities of determining the parity bits, then we fix Γ_γ and Γ_δ such that $P_\gamma \cdot P_\delta \cdot Q_\gamma \cdot Q_\delta = P_S$. Using the data obtained in Section 5.3, we calculated the total advantage $a_\gamma + a_\delta + 4$ for different N and different values of the success probability P_S. The results are shown in Figure 13, along with the corresponding attack complexities. For low data complexities, the search phase is dominant, and so the $2^{52-(a_\gamma+a_\delta)}$ term determines the time complexity. For high data complexities, however, the $N \cdot 2^{-4}$ term is dominant. This gives each complexity curve a clear minimum. In a comparison to Matsui's attack, we see that for $P_S = 85\%$, the minimum is achieved at $N = 2^{42.775}$ where the computational complexity is $2^{38.86}$ DES encryptions. This is 17.6 times faster than Matsui's attack estimate (or 4.4 times faster than Junod's estimate of the attack in [16]) using $2^{40.2}$ fewer texts.

6.3 Experimental Verification

While it would be possible to carry out the attack in practice, we would need to do this for many keys to get an idea of the actual advantage, making the experiment infeasible. Instead, we measured the actual values of $\mathbf{c}_\gamma$ and $\mathbf{c}_\delta$ over 14 and 18 rounds of DES (the right key and wrong key, respectively) with $N = 2^{42.78}$ for randomly chosen keys. This can be done in a bitsliced manner, and is therefore faster than performing the actual attack, while giving us all the information we need to verify our model. Using several months of CPU time, we collected 1300 data points for the right key and wrong key distributions. We first note that the observed distributions closely match those predicted by the model in e.g. Figure 8. Moreover, we obtain the advantages $a_\gamma = 6.72$ and $a_\delta = 10.31$, which would give us a complexity of $2^{38.88}$ – very close to that predicted by our model.

References

[1] Eli Biham and Adi Shamir. "Differential Cryptanalysis of the Full 16-Round DES". In: *Advances in Cryptology - CRYPTO '92*. 1992, pp. 487–496.

[2] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. "On Multiple Linear Approximations". In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 1–22.

[3] Andrey Bogdanov, Huizheng Geng, Meiqin Wang, Long Wen, and Baudoin Collard. "Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA". In: *Selected Areas in Cryptography - SAC 2013.* 2013, pp. 306–323.

[4] Andrey Bogdanov and Vincent Rijmen. "Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers". In: *Des. Codes Cryptography* 70.3 (2014), pp. 369–383.

[5] Andrey Bogdanov and Vincent Rijmen. "Zero-Correlation Linear Cryptanalysis of Block Ciphers". In: *IACR Cryptology ePrint Archive* 2011 (2011), p. 123.

[6] Andrey Bogdanov and Elmar Tischhauser. "On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2". In: *Fast Software Encryption - FSE 2013.* 2013, pp. 19–38.

[7] Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. "Multivariate Profiling of Hulls for Linear Cryptanalysis". In: *IACR Trans. Symmetric Cryptol.* 2018.1 (2018), pp. 101–125.

[8] Joo Yeon Cho. "Linear Cryptanalysis of Reduced-Round PRESENT". In: *Topics in Cryptology - CT-RSA 2010.* 2010, pp. 302–317.

[9] Baudoin Collard, François-Xavier Standaert, and Jean-Jacques Quisquater. "Improving the Time Complexity of Matsui's Linear Cryptanalysis". In: *Information Security and Cryptology - ICISC 2007.* 2007, pp. 77–88.

[10] Joan Daemen. "Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis". PhD thesis. KU Leuven, 1995.

[11] Joan Daemen and Vincent Rijmen. "Probability Distributions of Correlation and Differentials in Block Ciphers". In: *J. Mathematical Cryptology* 1.3 (2007), pp. 221–242.

[12] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2.

[13] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. "Multidimensional Extension of Matsui's Algorithm 2". In: *Fast Software Encryption, FSE 2009.* 2009, pp. 209–227.

[14] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. "Multidimensional Linear Cryptanalysis of Reduced Round Serpent". In: *Information Security and Privacy, ACISP 2008.* 2008, pp. 203–215.

[15] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. "Linear Cryptanalysis Using Multiple Approximations". In: *Advances in Cryptology - CRYPTO '94.* 1994, pp. 26–39.

[16] Pascal Junod. "On the Complexity of Matsui's Attack". In: *Selected Areas in Cryptography.* 2001, pp. 199–211.

[17] Lars R. Knudsen and John Erik Mathiassen. “A Chosen-Plaintext Linear Attack on DES”. In: *Fast Software Encryption, FSE 2000*. 2000, pp. 262–272.

[18] Susan K. Langford and Martin E. Hellman. “Differential-Linear Cryptanalysis”. In: *Advances in Cryptology - CRYPTO ’94*. 1994, pp. 17–25.

[19] Bruce G Lindsay. “Mixture Models: Theory, Geometry and Applications”. In: *NSF-CBMS Regional Conference Series in Probability and Statistics*. JSTOR. 1995, pp. i–163.

[20] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology - EUROCRYPT ’93*. 1993, pp. 386–397.

[21] Mitsuru Matsui. “The First Experimental Cryptanalysis of the Data Encryption Standard”. In: *Advances in Cryptology - CRYPTO ’94*. 1994, pp. 1–11.

[22] Phuong Ha Nguyen, Hongjun Wu, and Huaxiong Wang. “Improving the Algorithm 2 in Multidimensional Linear Cryptanalysis”. In: *Information Security and Privacy, ACISP 2011*. 2011, pp. 61–74.

[23] Kaisa Nyberg. “Linear Approximation of Block Ciphers”. In: *Advances in Cryptology - EUROCRYPT ’94*. 1994, pp. 439–444.

[24] Ali Aydin Selçuk. “On Probability of Success in Linear and Differential Cryptanalysis”. In: *Journal of Cryptology* 21.1 (2008), pp. 131–147.

[25] Igor A. Semaev. “New Results in the Linear Cryptanalysis of DES”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 361.

[26] *TLS stats from 1.6 billion connections to mozilla.org*. https://jve.linuxwall.info/blog/index.php?post/2016/08/04/TLS-stats-from-1.6-billion-connections-to-mozilla.org. Accessed: 07-09-2017.

[27] Jingyuan Zhao, Meiqin Wang, and Long Wen. “Improved Linear Cryptanalysis of CAST-256”. In: *Journal of Computer Science and Technology* 29.6 (2014), pp. 1134–1139.

Publication 2

Multivariate Profiling of Hulls for Linear Cryptanalysis

Publication Information

Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. “Multivariate Profiling of Hulls for Linear Cryptanalysis”. In: *IACR Transactions on Symmetric Cryptology* 2018.1 (2018), pp. 101–125

Contribution

- Main author.

Remarks

This publication has been slightly edited to fit the format.

Multivariate Profiling of Hulls for Linear Cryptanalysis

Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre

Technical University of Denmark

Abstract. Extensions of linear cryptanalysis making use of multiple approximations, such as multiple and multidimensional linear cryptanalysis, are an important tool in symmetric-key cryptanalysis, among others being responsible for the best known attacks on ciphers such as Serpent and PRESENT. At CRYPTO 2015, Huang et al. provided a refined analysis of the key-dependent capacity leading to a refined key equivalence hypothesis, however at the cost of additional assumptions. Their analysis was extended by Blondeau and Nyberg to also cover an updated wrong key randomization hypothesis, using similar assumptions. However, a recent result by Nyberg shows the equivalence of linear dependence and statistical dependence of linear approximations, which essentially invalidates a crucial assumption on which all these multidimensional models are based.

In this paper, we develop a model for linear cryptanalysis using multiple linearly independent approximations which takes key-dependence into account and complies with Nyberg's result. Our model considers an arbitrary multivariate joint distribution of the correlations, and in particular avoids any assumptions regarding normality. The analysis of this distribution is then tailored to concrete ciphers in a practically feasible way by combining a signal/noise decomposition approach for the linear hulls with a profiling of the actual multivariate distribution of the signal correlations for a large number of keys, thereby entirely avoiding assumptions regarding the shape of this distribution.

As an application of our model, we provide an attack on 26 rounds of PRESENT which is faster and requires less data than previous attacks, while using more realistic assumptions and far fewer approximations. We successfully extend the attack to present the first 27-round attack which takes key-dependence into account.

1 Introduction

Proposed by Matsui [34, 36] in the early 1990s, linear cryptanalysis has proven to be a seminal cryptanalytic technique for symmetric-key cryptography. Most prominently, linear cryptanalysis was successfully applied to the former U.S. encryption standard DES, breaking it experimentally for the first time. Influential cipher design paradigms, such as the wide-trail strategy [23], were specifically developed as a response to the advert of linear and differential cryptanalysis. Today, every newly proposed keyed symmetric primitive is expected to be accompanied by strong evidence of resistance against this attack.

In the last two decades, a number of advanced variants of linear cryptanalysis have been developed, among others differential-linear cryptanalysis [31], multiple linear cryptanalysis [5, 29], multidimensional linear cryptanalysis [25, 26, 27], zero-correlation linear cryptanalysis [16], and key-invariant bias attacks [13]. These extensions of linear cryptanalysis have provided the best single-key cryptanalytic results on ciphers such as Serpent [38], PRESENT [18, 47], CLEFIA [14], CAST-256 [46], and LBlock-s [45].

Parallel to the development of these cryptanalytic results, extensive research has been carried out to deepen our understanding of linear cryptanalysis [2] and its extensions [7], e.g. concerning links between differential and linear cryptanalysis [11] and truncated differential and multidimensional linear techniques [10]. How to provide resistance against these advanced cryptanalysis techniques has been studied in [6, 44].

Key-dependence in Linear Cryptanalysis. Linear cryptanalysis relies on identifying linear relations between the input and output bits of a cipher which exhibit large linear correlations. The correlation can be viewed as a random variable over the space of inputs as well as over the space of encryption keys. A central question in linear cryptanalysis is therefore this: *What is the stochastic behaviour of the linear correlation?*

While early analysis assumed that this behaviour was largely identical for all keys [4, 30, 34, 36, 42, 46], and so only depends on the randomness of the plaintexts, several works have demonstrated that this is not true in general [2, 32], and models have been developed for the key-dependent behaviour of the correlation of a single linear approximation [17, 22]. These models assert that the linear correlation follows a normal distribution, both in the case of a random permutation and specific block ciphers.

Even though we have a good understanding of the key-dependent behaviour of *single* approximations, it is only recently that the key-dependent behaviour of *multiple* approximations has been studied, despite the relatively large number of multiple and multidimensional linear attacks in the literature. In this work, we consider the three principal papers on this topic and reflect on the precise assumptions used by the models developed by them. We then develop a new model which aims to remove many of these assumptions in order to obtain more accurate estimates of the power of linear attacks.

State of the Art and Problems. There are three principal works considering key-dependence in the context of multiple and multidimensional linear cryptanalysis. First, [28] by Huang et al. considers the key-dependent behaviour of the multiple and multidimensional capacity and develops a model in which this follows a gamma distribution under the assumption that the individual correlations are independently and identically distributed. Second, [9] by Blondeau and Nyberg relaxes the assumptions of [28] such that the correlation distributions need not have identical means, which results in a model that describes the capacity as a scaled, non-central χ^2-distribution. However, this model assumes an accurate estimate of the parameters of the correlation distributions. Blondeau and Nyberg relaxed this assumption in [8] by incorporating the signal/noise decomposition from [17] into the model. Although the models developed in these works are a step on the way towards accurate assessments for multiple and multidimensional attacks, we identify the following main problems with the approaches:

- **Independence assumptions:** Multidimensional linear cryptanalysis was originally introduced to solve the requirement for statistically independent approximations, but recently Nyberg showed [40] that under reasonable assumptions about pair-wise statistical independence, linear independence and statistical independence of approximations are equivalent concepts. Multidimensional linear cryptanalysis uses many linearly dependent approximations, but the models described above often assume these to be statistically independent for a random permutation. Moreover, the models are typically derived in a setting with independent round keys – a setting that does not strictly reflect most actual ciphers.

- **Restricted approximation choice:** The models described above restrict which approximations can be used. In the case of multiple linear cryptanalysis, equal correlation variances are required, and so we cannot necessarily freely choose the approximations that best facilitate an attack, as they might have different distributions. Ideally, a cryptanalyst would like to be able to pick the best trade-off between strong approximations and approximations that make the attack efficient to perform. For multidimensional linear cryptanalysis, models are given in which a set of dominant approximations are present and the rest of the approximations are treated as noise. The advantage of the multidimensional approach then seems to stem from the fact that the attacker can sometimes get a few rounds for free, if the resulting approximations still allow for efficient key guessing.

- **Parameter estimation:** As mentioned, the models of [9, 28] require an accurate knowledge of the correlation distributions or multidimensional probability distributions. Obtaining this is extremely difficult for most reasonable block and key sizes. This problem is mostly solved in [8] by applying the signal/noise decomposition, but this approach is still quite computationally expensive if simplifying assumptions, such as independent round keys, are not used. In general, this problem seems to be quite difficult to avoid.

Our Results. The results of [40] poses a problem for any model of linear cryptanalysis with multiple approximations that uses linearly dependent approximations, including multidimensional linear attacks. This paper therefore revisits multiple linear cryptanalysis in the case where all approximations are linearly independent.

We first investigate the joint correlation distribution of such a set of approximations. We find that this distribution can be assumed to be jointly normal for a long-key cipher, in accordance with theory, but that this is not the case for other key-schedules. We therefore propose *multivariate linear cryptanalysis*. This model:

- Does not assume a specific key-schedule,
- Does not assume statistical independence of the correlations,
- Is able to model any arbitrary (not necessarily normal) joint correlation distribution,
- Uses signal/noise decomposition to practically obtain accurate attack estimates.

The model expresses the joint correlation distribution of M approximations as a general M-variate probability distribution. While the multivariate model relaxes many assumptions used by previous models, it comes at the cost of a larger effort during the off-line analysis of the cipher. In particular, the more accurate an estimate of the signal distribution the cryptanalyst can obtain the better. This only affects the amount of effort she has to put into the analysis, and not the effectiveness of the resulting attack. We confirm the accuracy of our model through experiments on 32-bit SmallPresent.

As a result, we are able to present new attacks on present (with an 80-bit key), which at the same time avoid the above modeling problems. Crucially, our analysis model is in accordance with [40]. We identify a very sparse set of 135 approximations over 22 rounds, and use these to attack 26 rounds of present. The computational complexity of this attack is $2^{68.6}$, while the data complexity is $2^{63.0}$. Interestingly, this attack is about 11 times faster than Cho's original attack on the same number of rounds, and uses half the data, all the while using far fewer approximations and more realistic assumptions. This demonstrates that a multidimensional linear attack is not necessarily stronger than a multiple linear attack. We extend the attack to 27 rounds, resulting in a computational complexity of $2^{77.3}$ and a data complexity of $2^{63.8}$. This is the first attack on 27 rounds of present in a model that accounts for key-dependence. Our attacks are compared to previous attacks on present in Table 1.

2 Preliminaries

We consider a block cipher $E(P, K) : \mathbb{F}_2^n \times \mathbb{F}_2^\kappa \to \mathbb{F}_2^n$ with a block size of n bits and key length of κ bits. For each key $K \in \mathbb{F}_2^\kappa$, $E_K := E(\cdot, K)$ is a permutation on $\mathbb{F}_2^n$. If

Rounds	*Success probability*	*#Approximations*	*Time complexity*	*Data complexity*	*Memory complexity*	*F1: Key-dependent*	***F2: Complies with [40]***	*T1: Signal/noise*	*T2: Profiling*	**Reference**
25	95%	2295	$2^{65.0}$	$2^{62.4}$	$2^{34.0}$		N/A			[18]
	95%	2295	$2^{65.0}$	$2^{61.6}$	$2^{34.0}$	✓				[28]
	74%	2295	$2^{72.0}$	$2^{61.0}$	$2^{34.0}$	✓				[8]
26	95%	2295	$2^{72.0}$	$2^{64.0}$	$2^{34.0}$		N/A			[18]
	80%	2295	$2^{76.0}$	$2^{62.5}$	$2^{34.0}$	✓				[28][1]
	51%	2295	$2^{72.0}$	$2^{63.8}$	$2^{34.0}$	✓		✓		[8]
	95%	**135**	$\mathbf{2^{68.6}}$	$\mathbf{2^{63.0}}$	$\mathbf{2^{48.0}}$	✓	✓	✓	✓	**Section 6.2**
27	95%	405	$2^{74.0}$	$2^{64.0}$	$2^{70.0}$		N/A			[47]
	95%	**135**	$\mathbf{2^{77.3}}$	$\mathbf{2^{63.8}}$	$\mathbf{2^{48.0}}$	✓	✓	✓	✓	**Section 6.3**[2]

1: For 3.7% of the key space.

2: Uses distinct texts. All other attacks use non-distinct texts.

Feature/Technique	**Explanation**
F1: Key-dependent	The model accounts for the fact that the linear correlation of an approximation varies over the key space.
F2: Complies with [40]	The model does not assume that linearly dependent approximations of a random permutation are statistically independent. Doing so contradicts [40].
T1: Signal/noise	The model uses the signal/noise decomposition of [17] to obtain accurate estimates of the correlation distributions.
T2: Profiling	The model measures the actual multivariate distribution of the signal for a large number of keys to avoid assumptions of the shape of this distribution.

Table 1: Comparison of attacks on PRESENT. The attacks of [18] and [47] do not take the key-dependence into account. All models, except the one presented in this work, use assumptions that contradict the equivalence of linear independence and statistical independence of linear correlations shown in [40].

a block cipher picks a permutation uniformly at random from the space of all $(2^n)!$ permutations for each key, we say that it is *ideal.*

Most modern block ciphers are *iterative block ciphers* where the encryption function is a composition of r key-dependent round functions. If each round function can be described as a key-independent transformation followed by an XOR of the round key, we call the cipher a *key-alternating cipher.* Additionally, an initial key is XOR'ed to the input before the first round. Usually, a *key-schedule* is used to expand the κ-bit master key K into the required $r+1$ n-bit round keys. We denote the expanded key by $\bar{K} = k_0 \| k_1 \| \ldots \| k_r$, i.e. the concatenation of the round keys. If all round keys are chosen uniformly and independently, i.e. $\kappa = (r+1)n$ and $K = \bar{K}$, we call the cipher a *long-key cipher*.

2.1 Linear Cryptanalysis

Linear cryptanalysis was introduced by Matsui in 1993 [34] and considers one or more *linear approximations* of a cipher. A linear approximation is a pair $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n \backslash (0,0)$, where α is called the *input mask* and β the *output mask*. The key-dependent *linear correlation* of the approximation is defined as $C^K_{\alpha,\beta} = 2\Pr(\alpha \cdot x = \beta \cdot E_K(x)) - 1$, where "$\cdot$" denotes the canonical inner product on $\mathbb{F}_2^n$, and the probability is taken over all $x \in \mathbb{F}_2^n$. Assuming that K is drawn uniformly at random, $C^K_{\alpha,\beta}$ is a random variable over the key space. If an estimate of $C^K_{\alpha,\beta}$ is calculated using N plaintext-ciphertext pairs, we denote this value by $C^{K,N}_{\alpha,\beta}$, which is a random variable over both the key and text space, where the latter is of size N. The goal of linear cryptanalysis is to find pairs (α, β) such that the probability distribution of the correlation for the block cipher in question is distinguishable from the correlation distribution of an ideal cipher.

Let (u_i, u_{i+1}), $i = 0, \ldots, r-1$, be a series of one round linear approximations of an iterative block cipher. Such a series of approximations is called a *linear trail.* We can also denote the trail by the concatenation of its masks, i.e. $U = u_0 \| \ldots \| u_r$. Then the *correlation contribution* of trail is defined by $C^K_U = \prod_{i=0}^{r-1} C^K_{u_i,u_{i+1}}$. The collection of all trails with $u_0 = \alpha$ and $u_r = \beta$ is called the *linear hull* of (α, β). Moreover, the correlation of (α, β) is the sum of the correlation contributions of all trails in the hull [20, 23]:

$$C^K_{\alpha,\beta} = \sum_{u_0=\alpha, u_r=\beta} C^K_U . \tag{1}$$

A useful concept is that of the *expected linear potential (ELP)*, defined by $\mathrm{E}((C^K_{\alpha,\beta})^2)$. For a long-key cipher, it can be shown that $ELP = \sum (C^K_U)^2$, and that $(C^K_U)^2$ is independent of the key [23].

2.2 Statistical Distinguishing

In cryptanalysis of block ciphers, a first step towards more powerful attacks is often to build a *distinguisher.* A distinguisher aims to determine whether some observed data is the output of a specific block cipher or an ideal cipher. In statistical cryptanalysis, a distinguisher consists of performing a statistical test which distinguishes between two probability distributions. Typically, the test computes a value from the data, which we refer to as the *test statistic* $\mathcal{T}$.

Note that the test statistic is a random variable. Let $\mathcal{T}_I$ be the random variable if the observed data was produced by an ideal cipher, and let $\mathcal{T}_N$ be the random variable if the observed data was produced by a specific block cipher. Assume that

$\mathcal{T}_I$ and $\mathcal{T}_N$ follow univariate distributions. Then a simple and often used statistical test is to check the value of $\mathcal{T}$ against some *threshold value* τ. Without loss of generality, assume that $\mathcal{E}(\mathcal{T}_I) \leq \tau \leq \mathcal{E}(\mathcal{T}_N)$. If $\mathcal{T} \geq \tau$, we conclude that $\mathcal{T}$ was drawn from the distribution of $\mathcal{T}_N$, otherwise we conclude that $\mathcal{T}$ was drawn from the distribution of $\mathcal{T}_I$. It may be the case that we need to compare against multiple threshold values – for a discussion of this case, we refer to [9]. Note that we can define several different tests of the type described above, namely by calculating the test statistic $\mathcal{T}$ in different ways. We consider a commonly used test statistic in Section 6.1, namely the χ^2 test statistics.

When assessing the efficiency of a threshold test, we are mainly interested in two parameters: the *success probability* and the *advantage*. Let F_X denote the cumulative distribution function of the random variable X. We define the probability of success as

$$P_S = 1 - F_{\mathcal{T}_N}(\tau),$$

i.e. the probability that $\mathcal{T}_N \geq \tau$. The advantage, a notion first introduced by Selçuk in [42, 43] in the context of key-ranking, is in turn defined by

$$a = -\log_2(1 - F_{\mathcal{T}_I}(\tau)),$$

and relates to the number of false successes that arises from the threshold test. This number is important when we want to use a distinguisher as part of a key recovery attack. In order to assess these quantities, we need to know the distributions of $\mathcal{T}_I$ and $\mathcal{T}_N$, and the question of determining these is therefore central to the study of linear cryptanalysis.

From Distinguishing to Key Recovery.

It is possible to turn a distinguisher over r rounds of an iterative block cipher into a key recovery attack over $r' > r$ rounds in a generic way. Consider the case $r' = r + 1$ as an example. Denote by E^r the r-round encryption function, and let F_k denote the last round function such that $E^{r'} = F_k \circ E^r$. Let $\bar{E}^r$ be the truncation of E^r such that only the bits required to calculate the test statistic $\mathcal{T}$ are output.

The attacker obtains some data from $E^{r'}$, and guesses the parts of k required to partially invert F_k and calculate the output of $\bar{E}^r$. The attacker then calculates the test statistic $\mathcal{T}$ and runs the distinguisher. If the attacker guessed the partial key k correctly, the distinguisher should indicate that $\mathcal{T}$ was drawn from the distribution of $\mathcal{T}_N$ with probability P_S. If not, the hypothesis is that the distinguisher will behave as if $\mathcal{T}$ was drawn from the distribution of $\mathcal{T}_I$. The reasoning here is that for a wrong key guess, the attacker is basically observing data from a cipher with $r + 2$ rounds, which should behave more like an ideal cipher than a cipher with r rounds. This idea was first formally stated by Harpes et al. [24] and later stated in the context of linear cryptanalysis by Junod [30]. Once all candidates for the partial key k have been tested, the attacker has to guess the remaining bits of the master key K, discarding any wrong guesses by trial encryption. By definition of the advantage, the attacker has to try $2^{\kappa-a}$ candidates.

2.3 PRESENT

PRESENT is an ultra-lightweight, key-alternating, block cipher. It is an SPN cipher with 31 rounds, a block size of 64-bit, and a key size of either 80 bit or 128 bit. Each round consists of an XOR with a round key, a layer of 16 parallel 4-bit S-boxes, and bit permutation. An additional round key is added after the last round. The 32 round keys are derived through a key-schedule. For details on the bit permutation and the key-schedule, we refer to [15]. Due to the choice of S-box, PRESENT exhibits some interesting linear properties [41]. It is therefore a common target for new linear cryptanalysis techniques. We consider new attacks on PRESENT in Section 6.

3 Survey of Previous Work

As discussed in Section 2.2, it is of primary interest to determine the distributions of $\mathcal{T}_I$ and $\mathcal{T}_N$ for a given statistical test. For linear cryptanalysis, the test statistic is derived from the observed correlation of one or more linear approximations. An equivalent question in this context is therefore what the distribution of the correlation $C^N_{\alpha,\beta}$, for a given approximation or set of approximations, looks like, both for a specific block cipher and for an ideal cipher. Starting with [39], this topic has been extensively investigated in the literature. In the following, we consider a series of models that have been proposed since the introduction of linear cryptanalysis, and reflect on their assumptions and requirements. We divide the models into two main categories: models that assume that $C^K_{\alpha,\beta}$ is approximately equal for all keys, and models that include the influence of the key.

3.1 Models Without Key Influence

Matsui introduced linear cryptanalysis in [34, 36] as a means to attack DES. The approximations used for this attack exhibit a single *dominant trail* each, i.e. there exists a trail U such that $|C^K_U| \gg |C^K_{U'}|$ for any $U' \neq U$. Then by Equation 1, $C^K_{\alpha,\beta} \approx C^K_U$ for all keys. Moreover, it can be shown that for key-alternating ciphers (or ciphers that can be expressed as such, e.g. DES) the correlation contribution is given by $C^K_U = (-1)^{U\cdot\bar{K}}|C^K_U|$, where $|C^K_U|$ is independent of the key [23]. Thus, Matsui asserts that for DES, $C^K_{\alpha,\beta} \approx \pm|C^K_U|$ for all keys. This leads to the concept of *right-key equivalence*:

Hypothesis 2 (Right-Key Equivalence – Matsui)**.** *If a linear approximation (α, β) has a single dominant trail U, then the absolute value of the linear correlation is approximately equal for all keys, with $|C^K_{\alpha,\beta}| \approx |C^K_U|$.*

Similarly, Matsui assumed that for a wrong key guess, the correlation would be approximately zero for all keys, leading to the concept of *wrong-key randomisation*:

Hypothesis 3 (Wrong-Key Randomisation – Matsui)**.** *During a key recovery attack, the linear correlation of a linear approximation* (α, β) *is approximately equal to zero for all wrong keys, i.e.* $C^K_{\alpha,\beta} = 0$*.*

Under Hypotheses 2 and 3 the distribution of $C^{K,N}_{\alpha,\beta}$ only depends on the number N of observed plaintext-ciphertext pairs. Using a normal approximation to the binomial distribution, it can be shown that

$$C^{K_R,N}_{\alpha,\beta} \sim \mathcal{N}(\pm|C^K_U|, N^{-1}) \quad \text{and} \quad C^{K_W,N}_{\alpha,\beta} \sim \mathcal{N}(0, N^{-1}) \tag{2}$$

where K_R and K_W represents a right and wrong key guess, respectively. This and similar models have been used extensively in the literature, first in classical linear cryptanalysis [4, 30, 34, 36, 42, 46], and later in its extensions *multiple linear cryptanalysis* [5, 29] and *multidimensional linear cryptanalysis* [18, 25, 26, 27, 41, 47]. Notably, the best attacks on the block cipher PRESENT (both multidimensional), the 26-round attack by Cho [18] and the 27-round attack by Zheng and Zhang [47], both use this model.

3.2 Models Incorporating the Key

Single Approximations

While the idea of identical behaviour for all keys simplifies analysis, it does not reflect the behaviour of most modern ciphers. Indeed, if the number of trails with a significant correlation contribution is large, then by Equation 1 the correlation $C^K_{\alpha,\beta}$ will take on many values over the key space. Dubbed the *linear hull effect*, this phenomenon was first discussed by Nyberg in [39]. Ohkuma later pointed out that for PRESENT this effect is very strong, as the number of trails with the same best correlation contribution is large [41]. The situation is similar for most other modern ciphers designed with resistance to linear cryptanalysis in mind. Thus, Hypothesis 2 is not true for most ciphers of interest.

Although the correlation $C^K_{\alpha,\beta}$ is a random variable over the key space, it is not immediately clear what distribution it follows. For a long-key cipher, it can be shown that the distribution is normal with mean zero and variance equal to ELP [21]. For other key-schedules, the distribution has been studied in several works [2, 17, 32], and have been found to be close to normal – however, the key-schedule can have an impact on the parameters of the distribution, invalidating the veracity of Hypothesis 2. This leads to the following revised right-key hypothesis, which has been used several times in the literature [8, 9, 28].

Hypothesis 4 (Right-Key Randomisation – Single [17, 22])**.** *The linear correlation* $C^K_{\alpha,\beta}$ *of a linear approximation* (α, β) *of a block cipher, which does not have a single dominant trail, is a random variable over the key space with distribution* $C^K_{\alpha,\beta} \sim \mathcal{N}(\mu, \sigma^2)$*.*

Note that by the definition of ELP and variance, we can write $\sigma^2 = ELP - \mu^2$. Moreover, for a long-key cipher, $\mu = 0$ [22, 23]. For the wrong-key, the situation is a little simpler. In [22], Daemen and Rijmen show that the correlation distribution of an ideal cipher is normal with mean zero and variance 2^{-n}. Thus, we obtain the following hypothesis in this case.

Hypothesis 5 (Wrong-Key Randomisation – Single)**.** *During a key recovery attack, for a wrong key guess, the linear correlation $C^K_{\alpha,\beta}$ of a linear approximation (α, β) is a random variable with distribution $C^K_{\alpha,\beta} \sim \mathcal{N}(0, 2^{-n})$.*

While the picture seems clear in the case of a single approximation, moving to extensions that use multiple approximations simultaneously in order to extract more information seems to complicate matters considerably.

Multiple Linear Cryptanalysis

Kaliski and Robshaw first proposed the use of multiple approximations simultaneously in [29]. The idea was extended by Biryukov et al. in [5], where they also defined the *capacity* of a set of linear approximations as a measure of the strength of this set. For a set of M approximations $(\alpha_1, \beta_1), \ldots, (\alpha_M, \beta_M)$, the capacity is defined as

$$\mathcal{C}^K = \sum_{i=1}^{M} (C^K_{\alpha_i,\beta_i})^2. \tag{3}$$

Similar to the correlations, we denote an estimate of the capacity based on N plaintext-ciphertext pairs by $\mathcal{C}^{K,N}$. The main problem with this approach is that the linear approximations are not in general statistically independent, making the analysis of the capacity very difficult. Indeed, statistical independence was assumed in [5, 29]. This approach is commonly referred to as *multiple linear cryptanalysis*.

Multidimensional Linear Cryptanalysis

To avoid the problem of independence, Hermelin et al. proposed *multidimensional linear cryptanalysis* in [25, 26], based on the work done by Baignères et al. in [3]. It considers an m-dimensional subspace of $\mathbb{F}_2^n \times \mathbb{F}_2^n$ and studies the distribution of a plaintext-ciphertext pair $(\bar{x}, \bar{E}_K(x))$ restricted to this subspace, which can be described by the vector $\boldsymbol{\eta}^K = (\eta^K_0, \ldots, \eta^K_{2^m-1})$, where $\eta^K_i = \Pr(\bar{x} \| \bar{E}_K(x) = i)$. $\boldsymbol{\eta}^K$ is a key-dependent, 2^m-dimensional, discrete probability distribution. It can then be shown that the capacity of the set of all linear approximations in the subspace can be calculated from $\boldsymbol{\eta}^K$.

Theorem 1. *[26] Consider an m-dimensional subspace of $\mathbb{F}_2^n \times \mathbb{F}_2^n$, and denote the multidimensional probabilities by η^K_i. The capacity of all linear approximations in* *this subspace can be calculated as*

$$\mathcal{C}^K = \sum_{i=1}^{2^m-1} (C^K_{\alpha_i,\beta_i})^2 = \sum_{i=0}^{2^m-1} \frac{(\eta_i^K - 2^{-m})^2}{2^{-m}},$$

The main advantage of multidimensional linear cryptanalysis is that it can be shown that the amount of data needed for a multidimensional distinguisher (with a fixed success probability) is inversely proportional to the capacity, regardless of statistical dependence of the associated approximations [3].

While the influence of the key on the correlation of a single approximation has been studied for some time, it is only recently that versions of Hypotheses 4 and 5 have been developed for multiple and multidimensional linear cryptanalysis. In the following, we give a short summary of the contributions of the three main works in this area, and in Section 4 we consider their results in depth.

Huang et al., CRYPTO'15 [28] To the best of our knowledge, this is the first work to study the key-dependent distribution of the multidimensional capacity, although the wrong-key capacity is not considered. Under some assumptions on the one-dimensional approximations, it is shown that the capacity follows a gamma distribution. Two cases are considered giving the following results.

Result 1 ([28], Proposition 2)**.** *Consider an m-dimensional linear approximation where m linearly independent base approximations have dominant ELPs. Moreover, let the correlations of these base approximations, $C^K_{\alpha_1,\beta_1}, \ldots, C^K_{\alpha_m,\beta_m}$, be i.i.d as $\mathcal{N}(0, ELP)$. Then $\mathcal{C}^K \sim \Gamma(\frac{m}{2}, 2ELP) = ELP \cdot \chi^2_m$.*

Result 2 ([28], Proposition 3)**.** *Consider an m-dimensional linear approximation with probability distribution $\boldsymbol{\eta}^K = (\eta_0^K, \ldots, \eta^K_{2^m-1})$. Assume that the multidimensional probabilities η_i^K are i.i.d as $\mathcal{N}(2^{-m}, \sigma^2)$. Then $\mathcal{C}^K \sim \Gamma(\frac{2^m-1}{2}, 2^{m+1}\sigma^2) = 2^m\sigma^2 \cdot \chi^2_{2^m-1}$.*

Blondeau and Nyberg, DCC'17 [9] This work improves upon [28] in several ways. First, both the key and data dependence are included in the models, as opposed to [28] that only consideres the exact distribution of capacity. Moreover, both sampling of the texts with and without replacement is considered; here, we will only cover the case without replacement, and refer to [9] and [12] for further details.

A model for the wrong-key is derived by using Hypothesis 5 and Theorem 1, under the assumption that approximations of ideal ciphers are statistically independent.

Result 3 ([9], Theorem 6)**.** *Consider a multiple or multidimensional attack using M approximations and N text pairs. Then, for a wrong key guess, $\mathcal{C}^{K,N} \sim (N^{-1} + 2^{-n})\chi^2_M$.*

For the right-key, [9] considers a more general case where the mean of the correlations is not necessarily zero. Let $\chi^2_\ell(k)$ be the non-central χ^2-distribution with ℓ degrees of freedom and non-centrality parameter k. The following result is given.

Result 4 ([9], Theorem 7 and 8)**.** *Consider a multiple or multidimensional attack using M approximations and N text pairs. For a multiple attack, assume that the linear correlations of the approximations, $C^K_{\alpha_i,\beta_i}$, are independently distributed as $\mathcal{N}(\mu_i, \sigma^2)$, $i = 1, \ldots, M$. For a multidimensional attack, assume that the multidimensional probabilities η_i^K are normally distributed with equal variances and that each set of M probabilities are statistically independent. Let μ_i be the mean of the correlation of the related approximation, $i = 1, \ldots, M$. Then*

$$\mathcal{C}^{K,N} \sim \Delta\chi^2_M\left(\frac{N\sum\mu_i^2}{N\Delta}\right) \quad \textit{where} \quad \Delta = N^{-1} + M^{-1}\sum(ELP_i - \mu_i^2).$$

For the multidimensional probabilities, note that the assumption of statistical indepedence of sets of size M arises since $\sum \eta_i^K = 1$.

Blondeau and Nyberg, ToSC'16 [8] While [9] derives the capacity distributions under some assumptions, Result 4 requires that the cryptanalyst can get accurate estimates of the distribution parameters of the one-dimensional correlations or the multidimensional probabilities. Obtaining these is left as an open problem. [8] aims to solve this problem by utilising the signal/noise decomposition technique developed in [17].

The idea of the signal/noise decomposition is to first get an estimate of the correlation distribution by computing a part of the linear hull, i.e. some (significant) terms of Equation 1. We call this set of known trails the *signal*, denoted by $\mathcal{S}$. Then, the unknown part of the hull, i.e. the trails not in $\mathcal{S}$, are modeled as *noise* with the distribution $\mathcal{N}(0, 2^{-n})$. We will take a closer look at this method in Section 5.2. Using the signal/noise decomposition, the following result is given for the right-key distribution of capacity. Note that [8] uses the wrong-key result given in Result 3.

Result 5 ([8], Theorem 4)**.** *Given M linear approximations, assume that a signal $\mathcal{S}$ is known for ℓ approximations, and that the noise of these ℓ approximations, as well as the correlations of the remaining $M - \ell$ approximations, are statistically independent. Let $C_\mathcal{S} = \sum_{i=1}^{\ell}\sum_{U\in\mathcal{S}_i}(C_U^K)^2$ be the signal capacity. Then, for a long-key cipher,*

$$\mathrm{E}(\mathcal{C}^K) = C_\mathcal{S} + M2^{-n}, \quad \text{and}$$

$$\mathrm{Var}(\mathcal{C}^K) = 2\sum_{i=1}^{\ell}\left(\sum_{U\in\mathcal{S}_i}(C_U^K)^2\right)^2 + C_\mathcal{S}2^{2-n} + M2^{1-2n}.$$

4 Limitations of Current Models

The results described in Section 3 use one or more assumptions about the linear correlation distributions. Moreover, the results are not as general as a cryptanalyst might want, i.e, the situations in which they can be used are restricted in some way. In the following, we consider the validity of these assumptions and describe some of these restrictions.

4.1 Independence Assumptions

Dealing with statistical independence has long been a problem for linear cryptanalysis. Indeed, the very reason for the introduction of multidimensional linear cryptanalysis was to avoid this issue. When trying to incorporate the key-dependence in the models, however, it seems difficult to avoid assumptions on the statistical behaviour of the approximations. We note that Results 1 to 5 all use some assumptions on the statistical independence of (some of) the approximations. Recently, Nyberg proved the following theorem:

Theorem 2 ([40])**.** *Let A be a set of pair-wise statistically independent linear approximations. Then the correlations of the linear approximations in A are statistically independent if and only if they are linearly independent.*

While it is an open problem to formally prove when two approximations are statistically independent, for all practical intents and purposes, assuming pair-wise statistical independence seems reasonable in the case of random permutations of the block size used in practice. With this assumption in mind, let us consider a general set of M linear approximations, (α_i, β_i), $i = 1, \ldots, M$. We denote the vector of their correlations by $\mathbf{C}^K = (C^K_{\alpha_1,\beta_1} \cdots C^K_{\alpha_M,\beta_M})^\top$. Under the wrong-key hypothesis, Hypothesis 5, $C^K_{\alpha_i,\beta_i} \sim \mathcal{N}(0, 2^{-n})$, $i = 1, \ldots, M$. In this case, if the approximations are linearly independent, Theorem 2 asserts that $\mathbf{C}^K \sim \mathcal{N}_M(\mathbf{0}, \mathrm{diag}(2^{-n}))$. But this is not the case if the linear approximations are linearly dependent, which poses an interesting problem for the multidimensional models. In particular, not all the one-dimensional approximations are linearly independent, and so by Theorem 2, they cannot be statistically independent. The consequence for Result 3 is that it is unknown whether the capacity is χ^2-distributed in a multidimensional linear attack. For a multiple linear attack the result still holds if the approximations are linearly independent.

For the right-key models, Theorem 2 has the biggest impact on Result 5. When adding noise to the model, the assumption is that the noise distributions behave as for a random permutation and are independent, but this cannot be the case for a multidimensional approximation. For Results 2 and 4, it is assumed that the multidimensional probabilities are independent, and thus Theorem 2 does not affect these models. Whether this assumption is sound is an open problem.

Finally, we note that an often used assumption when deriving these models is that the cipher is a long-key cipher, where pair-wise statistical independence might also be a reasonable assumption in practice. In this case, we could choose linearly independent approximations, and then by Theorem 2 and [21], $\mathbf{C}^K \sim \mathcal{N}_M(\mathbf{0}, \mathrm{diag}(ELP_i))$. However, most ciphers do not actually have independent round keys. If a key-schedule is used, we can no longer use Theorem 2 to equate linear independence with statistical independence. Moreover, we cannot even guarantee that the distribution is jointly normal. We take a close look at the key-schedule influence in the following.

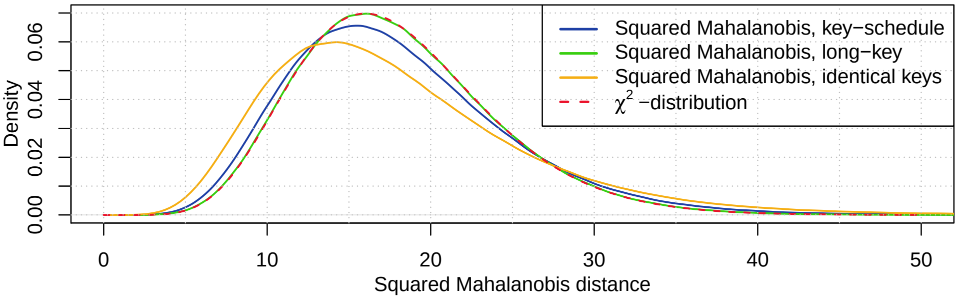


Figure 1: The densities of the squared Mahalanobis distance of the joint correlation distribution for 18 approximations over 9 rounds of 32-bit SmallPresent for three different key-schedules. The plot show a connection between dependence between the round keys, and how much the correlation distribution deviates from joint normality.

Non-Normality of Linearly Independent Approximations

In light of Theorem 2, the joint correlation distribution of multiple linear approximations of an ideal cipher is currently unknown. Since knowledge of this distribution is crucial to linear cryptanalysis, it seems safer to consider sets of linearly independent approximations. But how do these behave for a specific block cipher that does not have independent round keys? To investigate this, we consider a set of 18 linearly independent approximations over 9 rounds of 32-bit SmallPresent [33]. The input and output masks are given by

$$\alpha = 2^{4i+3}, i \in 5, 6, 7, \quad \text{and} \quad \beta = 2^{4i+j}, i \in 5, 6, 7, j \in 2, 3.$$

We note that these approximations have the same form as those we will later use to attack present in Section 6. We consider three different key-schedules: long-key, identical round keys, and a 40-bit key-schedule described in Section A. For each key-schedule, we calculated the linear correlation of each approximation for the full code-book and 2 000 000 randomly chosen keys. Let $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ be the mean vector and covariance matrix of each of the data sets, respectively. To assess how much the distribution of $\mathbf{C}^K$ deviates from joint normality, we consider the squared Mahalanobis distance, defined by

$$d^2 = (\mathbf{C}^K - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{C}^K - \boldsymbol{\mu}).$$

Note that if $\mathbf{C}^K \sim \mathcal{N}_{18}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $d^2 \sim \chi^2_{18}$. Figure 1 shows the density of d^2 for the three data sets against the density of the χ^2-distribution.

We make the following observations: For the long-key, the joint distribution of $\mathbf{C}^K$ is very close to the multivariate normal distribution $\mathcal{N}_{18}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. When we switch to a key-schedule with dependent round keys, we observe a deviation from normality. The most drastic effect is seen in case of the strongest dependence between the round keys, namely for identical round keys. Here, the distribution of d^2 is heavier towards zero, but also has a heavier tail towards infinity, compared to the χ^2_{18}-distribution. For such a key-schedule, it does not seem reasonable to approximate the distribution of $\mathbf{C}^K$ by a multivariate normal distribution. For the 40-bit key-schedule, the distribution of d^2 also deviates somewhat from χ^2_{18}. The 40-bit key-schedule we have used here is a scaled down version of the 80-bit key-schedule used in PRESENT, and so it might be natural to assume that the cipher behaves as a long-key cipher, in order to simplify analysis. However, there is still quite some overlap of the bits in consecutive round keys, which seems to have a non-negligible influence on the shape of the joint correlation distribution. It would then seem that, strictly speaking, joint normality is not a fair assumption, even for good key-schedules.

4.2 Restricted Approximation Choices

The right-key models of [9, 28] set certain requirements for the set of approximations used. The primary requirement is on the parameters of either the correlation or multidimensional probability distributions. For Results 1 and 2, the assumption is that all the distributions are identical. For Result 4, the assumption is that the distributions have identical variances. Although it might be possible to find sets of approximations such that these assumptions are satisfied, it does restrict the ability of the cryptanalyst to freely choose a set of approximations that can optimally facilitate an attack. This can for example make it hard to do efficient key-guessing, and so would result in a worse attack than if the cryptanalyst could choose approximations freely.

While the use of the multidimensional probability distribution in Result 2 is promising, it seems that there are more works that analyse the correlation distributions directly – perhaps because the distribution of these is more well understood. For models that use the correlation distributions directly, it seems that these are currently either multiple (Result 4) or multidimensional with similar restrictions to the multiple case (Results 1 and 5). For Result 1, a set of (linearly independent) dominant base approximations are required, and so the combined approximations derived from these cannot by assumption contribute significantly to the attack. For Result 5, the noise part of the ℓ known approximations are modelled as approximations of a random permutation and must be independent, and so by Theorem 2 and Section B, they must be linearly independent. Additionally, the remaining approximations only contribute with noise.

4.3 Parameter Estimation

As noted by [8], one major challenge when trying to apply Results 1, 2 and 4 is to get an accurate estimate of the various distribution parameters. For single approximations, this problem was identified in [17] and the signal/noise decomposition was proposed. This approach was nicely applied in [8], and was shown to give more accurate results. However, [8] uses the long-key assumption to avoid considering the actual distribution of the signal, instead only considering the signal ELP. Extending the discussion of Section 4.1, this might not be accurate for other key-schedules. In this case, the cryptanalyst would have to get an estimate of the actual signal distribution.

To estimate the parameters of the signal, one could find a set of trails with large correlation contribution, and calculate part of the sum in Equation 1 for a significant number of randomly chosen keys. Doing this can be a significant challenge, especially for PRESENT-like ciphers where the number of good trails is extremely large. Various methods for finding good trails of a cipher have been proposed, e.g. the branch-and-bound method [35] and sparse correlation matrices [1], but it can still be quite the computational challenge to obtain good parameters for the signal. In Section 6, we use a method similar to that of [1] and significant computational power to obtain estimates for a set of PRESENT approximation.

While it might be possible to avoid the other issues discussed in this section, if we abandon the long-key assumption, parameter estimation seems like a challenge that is difficult to avoid. Indeed, the model we propose in Section 5 in some sense trades assumptions for increased computational effort. As such, efficient algorithms for computation of the signal trails seems like an increasingly important research topic. In connection to this, note that while an estimate of the parameters of the correlation distributions can be obtained by the above method, we are not aware of any such method to estimate the parameters of the multidimensional probability distributions.

5 Multivariate Linear Cryptanalysis

As argued in Section 4.1, when a cipher uses round keys that exhibit some dependence between them, the joint distribution $\mathbf{C}^K$ of linear correlations for a set of linearly independent approximations can deviate from the joint normality we would expect from a long-key cipher. Indeed, it seems very difficult to describe the exact joint distribution in this case. On a lower level, the marginal distributions do not necessarily have identical variances, as was assumed in [9, 28]. Additionally, as discussed in Section 4.2, the current models for multidimensional linear cryptanalysis do not seem to fully use most of the approximations in the chosen subspace of $\mathbb{F}_2^n \times \mathbb{F}_2^n$, and so by using the multidimensional approach, the attacker has to consider approximations that only add noise. What is worse, it seems that we are not able to formulate a wrong-key hypothesis in the multidimensional case that fully agrees with Theorem 2. Thus, the need for a wrong-key model forces us to consider the case of multiple, linearly independent approximations. It is therefore our aim to create a more powerful model for this setting which: models the behaviour of any set of linearly independent approximations; does not assume statistical independence of approximations or round keys; does not assume the shape of the joint correlation distribution; and takes into account the unknown part of the linear hull.

In the following we propose *multivariate linear cryptanalysis*. In Section 5.1 we present the main right- and wrong-key hypotheses the model relies on. This model in some sense trades assumptions for computational effort during the off-line analysis. In Section 5.2 we incorporate the signal/noise decomposition of [17] into the model, similar to [8], in order to make the model practically usable. In Section 5.3 we describe the model as used in a key-recovery attack where the attacker does not have access to the full codebook.

5.1 The Main Model: Arbitrary Right-Key Distribution

The first part of our model is very general, and simply expresses the fact that the correlations of a set of M linear approximations follow *some* multivariate probability distribution. Consider the vector $\mathbf{C}^K$ containing the correlations of M linear approximations with linearly independent masks. We propose the following right-key and wrong-key models.

Model 5 (Right-key – Multiple). *Let (α_i, β_i), $i = 1, \dots, M$, be M different linear approximations of a block cipher with linearly independent masks, and let $\mathbf{C}^K = (C^K_{\alpha_1,\beta_1} \cdots C^K_{\alpha_M,\beta_M})^\top$ be a vector containing the linear correlations. Then $\mathbf{C}^K \sim \mathcal{D}_M$ over the key space, for some M-variate probability distribution $\mathcal{D}_M$.*

Hypothesis 6 (Wrong-key – Multiple). *Let $\mathbf{\Sigma}^\delta = \mathrm{diag}(2^{-n})$. During a key recovery attack, for a wrong key guess, the linear correlation vector $\mathbf{C}^K$ of M linear approximations with linearly independent masks is a random vector with distribution $\mathbf{C}^K \sim \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^\delta)$.*

The wrong-key hypothesis is inspired by Theorem 2 and the result of [22], and the veracity of the hypothesis therefore relies on the assumption of pair-wise statistical independence of linear approximations of a random permutation. We take some steps towards validating Hypothesis 6 in Section B. For the right-key, this model allows the attacker to pick any set of linearly independent approximations, but requires that she can somehow estimate the shape of the distribution $\mathcal{D}_M$. While this at first does not seem very useful, as determining this distribution seems like a very hard problem in general, we propose a way to do this in the following by applying the signal/noise decomposition. We note that, interestingly, Model 5 could be extended to any arbitrary set of approximations, but it is currently unknown how to express Hypothesis 6 in this setting. It is therefore a very interesting open problem to derive the distribution of linearly dependent approximations of an ideal cipher.

5.2 The Practical Model: Signal/Noise Decomposition

The model presented requires the cryptanalyst to somehow obtain the distribution $\mathcal{D}_M$ for the right-key distribution. In most cases, we will be unable to calculate the exact distribution of $C^K_{\alpha,\beta}$ for any single approximation, and we therefore have to estimate $\mathcal{D}_M$. In order to do this, we take a similar approach to [8, 17]. Let $\mathcal{S}$ be the set of known *signal trails* for an approximation (α, β). Then we define the signal correlation as

$$C^{K\star}_{\alpha,\beta} = \sum_{U \in \mathcal{S}} C^K_U . \tag{4}$$

The signal correlation $C^{K\star}_{\alpha,\beta}$ will itself follow some probability distribution – we denote this by $\mathcal{D}^\star_{\alpha,\beta}$. We then assume that the unknown trails, the *noise*, behave as for a random permutation, i.e. their correlation is distributed as $\mathcal{N}(0, 2^{-n})$. Then we can approximate the full correlation with the distribution

$$C^K_{\alpha,\beta} \sim \mathcal{D}^\star_{\alpha,\beta} + \mathcal{N}(0, 2^{-n})$$

However, we still have the problem that $\mathcal{D}^\star_{\alpha,\beta}$ is unknown. This problem can be solved computationally. By computing Equation 4 for a large number of keys, we obtain a set of values drawn from $\mathcal{D}^\star_{\alpha,\beta}$. Whenever we need to randomly sample from $\mathcal{D}^\star_{\alpha,\beta}$, as we will need to do to estimate the strength of an attack, we simply sample from this data set. The same can be done for multiple approximations by calculating the signal correlations simultaneously for all M approximations for a randomly chosen set of keys. In this way, we trade any assumptions on the shape of the distribution $\mathcal{D}^\star_M$ for a potentially large computational effort. However, this computational effort is only required during the off-line analysis, and so has no influence on the computational complexity of an attack.

Under the assumption that the noise behaves as for a random permutation, the noise of linearly independent approximations will also be statistically independent, by Theorem 2 and Section B. Then we can make the following generalisation of the signal/noise decomposition to several approximations. Note that compared to [8], we here consider the distribution of the signal over the keys, as opposed to only the ELP of the approximations.

Model 6. *Let* $\mathbf{\Sigma}^\delta = \operatorname{diag}(2^{-n})$. *If the distribution,* $\mathcal{D}^\star_M$, *of the signal* $\mathbf{C}^{K\star}$ *is known, then the distribution of* $\mathbf{C}^K$ *in Model 5 is closely approximated by* $\mathbf{C}^K \sim \mathcal{D}^\star_M + \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^\delta)$.

Experimental Verification.

In order to verify Model 6, we again consider the set of 18 approximation over 9 rounds of 32-bit SmallPresent defined in Section 4.1. We considered the version with the 40-bit key-schedule, and enumerated part of the hull of each approximation, by using an approach very similar to the sparse correlation matrix method in [1]. In this way, we obtain a set of signal trails that includes all trails having intermediate masks with hamming weight at most four in each round. We did this simultaneously for all 18 approximations and 500 000 randomly chosen keys, in order to get an estimate of the distribution $\mathcal{D}^\star_{18}$. Furthermore, we measured the actual correlation values of the cipher for 2 000 000 randomly chosen keys. We then applied Model 6

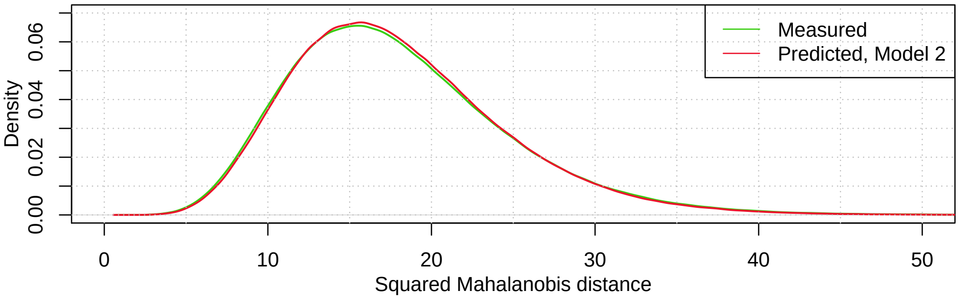


Figure 2: A density of the squared Mahalanobis distance for the joint distribution of linear correlation for 18 approximations over 9 rounds of 32-bit SmallPresent using a 40-bit key-schedule. The plot compares the density measured using the full codebook to a prediction made using Model 6.

to our signal estimate, and calculated the squared Mahalanobis distance of the two resulting data sets. The result is shown in Figure 2. The figure shows that Model 6 gives us a very close estimate of the actual distribution.

5.3 The Attack Model: Dealing with Undersampling

Even though Model 6 provides a way to get a good estimate of the multivariate correlation distribution, we would often like to avoid using the full codebook in a key-recovery attack. Thus, we also need to be able to express the distribution of the undersampled correlation, $\mathbf{C}^{K,N}$. Using a result due to Murphy, we develop such a model next.

Murphy showed [37] that the joint distribution over the text space of the empirical correlations, measured using N randomly drawn text pairs for a *fixed* key K_0, has a multivariate normal distribution, $\mathbf{C}^{K_0,N} \sim \mathcal{N}_M(\boldsymbol{\mu}^{K_0}, \boldsymbol{\Sigma}^{K_0,N})$, where $\boldsymbol{\mu}_i^{K_0} = C_{\alpha_i,\beta_i}^{K_0}$ and

$$\boldsymbol{\Sigma}_{i,j}^{K_0,N} = \begin{cases} N^{-1} C_{\alpha_i \oplus \alpha_j, \beta_i \oplus \beta_j}^{K_0} & \text{for } i \neq j, \\ N^{-1} & \text{for } i = j. \end{cases}$$

When taken as a random variable over the key space, we note that $\boldsymbol{\mu}^{K_0} = \mathbf{C}^K$ and therefore has distribution $\mathcal{D}_M$. Indeed, $\boldsymbol{\Sigma}^{K_0,N}$ also has a distribution over the key space, making the distribution over both the key and text space extremely difficult to analyse. However, as Murphy notes, it is often the case that the combined approximations $(\alpha_i \oplus \alpha_j, \beta_i \oplus \beta_j)$ are extremely weak, e.g. in the case where (α_i, β_i) and (α_j, β_j) activate different S-boxes at the input and output. In this case, $N^{-1} C_{\alpha_i \oplus \alpha_j, \beta_i \oplus \beta_j}^{K_0} \ll N^{-1}$, and we can set these covariances to zero. As Murphy says, in this case the fixed-key correlations are “approximately statistically independent” over the text space, in the sense that any contribution by the covariances is negligible. Under this assumption, we obtain the following theorem.

Theorem 3. *Let* $\mathbf{\Sigma}^N = \mathrm{diag}(N^{-1})$*. Consider a set of* M *approximations as given in Model 5. Assume that the correlation of any combination of two such approximations is zero. Then the empirical correlation vector of these approximations, measured with* N *randomly drawn plaintext-ciphertext pairs, has distribution* $\mathbf{C}^{K,N} \sim \mathcal{D}_M + \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^N)$*. For the wrong-key scenario of Hypothesis 6,* $\mathbf{C}^{K,N} \sim \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^\delta + \mathbf{\Sigma}^N)$*.*

Proof. From [37] we have that $\mathbf{C}^{K_0,N} \sim \mathcal{N}_M(\boldsymbol{\mu}^{K_0}, \mathbf{\Sigma}^{K_0,N})$ for a fixed key K_0. By assumption, we further have that $\mathbf{\Sigma}^{K_0,N} = \mathbf{\Sigma}^N = \mathrm{diag}(N^{-1})$, and so is independent of the key. The distribution of $\mathbf{C}^{K_0,N}$ over keys is therefore $\mathcal{N}_M(\mathcal{D}_M, \mathbf{\Sigma}^N) = \mathcal{D}_M + \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^N)$. For the wrong-key, $\mathcal{D}_M = \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^\delta)$, finishing the proof.

By applying Model 6 to this theorem, we obtain the following corollary.

Corollary 4. *For a set of* M *approximations as in Theorem 3, if the distribution,* $\mathcal{D}^\star_M$*, of the signal* $\mathbf{C}^{K\star}$ *is known, then the distribution of* $\mathbf{C}^{K,N}$ *is closely approximated by* $\mathbf{C}^{K,N} \sim \mathcal{D}^\star_M + \mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^N + \mathbf{\Sigma}^\delta)$*.*

As an interesting observation, this result shows how the original model by Matsui, Equation 2, can misleadingly give accurate results when N is relatively small, as is the case for the attack on DES. In this case, and as long as $\mathcal{D}_M$ does not deviate too much from joint normal distribution, N^{-1} will dominate the variance terms of $\mathrm{Cov}(\mathcal{D}_M)$ and $\mathbf{\Sigma}^\delta$, making the key-variance undetectable. This also shows that conducting experiments for a low number of rounds with low data complexity can not necessarily confirm a model.

Corollary 4 gives us a way to estimate the distribution of the correlation vector over the keys for a set of linearly independent approximations. In contrast to Results 1, 2, 4 and 5, no assumptions about independence or the parameters of the involved distributions are required, and we do not assume independent round keys. This generality of course comes with a cost: the approximations have to be linearly independent (although we are not forced to consider weak approximations), and we have to estimate the distribution $\mathcal{D}^\star_M$. We have partially discussed the latter issue in Section 5.2, and we will discuss how we have done this for PRESENT in Section 6.

6 Multivariate Linear Attacks on PRESENT

Different methods for distinguishing when using many approximations have been proposed. The LLR method was proposed by Baignères et al. in [3] as an optimal distinguisher and used in [26] in a multidimensional attack against the block cipher Serpent. Both the LLR method and the χ^2 method were studied in [25], where the LLR method was concluded to have better performance. However, as noted by Cho in [18], the LLR method is often not practical to use, as it requires an accurate knowledge of the key-dependent behaviour of the multidimensional probability distribution. For this reason, the χ^2 method is more commonly used. We now present a new attack on 26 and 27 rounds of PRESENT using this method and the improved multiple linear model of Section 5.

6.1 Determining the Advantage

The χ^2 method has been widely used as a distinguisher in various attacks. For this method, the test statistic is defined as

$$\mathcal{T}_{\chi^2} = N \sum_{i=1}^{M} (C^{K,N}_{\alpha_i,\beta_i})^2.$$

In the following, we describe how to determine the advantage of the χ^2 distinguisher using the theory developed in this paper. The approximations used are chosen based on the observations made by Ohkuma in [41]: the best approximations of PRESENT are those that start and end with the S-boxes S_i with $i \in \{5, 6, 7, 9, 10, 11, 13, 14, 15\}$. For our attack, we consider the input and output masks

$$\alpha = 2^{4i+3}, i \in \{5, 6, 7, 9, 10, 11, 13, 14, 15\},$$
$$\beta_1 = 2^{4i+3}, i \in \{5, 6, 7, 9, 10, 11, 13, 14, 15\}, \quad \beta_2 = 2^{4i+2}, i \in \{5, 6, 7, 9, 10, 11\}.$$

Taking all possible combinations of these input and output masks gives us $M = 135$ approximations. These approximations are chosen to facilitate efficient key-guessing over a large number of rounds, as will become evident in Section 6.2. We note that due to the structure of the approximations, it can be assumed that the undersampling matrix $\mathbf{\Sigma}^N$ is a diagonal matrix, as discussed in Section 5.3. This does not imply independence of the approximations, but simplifies our analysis considerably.

With this choice, we obtain the advantage in the following way. By using a signal that includes all trails having intermediate masks with hamming weight at most four in each round, and a technique similar to that of [1], we obtain a data set of observations from the signal distribution $\mathcal{D}^{\star}_{135}$. We used 217 100 random master keys to generate these observations. We now simulate observations from $\mathbf{C}^{K,N}$ in the following way: We fix a sample size for the simulation, say k. For the right key, we randomly sample k observations of $\mathcal{D}^{\star}_{135}$ (with replacement, if k is larger than the number of observations we have collected) from our data set. We then sample k random observations from the normal distribution $\mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^N + \mathbf{\Sigma}^{\delta})$. These two samples are then added together, following Corollary 4. The wrong-key distribution is simulated by randomly sampling k times from the normal distribution $\mathcal{N}_M(\mathbf{0}, \mathbf{\Sigma}^N + \mathbf{\Sigma}^{\delta})$, according to Theorem 3.

We note for comparison to previous works that the expected right-key capacity obtained from these simulations is $\mathrm{E}(\mathcal{C}^K) = 2^{-55.01}$ with a variance of $\mathrm{Var}(\mathcal{C}^K) = 2^{-115.59}$, whereas the wrong-key capacity has $\mathrm{E}(\mathcal{C}^K) = 2^{-56.92}$ and $\mathrm{Var}(\mathcal{C}^K) = 2^{-119.92}$.

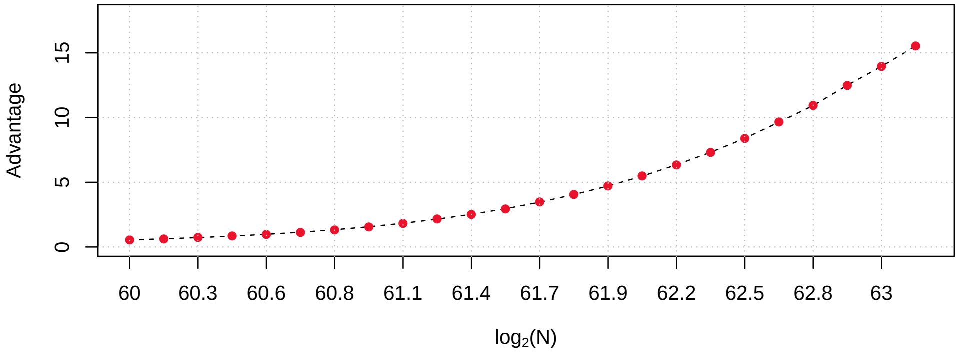


Figure 3: Advantage of the χ^2 distinguisher using 135 approximations of 22-round PRESENT, with $P_S = 0.95$. At half the codebook, $N = 2^{63}$, the advantage is 14.5 bits.

We can now calculate the empirical CDFs of the simulated right-key and wrong-key distributions. For a fixed success probability P_S, we use the right-key CDF to obtain a threshold τ, as described in Section 2.2. The advantage is finally calculated using the wrong-key CDF and τ, as defined in Section 2.2. Figure 3 shows the result of applying this procedure for $k = 2^{22}$, $P_S = 0.95$, and varying values of N. We note that we need to set k fairly high to obtain sufficient resolution of the empirical CDFs. For the chosen k, we can detect probabilities down to 2^{-22}, allowing us in turn to detect advantages of up to 22 bits. At half the codebook, $N = 2^{63}$ we obtain an advantage of 14.5 bits.

6.2 Attacking 26 rounds

Under the wrong key randomisation hypothesis, Hypothesis 6, we can turn our multivariate linear distinguisher into a key-recovery attack, as described in Section 2.2. That is, the attack proceeds as follows: Collect N plaintext-ciphertext pairs. Guess the bits of the outer round keys required to (partially) encrypt/decrypt the desired number of rounds. Apply the χ^2 distinguisher to the resulting correlations, and save the key guess if the distinguisher indicates a non-ideal cipher. Repeat for all guesses of the round key bits. For each saved key we can find the master key by exhaustively guessing the remaining bits and verifying by trial encryption.

We aim to recover the master key for r rounds of PRESENT-80 by using a multiple linear approximation over $r - 4$ rounds. Because of the large number of outer rounds we need to bypass, the approximations are chosen such that the involved round key bits are sparse. We consider the set of 135 approximations described above. The bit positions of the input and output masks are highlighted in Figure 4. Figure 4 shows the S-box positions we need to encrypt/decrypt to calculate the linear correlations of these approximations. The straightforward approach to partially

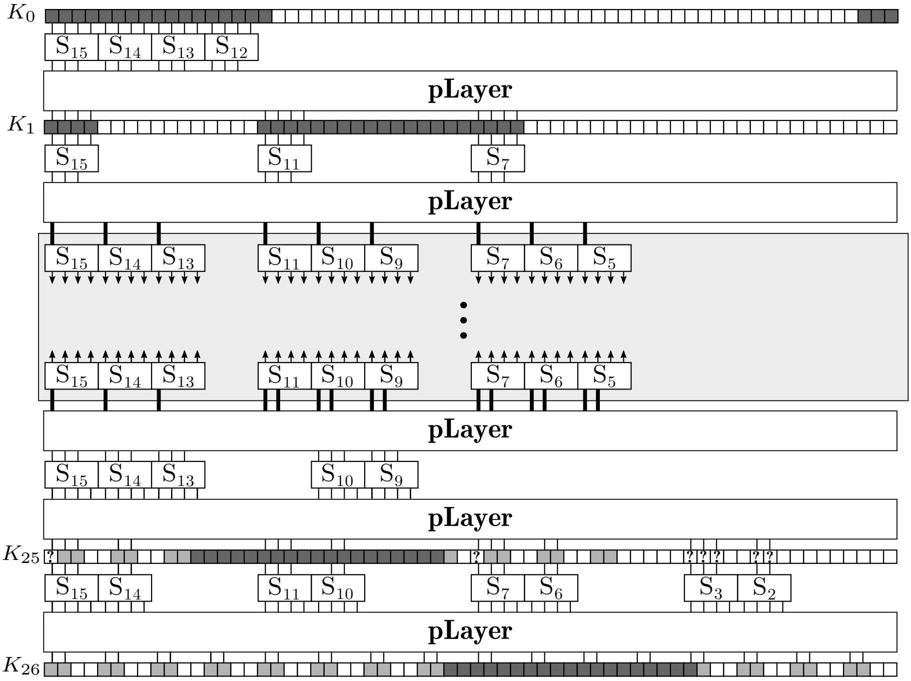


Figure 4: An outline of the 26-round attack using 22 round approximations. The input/output mask bits are indicated by bold lines. The dark grey squares indicate the round key bits obtained by guessing 24 bits of the master key. The light grey squares indicate the round key bits obtained by guessing 23 bits of the last round key. The squares indicated by **?** are extra bits of the second to last round key that need to be guessed.

encrypting/decrypting these positions would require guessing 80 key bits across the four round keys. By considering the key-schedule, we can dramatically improve this. We first guess the following 24 bits of the master key:

$$k_i, i \in [0,2] \cup [15,18] \cup [63,79]. \tag{5}$$

The round key bits we obtain from this guess are marked in dark grey in Figure 4, as well as 42 additional bits needed by the attack. By guessing the missing 23 bits of K_{26}, we also obtain 13 bits of K_{25}. Finally, we only need to guess an additional 7 bits of K_{25}. In total, we need to guess 54 key bits. Note that each approximation only depends on 4 bits of K_{25} and 16 bits of K_{26}. With these considerations in mind, the attack proceeds as follows.

Distillation phase

1. Obtain N partial text pairs (p_i, c_i), where p_i is 16 bits and c_i is 32 bits.

2. Generate a vector $\mathbf{t}$ of size 2^{48} where $\mathbf{t}[s\|t] = \#\{i \mid p_i = s \text{ and } c_i = t\}$.

Analysis phase

1. For each 24-bit guess of the partial master key, K_M, perform these steps:
    a) For each input mask α, calculate two vectors $\mathbf{t}_{\alpha_1}^{K_M}$ and $\mathbf{t}_{\alpha_2}^{K_M}$ of size 2^{16}, where

    $$\mathbf{t}_{\alpha_x}^{K_M}[j] = \#\{(p_i, c_i) | G_x(c_i) = j \text{ and } \alpha \cdot \mathcal{E}_{K_M}(p_i) = 0\},$$

    where $\mathcal{E}_{K_M}(p)$ is the partial two-round encryption of p under key K_M, and G_x selects the bits of c_i required to calculate the output masks of β_x, $x \in \{1, 2\}$.

    b) For each output mask β, fix a guess of the relevant 4 bits of K_{25}. Denote the guess K_I. Then calculate the $2^{16} \times 2^{16}$ matrix $\mathbf{A}_\beta^{K_I}$, where

    $$\mathbf{A}_\beta^{K_I}[i, j] = \beta \cdot \mathcal{D}_{K_I}(i \oplus j),$$

    and $\mathcal{D}_{K_I}(c)$ is the partial two-round decryption of the 16-bit value c using K_I, but *excluding* the first key XOR.

    c) Calculate the correlations of all 135 approximations and 2^{16} guesses of the partial K_{26} by calculating the matrix-vector products

    $$\mathbf{C}_{\alpha,\beta} = \tfrac{2}{N} \mathbf{A}_\beta^{K_I} \mathbf{t}_{\alpha_x}^{K_M} - 1.$$

    d) Repeat steps (b) and (c) for all values of K_I, resulting in correlation values for all approximations for at most 2^{36} guesses of the last two round keys.

    e) Extract the correlations of at most 2^{30} guesses that agree with K_M.

    f) Calculate the χ^2 test statistic $\mathcal{T}_{\chi^2}$ for each surviving key guess. Save all keys (of 54 bits) with $\tau < \mathcal{T}_{\chi^2}$.

Search phase

1. For each key candidate, perform trial encryption to find the remaining $80 - 54 = 26$ bits of the master key.

Attack Complexity.

We now consider the computational complexity of the attack. We consider the number of single round encryption equivalent operations performed.

- The distillation phase requires N operations.

- For the analysis phase:

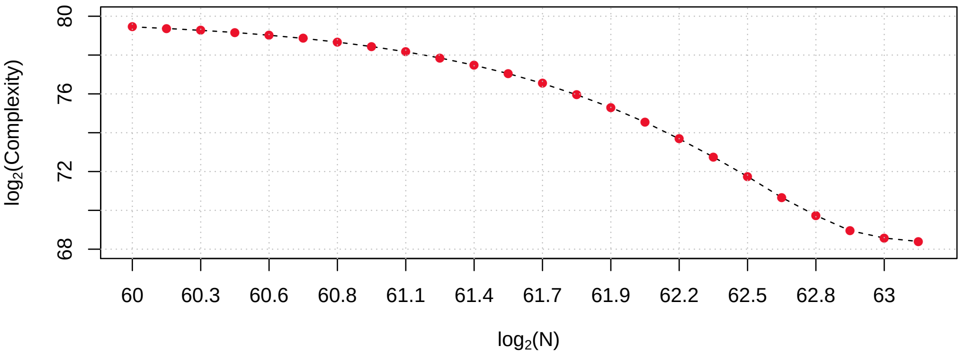


Figure 5: *Our 26 round attack:* Computational complexity as a function of data complexity for the 26-round attack on PRESENT using 135 approximations over 22 rounds. Non-distinct random texts were used, and $P_S = 0.95$. Note that the complexity reaches a lower limit close to $N = 2^{63}$ when the advantage becomes sufficiently large.

- Step 1a can be done by iterating over $\mathbf{t}$ once and encrypting two rounds, using $2 \cdot 2^{48}$ operations.
- Steps 1b and 1c can be performed using the FFT technique given in [19]. Using this technique, we only need to compute the first column of each $\mathbf{A}_\beta^{K_I}$, at a cost of $2 \cdot 2^{16}$ operations, and then calculate $\mathbf{C}_{\alpha,\beta}$ for a fixed β and all α in time $(2 \cdot 9 + 1) \cdot 16 \cdot 2^{16}$.
- There are 2^4 values of K_I and 15 output masks. Thus, steps 1d needs a total of $15 \cdot 2^4 \cdot (2 \cdot 2^{16} + (2 \cdot 9 + 1) \cdot 16 \cdot 2^{16}) \approx 2^{32.16}$ operations.
- Step 1e uses 2^{30} operations.
- Step 1f takes roughly $2 \cdot 135 \cdot 2^{30} = 2^{38.08}$ operations.
- In total, this phase uses $2^{24} \cdot (2^{49} + 2^{32.16} + 2^{30} + 2^{38.08}) \approx 2^{73.00}$ operations.

- Finally, the search phase requires $2^{\kappa-54}$ full encryptions of 2^{54-a} candidate keys, using a total of $26 \cdot 2^{\kappa-a}$ operations.

From Figure 3, we obtain a plot of the computational complexity of the 26-round attack, given in Figure 5. Here, we have fixed the success probability at 95%. As long as the search phase dominates, we can increase the number of texts to decrease to computational complexity. We can highlight two 26 round attacks with different trade-offs. For $N = 2^{63.0}$, the advantage is 14.0 bits, and the computational complexity is $2^{73.27}/26 = 2^{68.57}$ encryptions. Interestingly, this multiple attack uses far fewer approximations than Cho's multidimensional attack [18], but at half the data complexity and a computational complexity that is 11 times smaller, all the while needing far fewer assumptions. Compared to the reevaluation of Cho's attack

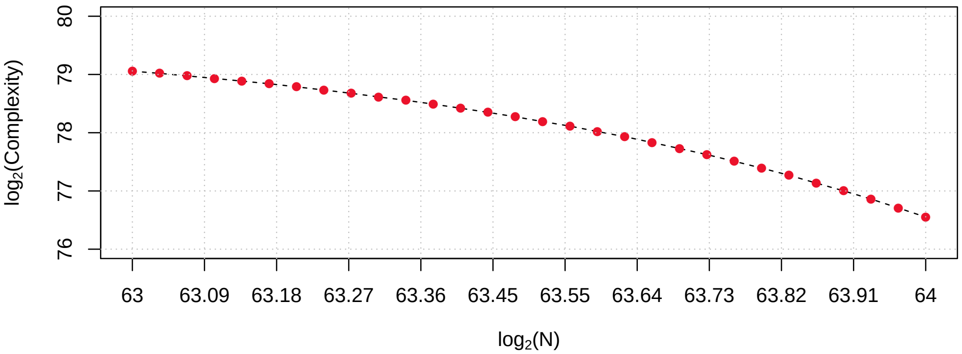


Figure 6: *Our 27 round attack:* Computational complexity as a function of data complexity for the 27-round attack on PRESENT using 135 approximations over 23 rounds. Distinct random texts were used, and $P_S = 0.95$.

in [8] (which has the same computational complexity as the original attack), our attack uses less data, and has a higher success probability. Alternatively, we can decrease the data complexity to $N = 2^{61.9}$, giving an advantage of 4.7 bits, and a computational complexity of $2^{80.00}/26 = 2^{75.30}$ encryptions. While being slower than Cho's attack, to the best of our knowledge, this attack has the lowest data complexity of any 26-round attack on PRESENT presented in the literature.

6.3 Attacking 27 rounds

The attack can be extended to 27 rounds by using the same approximations over 23 rounds. By guessing the bits of the master key given in Equation 5, we determine 41 required bits of the round keys. We then have to guess 25 bits of K_{27} and 6 bits of K_{26}, for a total of 55 bits of key material. Due to the way we carry out the attack, the complexity calculation is not affected by this – only the lower advantage has an influence. However, if we use non-distinct random texts for the attack, the advantage is too low. If we instead use distinct random texts, we obtain a better advantage. This scenario is in some sense a chosen plaintext attack, and has been studied in [9, 12]. The only change to our model is that $\mathbf{\Sigma}^N = \mathrm{diag}(\frac{2^n-N}{N\cdot(2^n-1)})$ in Corollary 4. The distribution of $\mathbf{C}^{K\star}$ was again estimated using 217 100 random keys as for the 26 round attack, and we obtain $\mathcal{E}(\mathcal{C}^K) = 2^{-56.38}$ and $\mathrm{Var}(\mathcal{C}^K) = 2^{-118.73}$ for the right-key. The resulting attack complexities are shown in Figure 6. Using the χ^2 distinguisher with $P_S = 0.95$ and $N = 2^{63.83}$, we obtain an advantage of 2.73 bits and a computational complexity of $2^{77.27}$ encryptions.

References

[1] Mohamed Ahmed Abdelraheem. “Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers”. In: *Information Security and Cryptology - ICISC 2012*. 2012, pp. 368–382.

[2] Mohamed Ahmed Abdelraheem, Martin Ågren, Peter Beelen, and Gregor Leander. “On the Distribution of Linear Biases: Three Instructive Examples”. In: *Advances in Cryptology - CRYPTO 2012*. 2012, pp. 50–67.

[3] Thomas Baignères, Pascal Junod, and Serge Vaudenay. “How Far Can We Go Beyond Linear Cryptanalysis?” In: *Advances in Cryptology - ASIACRYPT 2004*. 2004, pp. 432–450.

[4] Eli Biham. “On Matsui’s Linear Cryptanalysis”. In: *Advances in Cryptology - EUROCRYPT ’94*. 1994, pp. 341–355.

[5] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. “On Multiple Linear Approximations”. In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 1–22.

[6] Céline Blondeau, Asli Bay, and Serge Vaudenay. “Protecting Against Multidimensional Linear and Truncated Differential Cryptanalysis by Decorrelation”. In: *Fast Software Encryption, FSE 2015*. 2015, pp. 73–91.

[7] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. “Differential-Linear Cryptanalysis Revisited”. In: *Fast Software Encryption, FSE 2014*. 2014, pp. 411–430.

[8] Céline Blondeau and Kaisa Nyberg. “Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis”. In: *IACR Transactions on Symmetric Cryptology* 2016.2 (2016), pp. 162–191.

[9] Céline Blondeau and Kaisa Nyberg. “Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and Its Impact to Data Complexity”. In: *Design, Codes and Cryptography* 82.1-2 (2017), pp. 319–349.

[10] Céline Blondeau and Kaisa Nyberg. “Links between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities”. In: *Advances in Cryptology - EUROCRYPT 2014*. 2014, pp. 165–182.

[11] Céline Blondeau and Kaisa Nyberg. “New Links between Differential and Linear Cryptanalysis”. In: *Advances in Cryptology - EUROCRYPT 2013*. 2013, pp. 388–404.

[12] Céline Blondeau and Kaisa Nyberg. “On Distinct Known Plaintext Attacks”. In: *The Ninth International Workshop on Coding and Cryptography*. 2015.

[13] Andrey Bogdanov, Christina Boura, Vincent Rijmen, Meiqin Wang, Long Wen, and Jingyuan Zhao. “Key Difference Invariant Bias in Block Ciphers”. In: *Advances in Cryptology - ASIACRYPT 2013*. 2013, pp. 357–376.

[14] Andrey Bogdanov, Huizheng Geng, Meiqin Wang, Long Wen, and Baudoin Collard. “Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA”. In: *Selected Areas in Cryptography, SAC 2013*. 2013, pp. 306–323.

[15] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. “PRESENT: An Ultra-Lightweight Block Cipher”. In: *Cryptographic Hardware and Embedded Systems - CHES 2007*. 2007, pp. 450–466.

[16] Andrey Bogdanov and Vincent Rijmen. “Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers”. In: *Designs, Codes and Cryptography* 70.3 (2014), pp. 369–383.

[17] Andrey Bogdanov and Elmar Tischhauser. “On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui’s Algorithm 2”. In: *Fast Software Encryption, FSE 2013*. 2013, pp. 19–38.

[18] Joo Yeon Cho. “Linear Cryptanalysis of Reduced-Round PRESENT”. In: *Topics in Cryptology - CT-RSA 2010*. 2010, pp. 302–317.

[19] Baudoin Collard, François-Xavier Standaert, and Jean-Jacques Quisquater. “Improving the Time Complexity of Matsui’s Linear Cryptanalysis”. In: *Information Security and Cryptology, ICISC 2007*. 2007, pp. 77–88.

[20] Joan Daemen. “Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis”. PhD thesis. Doctoral Dissertation, March 1995, KU Leuven, 1995.

[21] Joan Daemen and Vincent Rijmen. “Probability distributions of Correlation and Differentials in Block Ciphers”. In: *IACR Cryptology ePrint Archive* 2005 (2005), p. 212.

[22] Joan Daemen and Vincent Rijmen. “Probability Distributions of Correlation and Differentials in Block Ciphers”. In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 221–242.

[23] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2.

[24] Carlo Harpes, Gerhard G. Kramer, and James L. Massey. “A Generalization of Linear Cryptanalysis and the Applicability of Matsui’s Piling-Up Lemma”. In: *Advances in Cryptology - EUROCRYPT ’95*. 1995, pp. 24–38.

[25] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. “Multidimensional Extension of Matsui’s Algorithm 2”. In: *Fast Software Encryption, 16th International Workshop, FSE 2009*. 2009, pp. 209–227.

[26] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. “Multidimensional Linear Cryptanalysis of Reduced Round Serpent”. In: *Information Security and Privacy, ACISP 2008*. 2008, pp. 203–215.

[27] Miia Hermelin and Kaisa Nyberg. “Multidimensional Linear Distinguishing Attacks and Boolean Functions”. In: *Cryptography and Communications* 4.1 (2012), pp. 47–64.

[28] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. “Capacity and Data Complexity in Multidimensional Linear Attack”. In: *Advances in Cryptology - CRYPTO 2015.* 2015, pp. 141–160.

[29] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. “Linear Cryptanalysis Using Multiple Approximations”. In: *Advances in Cryptology - CRYPTO ’94.* 1994, pp. 26–39.

[30] Pascal Junod. “On the Complexity of Matsui’s Attack”. In: *Selected Areas in Cryptography, SAC 2001.* 2001, pp. 199–211.

[31] Susan K. Langford and Martin E. Hellman. “Differential-Linear Cryptanalysis”. In: *Advances in Cryptology - CRYPTO ’94.* 1994, pp. 17–25.

[32] Gregor Leander. “On Linear Hulls, Statistical Saturation Attacks, PRESENT and a Cryptanalysis of PUFFIN”. In: *Advances in Cryptology - EUROCRYPT 2011.* 2011, pp. 303–322.

[33] Gregor Leander. “Small Scale Variants Of The Block Cipher PRESENT”. In: *IACR Cryptology ePrint Archive* 2010 (2010), p. 143.

[34] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology - EUROCRYPT ’93.* 1993, pp. 386–397.

[35] Mitsuru Matsui. “On Correlation Between the Order of S-boxes and the Strength of DES”. In: *Advances in Cryptology - EUROCRYPT ’94.* 1994, pp. 366–375.

[36] Mitsuru Matsui. “The First Experimental Cryptanalysis of the Data Encryption Standard”. In: *Advances in Cryptology - CRYPTO ’94.* 1994, pp. 1–11.

[37] S. Murphy. “The Independence of Linear Approximations in Symmetric Cryptanalysis”. In: *IEEE Trans. Information Theory* 52.12 (2006), pp. 5510–5518.

[38] Phuong Ha Nguyen, Hongjun Wu, and Huaxiong Wang. “Improving the Algorithm 2 in Multidimensional Linear Cryptanalysis”. In: *Information Security and Privacy, ACISP 2011.* 2011, pp. 61–74.

[39] Kaisa Nyberg. “Linear Approximation of Block Ciphers”. In: *Advances in Cryptology - EUROCRYPT ’94.* 1994, pp. 439–444.

[40] Kaisa Nyberg. “Statistical and Linear Independence of Binary Random Variables”. In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 432.

[41] Kenji Ohkuma. “Weak Keys of Reduced-Round PRESENT for Linear Cryptanalysis”. In: *Selected Areas in Cryptography, SAC 2009.* 2009, pp. 249–265.

[42] Ali Aydin Selçuk. "On Probability of Success in Linear and Differential Cryptanalysis". In: *Journal of Cryptology* 21.1 (2008), pp. 131–147.

[43] Ali Aydin Selçuk and Ali Biçak. "On Probability of Success in Linear and Differential Cryptanalysis". In: *Security in Communication Networks, SCN 2002*. 2002, pp. 174–185.

[44] Bing Sun, Meicheng Liu, Jian Guo, Vincent Rijmen, and Ruilin Li. "Provable Security Evaluation of Structures Against Impossible Differential and Zero Correlation Linear Cryptanalysis". In: *Advances in Cryptology - EUROCRYPT 2016*. 2016, pp. 196–213.

[45] Hong Xu, Ping Jia, Geshi Huang, and Xuejia Lai. "Multidimensional Zero-Correlation Linear Cryptanalysis on 23-Round LBlock-s". In: *Information and Communications Security, ICICS 2015*. 2015, pp. 97–108.

[46] Jingyuan Zhao, Meiqin Wang, and Long Wen. "Improved Linear Cryptanalysis of CAST-256". In: *Journal of Computer Science and Technology* 29.6 (2014), pp. 1134–1139.

[47] Lei Zheng and Shao-Wu Zhang. "FFT-Based Multidimensional Linear Attack on PRESENT Using the 2-Bit-Fixed Characteristic". In: *Security and Communication Networks* 8.18 (2015), pp. 3535–3545.

A 40-bit Key-Schedule for SmallPresent

We define a 40-bit key-schedule for 32-bit SmallPresent, which is a scaled down version of the 80-bit PRESENT key-schedule. Let $K = k_{39}k_{38}\dots k_1k_0$ be a 40-bit key register, initialised to the master key. At round i, the round key is extracted as the 32 most significant bits of K, i.e. $K_i = k_{39}k_{38}\dots k_9k_8$. Then, K is updated as follows:

- K is rotated 9 bits to the right,
- The PRESENT S-box is applied to $k_{39}k_{38}k_{37}k_{36}$,
- A round counter is xor'ed to the least significant bits.

The round counter starts at 1 and is incremented by 1 for each round.

B Pair-Wise Independence of Linear Correlations

The wrong-key hypothesis presented here, Hypothesis 6, follows from Theorem 2 and [22], assuming that linear approximations of random permutations can be considered pair-wise independent. While it seems difficult to show when this assumption is true, we here take some steps towards verifying Hypothesis 6 experimentally. We first note that the normality of the marginal distributions of $\mathbf{C}^K$ for a random permutation is

Table 2: Results of the Pearson χ^2 test of independence for various permutation sizes. A p-value larger than 0.05 indicates that the correlations of two linear approximations are statistically independent at the 95% significance level.

Size	**Experiments**	**% of Experiments with p-value > 0.05**	**Smallest observed p-value**
2^{16}	20000	99.995	0.021
2^{20}	20160	100.00	0.975
2^{24}	15342	100.00	1.000

proven in [22]. Moreover, it seems unlikely that the joint distribution would deviate much from a multivariate normal distribution for most sets of approximations. Thus, if we can demonstrate that pairs of correlation distributions are independent, we can be confident that Hypothesis 6 is reasonable. To this end, we performed the following experiment:

- Fix a size of the permutation, say 2^n,
- Pick two random linear approximations,
- Generate 10 000 random permutations of the given size and measure the exact correlation of both approximations for each permutation using the full codebooks,
- Perform Pearson's χ^2 test of independence between the two correlation distributions and record the p-value,
- Repeat the above process the desired number of times.

We note that when performing Pearson's χ^2 test of independence, the null hypothesis is that the two observed distributions are independent, and thus a p-value larger than e.g. 0.05 would indicate independence at the 95% significance level.

We performed the above experiment for varying sizes of the permutations, and the results are shown in Table 2. Here, we observe that for a 16-bit permutation, one out of 20 000 pairs of permutations had a significant p-value of 0.021. However, already for the slightly larger 20-bit permutation, the lowest p-value was 0.975; in other words, even in the worst case, there was only 2.5% chance that the two correlation distributions were dependent. For a 24-bit permutation, this results are even clearer, with the lowest p-value being extremely close to 1.

Additionally, we repeated the experiments for the 20-bit permutations, but this time using approximations that only differed in a single bit. Even for these very similar approximations, we observed the exact same results as for randomly chosen pairs of approximations. In light of these experimental results, it thus seems quite reasonable that correlations of 64- or 128-bit permutations would be independent for all practical intents and purposes.

Publication 3

Generating Graphs Packed with Paths

Publication Information

Mathias Hall-Andersen and Philip S. Vejre. “Generating Graphs Packed with Paths: Estimation of Linear Approximations and Differentials”. In: *IACR Transactions on Symmetric Cryptology* 2018.3 (2018), pp. 265–289

Contribution

- Main author.

Remarks

This publication has been slightly edited to fit the format.

Generating Graphs Packed with Paths
Estimation of Linear Approximations and Differentials

Mathias Hall-Andersen[1] and Philip S. Vejre[2]

[1] University of Copenhagen, Denmark,
[2] Technical University of Denmark

Abstract. When designing a new symmetric-key primitive, the designer must show resistance to known attacks. Perhaps most prominent amongst these are linear and differential cryptanalysis. However, it is notoriously difficult to accurately demonstrate e.g. a block cipher's resistance to these attacks, and thus most designers resort to deriving bounds on the linear correlations and differential probabilities of their design. On the other side of the spectrum, the cryptanalyst is interested in accurately assessing the strength of a linear or differential attack.

While several tools have been developed to search for optimal linear and differential trails, e.g. MILP and SAT based methods, only few approaches specifically try to find as many trails of a single approximation or differential as possible. This can result in an overestimate of a cipher's resistance to linear and differential attacks, as was for example the case for PRESENT.

In this work, we present a new algorithm for linear and differential trail search. The algorithm represents the problem of estimating approximations and differentials as the problem of finding many long paths through a multistage graph. We demonstrate that this approach allows us to find a very large number of good trails for each approximation or differential. Moreover, we show how the algorithm can be used to efficiently estimate the key dependent correlation distribution of a linear approximation, facilitating advanced linear attacks. We apply the algorithm to 17 different ciphers, and present new and improved results on several of these.

1 Introduction

Whenever a new design for a symmetric-key primitive is proposed, it is usually accompanied by a design rationale. This rationale explains how the specific choice of components ensure resistance to a set of common attack techniques. However, thoroughly checking maybe a dozen different attacks is laborious work for the designer, and it is therefore common to somehow make an estimate of how well a design resists a specific attack.

Two attack techniques that are almost always featured in the security analysis of a new design, due to their long history and many strong results, are linear [35] and differential [10] cryptanalysis. However, it is notoriously difficult to make an accurate and complete analysis of a cipher's security against these attacks, and for this reason methods of estimating the strength of these attacks feature prominently in the initial analysis of a new design. For block ciphers, this will often consist of lower-bounding the number of active S-boxes in a linear or differential trail, thus showing how many rounds the cipher needs to resist these attacks.

Nevertheless, several examples exist of this approach not giving the full picture, in particular due to the existence of linear approximations or differentials that contain a very large number of good trails. This effect was already recognised for differential cryptanalysis in [34] and subsequently extended to linear cryptanalysis in [39] where it was dubbed the linear hull effect. As an example of this phenomenon, it was demonstrated in [40] that for the block cipher PRESENT the difference between a single linear trail and the full linear approximation is quite significant. Thus, it would be extremely helpful for a designer if a simple tool existed that could more accurately find linear approximations and differentials for a given design. This would not only save the designer time, but potentially also allow for exploration of a larger design space as well as enabling a more informed choice of the number of rounds needed to obtain adequate security.

1.1 Previous Work

Several approaches for finding linear and differential trails have been suggested in the literature. Perhaps the most well known technique is Matsui's original branch-and-bound algorithm [36], which can essentially be viewed as a depth-first search with pruning. While this algorithm does guarantee to return the optimal trail for any starting value, one still needs to have some idea what a good starting value might be. Moreover, while the algorithm can be adapted to return multiple trails, this is not very efficient if the number of trails is extremely large.

Several other approaches for finding linear and differential trails have been proposed. Amongst these are MILP based algorithms [28, 38, 43, 46] and SAT based algorithms [4, 33, 37], as well as more dedicated approaches [26, 27, 42]. Both the MILP and SAT based approaches can be extended in order to find multiple trails by removing already known trails from the solution space, but this approach also has the problem of scaling linearly with the number of trails. Additionally, in order to use these algorithms, every design one wishes to analyse has to be formulated as a MILP/SAT model.

A few approaches for finding large numbers of linear or differential trails have been suggested. Matsui's algorithm was generalised in [21, 22] to search for multiple differential trails of generalised Feistel networks. A more versatile approach was presented in [1], where the idea of using partial, sparse correlation/differential transitions matrices to find multiple trails was proposed. While this approach does scale well with the number of trails found, it potentially has high memory requirements. This problem was acknowledged in [2] where the matrix method was combined with the MILP method to improve results for ARX designs. Still, these works do not offer a general, design agnostic strategy for choosing the partial matrices.

While the mentioned works focus on estimating expected differential probabilities or expected squared correlations, we note that for linear cryptanalysis especially, there has recently been an increased focus on the key dependent distribution of correlations. Namely, several works developing models for the key dependent behaviour of correlations have been published [12, 13, 15, 31] as well as some advanced attack techniques exploiting these correlations distributions [16, 17]. Thus, it is of additional interest to develop algorithms that also allow for efficient estimation of these distributions.

1.2 Contributions

In this work, we propose a new algorithm for linear and differential trail search. The overall concept of the algorithm is to represent all linear/differential trails as paths in a multistage graph, and then find a manageable subgraph which hopefully contains good trails. By performing a breadth first traversal of this subgraph, we can very efficiently consider a larger number of trails when estimating the squared correlation/differential probability, and even do so for many linear approximations/differentials simultaneously. Moreover, for linear cryptanalysis, the algorithm allows us to very efficiently approximate the correlation distributions over the key space.

While the overall concept of this approach is related to the idea of partial correlation and difference transition matrices, the graph representation allows a designer or cryptanalyst to gain additional insight, e.g. one can extract the actual trails from the graph or visualise the trail structure in order to gain deeper understanding of a cipher's linear and differential behaviour (see e.g. Figure 5). Moreover, we can more easily obtain the key-dependent linear correlations without having the recompute everything for each new key. In more detail, we achieve the following:

- **Efficient graph generation** We present a heuristic approach for selecting a subgraph of the linear/differential trail graph, i.e. we identify good approximations/differentials over a single round. For SPN ciphers, we give a highly efficient algorithm for generating these. Moreover, we show how to remove redundant information from the graph in order to reduce memory costs. As opposed to the strategy for choosing partial correlation/difference matrices in [1], our heuristic is design agnostic.

- **Algorithm optimisations** We present a number of optimisations to the basic algorithm which both reduces the time it takes to generate the trail graph and the amount of memory consumed while generating the graph. The latter is done by removing single round approximations/differentials which are not part of any trail before it is ever added to the graph. While the most effective improvements only apply to SPN ciphers, they allow us to increase the effective size of our search space; as an example, for Midori64 [6] we were able to include as many as $2^{46.5}$ single round approximations in our search space.

- **Improved estimates of ELP and EDP** Compared to algorithms that find one trail at a time (e.g. MILP and SAT based methods), our graph representation allows us to consider a much larger number of trails when estimating the expected squared correlation or the expected differential probability. As an example, we are able to consider $2^{112.4}$ linear trails for a single approximation of PUFFIN [23]. This ensures a more accurate estimate of these statistics.

- **Extensive application** We use the new algorithm to find linear approximations and differentials for 17 different SPN ciphers. The selection of ciphers have block sizes ranging from 48 to 128 bits, use 4- and 8-bit S-boxes, and apply a variety of different design approaches for choosing the linear layer, e.g. from very lightweight bit permutations to heavy MDS matrices. We present new results on several ciphers, and improve existing results for five ciphers.

- **Correlation distributions** We demonstrate that for SPN ciphers, the graph representation can also be used to efficiently obtain estimates for the key dependent correlation distribution of a linear approximation. In particular, it takes at most a couple of minutes to generate key dependent correlation values for 10 000 randomly chosen keys. We use this fact to investigate the correlation distributions of several ciphers, and show for example that GIFT-64 [7] exhibits multiple approximations with asymmetries similar to those observed for DES in [17]. In general, this feature of our algorithm facilitates easier application of more advanced linear attacks.

- **Software implementation** Finally, we make our implementation of the algorithm freely available at https://gitlab.com/psve/cryptagraph. This implementation is written in Rust, and is highly optimised and parallelised. At the time of writing, it supports analysis of SPN ciphers whose substitution layer consists of applying S-boxes to the state, as well as Feistel ciphers with SPN-like F-functions. Additionally, adding new ciphers to the implementation only requires the usual implementation of the S-box and the linear layer, as opposed to MILP and SAT based tools that require modelling of the cipher in the relevant framework. We hope that the availability of this tool, as well as its ease of use, will facilitate more informed design processes and improved cryptanalysis.

The rest of this work is structured as follows: In Section 2 we introduce the basic definitions for linear and differential cryptanalysis. Section 3 introduces the idea of the graph framework, while Section 4 outlines the basic algorithm for trail search. Section 5 contains several improvements to the basic algorithm. In Section 6 and Section 7 we present the results we obtain by using the algorithm on various ciphers. Finally, Section 8 discusses some prospects for future work.

2 Preliminaries

Throughout the paper we consider *block ciphers*, i.e. functions of the type

$$\mathcal{E}(k, m) : \mathbb{F}_2^\kappa \times \mathbb{F}_2^n \to \mathbb{F}_2^n,$$

where $\mathcal{E}$ is a permutation on the plaintext space $\mathbb{F}_2^n$ for each key $k \in \mathbb{F}_2^\kappa$. In particular, we consider *iterative block ciphers* where $\mathcal{E}$ is defined as a composition of several (potentially different) round-functions, i.e.

$$\mathcal{E} = f_r \circ \ldots \circ f_1.$$

We define a *distinguisher* as an algorithm which attempts to distinguish between the function $\mathcal{E}$ and a permutation picked uniformly at random from the space of all permutations on $\mathbb{F}_2^n$. In particular, the cryptanalyst is interested in a distinguisher which succeeds with high probability and uses time less than 2^κ.

In the following, we briefly describe the main ideas of linear and differential distinguishers as well as the problem of finding good properties of these types. While we only describe the techniques from a distinguisher viewpoint, distinguishers of both types can be turned into key-recovery attacks in most cases.

2.1 Linear Cryptanalysis

We define a *linear approximation* of a block cipher as the pair of masks $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. Let $\langle \cdot, \cdot \rangle$ denote the canonical inner product on $\mathbb{F}_2^n$. We say that the approximation (α, β) has a *linear correlation* defined by

$$C_{(\alpha,\beta)}^k = 2 \cdot \Pr_{m \in \mathbb{F}_2^n}(\langle \alpha, m \rangle = \langle \beta, \mathcal{E}(k, m) \rangle) - 1.$$

Note that the correlation is key dependent, and thus has some distribution over $\mathbb{F}_2^\kappa$. For a randomly chosen permutation, the correlation is drawn from the distribution $\mathcal{N}(0, 2^{-n})$ [24]. Thus, if there exists a linear approximation (α, β) of a block cipher such that $C_{(\alpha,\beta)}^k$ is distributed significantly differently from $\mathcal{N}(0, 2^{-n})$, we can use this approximation to build a distinguisher.

2.2 Differential Cryptanalysis

We define a *differential* of a block cipher as the pair of differences $(\Delta, \nabla) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. Let $\oplus$ denote the componentwise addition of vectors in $\mathbb{F}_2^n$. Then we say that the differential (Δ, ∇) has a *differential probability* defined by

$$p_{(\Delta,\nabla)}^k = \Pr_{m \in \mathbb{F}_2^n}(\mathcal{E}(k, m) \oplus \mathcal{E}(k, m \oplus \Delta) = \nabla).$$

For a randomly chosen permutation, we expect the differential probability to be close to 2^{-n}. Thus, if there exists a differential (Δ, ∇) of a block cipher such that $p_{(\Delta,\nabla)}^k$ is significantly bigger than 2^{-n} for almost all keys, we can us this differential to build a distinguisher.

2.3 Finding Approximations and Differentials

Determining either $C^k_{(\alpha,\beta)}$ or $p^k_{(\Delta,\nabla)}$ is not feasible for the values of n and κ used in practice. Therefore, for iterative block ciphers, the problem is usually reduced to that of finding linear and differential *trails*. A linear trail of an approximation (α, β) is defined as a sequence of masks $U = (u_0, \ldots, u_r)$, with $(u_0, u_r) = (\alpha, \beta)$. Then, we define the *correlation contribution* of this trail as

$$C^k_U = \prod_{i=0}^{r} C_{(u_i,u_{i+1})}(i),$$

where $C_{(u_i,u_{i+1})}(i)$ is the correlation of the approximation (u_i, u_{i+1}) for the i'th round function f_i. Since the f_i usually have a simple form, it is easier to determine the correlation of these functions. The set of all trails of an approximation is called the *linear hull* of the approximation. It can be shown that the correlation of (α, β) is the sum of the correlation contributions of all trails in the linear hull [25], i.e.

$$C^k_{(\alpha,\beta)} = \sum_{(u_0,u_r)=(\alpha,\beta)} C^k_U.$$

The situation is analogous for differentials. Although the number of trails is extremely large, it often suffices to find a set of trails with high correlation or probability contribution, such that computing the partial sum over these trails is a good approximation of the actual correlation or probability. Thus, finding a good set of trails is essential to both linear and differential cryptanalysis, and it is this problem that we will consider in the following.

A note on ELP and EDP As explained above, the linear correlation and differential probability of a cipher depends on the specific key used. However, for the initial analysis of these attacks, it is often more convenient to consider the *expected linear potential* and the *expected differential probability*.

In the case of differentials, EDP is defined as $\mathrm{E}(p^k_{(\Delta,\nabla)})$ and it is often assumed that $p^k_{(\Delta,\nabla)} \approx \mathrm{E}(p^k_{(\Delta,\nabla)})$ for most keys. For approximations, ELP is defined as $\mathrm{E}((C^k_{(\alpha,\beta)})^2)$, and it can be shown that $\mathrm{E}((C^k_{(\alpha,\beta)})^2) \approx \sum (C^k_U)^2$, and that for key-alternating ciphers (or Feistel ciphers with SPN-like structures) $(C^k_U)^2$ is independent of the key [25].

Thus, considering ELP and EDP eliminates the key and therefore greatly simplifies the search, and usually gives a good indicator for the strength of an approximation or differential. We will therefore initially take this approach, and then show in Section 7 how to find the key-dependent linear correlation distributions.

3 Trail Search Viewed as a Graph Problem

Although finding trails of a specific approximation or differential is already a difficult problem, for a newly designed block cipher it might not even be clear what approximations or differentials we should be considering. In the following, we will view the problem of finding good approximations and differentials more abstractly as a graph problem. This perspective will help us develop a trail search algorithm which does not require any initial understanding of the linear or differential behaviour of the cipher being analysed. We will describe the graph problem and the algorithm in terms of linear cryptanalysis, but all observations are directly applicable to the case of differential cryptanalysis.

We first introduce some graph notation. A *directed graph* G is a set of *vertices* V and a set of *directed edges* E. We associate a value to each vertex. Throughout the paper, we will not differentiate between a vertex and its value, and use the two concepts interchangeably. We denote a directed edge from a vertex $u \in V$ to a vertex $v \in V$ by $u \to v$. For a weighted graph, each edge $u \to v$ has a *length*, denoted by $l(u \to v)$. We furthermore denote a *path* from a vertex u to a vertex v by $u \rightsquigarrow v$. If $v = v_1, \dots, u = v_k$ are the vertices traversed by this path, then we define the length of the path as:

$$l(u \rightsquigarrow v) = \prod_{i=1}^{k-1} l(v_i \to v_{i+1}).$$

Furthermore, we call the set of all paths $u \rightsquigarrow v$ the *hull of* (u, v). We denote the hull by $u \diamond v$ and associate to it a *weight* defined as:

$$w(u \diamond v) = \sum l(u \rightsquigarrow v),$$

i.e. the sum of the length of all the paths contained in the hull. We will exclusively consider a special type of directed graph, called a *multistage graph.*

Definition 1 (Multistage Graph)**.** Let G be a directed graph with vertices V and edges E. If the vertices in V are partitioned into ℓ subsets $S_0, \dots, S_{\ell-1}$, called *stages*, such that any edge in E has the form $u \to v$ with $u \in S_i$ and $v \in S_{i+1}$, for some $i \in [0, \ell - 1[$, we call the graph a multistage graph.

By definition, a multistage graph is a directed acyclic graph (DAG). We now define a weighted multistage graph $G_{\mathcal{E}}$ which represents the linear hulls of all approximations of an iterative block cipher $\mathcal{E}$. Assume that $\mathcal{E}$ has r rounds. Then $G_{\mathcal{E}}$ has $r + 1$ stages each with 2^n vertices representing the elements of $\mathbb{F}_2^n$. $G_{\mathcal{E}}$ contains all edges $u \to v$ for $u \in S_i$ and $v \in S_{i+1}$, $i \in [0, r[$. The length of an edge is defined as

$$l(u \to v) = (C_{(u,v)}(i))^2 \quad \text{if } u \in S_i.$$

Note that if $\alpha \in S_0$ and $\beta \in S_r$, then a path $(\alpha \rightsquigarrow \beta)$ is equivalent to a linear trail $U = (\alpha, \dots, \beta)$ and its length is exactly $(C_U^k)^2$. Moreover, $\alpha \diamond \beta$ corresponds exactly to the linear hull of the approximation (α, β) and its weight is equal to the ELP of (α, β). Thus, the graph $G_{\mathcal{E}}$ represents the linear hulls of all approximations of $\mathcal{E}$. Finding good approximations therefore corresponds to finding pairs of vertices $(\alpha, \beta) \in S_0 \times S_r$ such that $w(\alpha \diamond \beta)$ is as large as possible. In the following section, we describe an algorithm that aims to solve this problem.

4 A New Algorithm for Trail Search

The graph $G_{\mathcal{E}}$ defined above is exceedingly huge; it has $(r+1) \cdot 2^n$ vertices and $r \cdot 2^{2n}$ edges. Thus, it is completely infeasible to run even a linear time algorithm on the graph[3]. We therefore have to somehow reduce the size of the graph, i.e. we have to reduce the size of the *search space*. Straight away, we can remove any edges $u \to v$ with $l(u \to v) = 0$ as any path which includes this edge will have length zero and therefore not contribute to the hull. Nevertheless, for most ciphers the set of non-zero edges in $G_{\mathcal{E}}$ is still intractable. Thus, we propose the following approach:

1. Determine an interesting subgraph $\bar{G}_{\mathcal{E}}$ of $G_{\mathcal{E}}$.
2. Calculate $w(\alpha \diamond \beta)$ for all $(\alpha, \beta) \in S_0 \times S_r$ in $\bar{G}_{\mathcal{E}}$.

For this approach to give a good result, would like many of the longest paths of $\alpha \diamond \beta$ to appear in $\bar{G}_{\mathcal{E}}$. How to ensure this is clearly highly dependent on the cipher $\mathcal{E}$ in question. Moreover, at first glance it seems that if we can specify $\bar{G}_{\mathcal{E}}$, then we already know a good collection of trails. However, we note that finding good approximations in some sense corresponds to finding a minimal subgraph. In contrast, in the process of finding the subgraph $\bar{G}_{\mathcal{E}}$ we can start with a larger subgraph that might contain a lot of unnecessary vertices and edges. While this graph might be too large for us to perform Step 2 above, we can then remove any superfluous information and hopefully arrive at a suitable subgraph $\bar{G}_{\mathcal{E}}$.

In Section 4.1, we propose a simple, generic approach to Step 1. Section 4.2 then details how to efficiently perform Step 2 on the resulting subgraph. In Section 5 we propose various improvements to the naive algorithm.

4.1 Choosing a Subgraph

We propose the following general, design agnostic heuristic for generating $\bar{G}_{\mathcal{E}}$.

- *Selection:* Select the k longest edges going out from each stage in $G_{\mathcal{E}}$.
- *Pruning:* Remove any irrelevant edges and vertices from the resulting graph.

It is clear that this way of selecting edges does not guarantee that we find optimal paths. Indeed, it could be the case that the longest paths contain a single very short edge. However, as long as we are able to use fairly large values of k, we should be able to cover a good fraction of the search space. Additionally, if we *do* find paths using this strategy, we can at least be confident that they are quite close to optimal. A similar heuristic was used in [1, 2] for constructing partial correlation matrices – here, single round approximations with low hamming weight were selected. How well this heuristic works is however heavily dependent on the cipher's structure. Indeed, choosing the longest edges seem like an approach that will work well in a more general setting.

[3] Note that the longest path problem can be solved in linear time for DAGs.

We now show how the selection step can be performed efficiently for ciphers with SPN-like round-functions and then detail how the pruning step works.

Edge Selection for SPN Ciphers

For the sake of simplicity, we will initially consider *substitution-permutations networks* (SPN ciphers) with identical round-functions (aside from the key addition), i.e. $\forall i, f_i = f \oplus k_i$, although the approach also applies to the more general case of SPN ciphers with different round-functions. Our goal is then to find a set $\mathcal{A}$ of the k approximations (each representing an edge) with largest squared correlation. Following Section 2.3, we can ignore the key addition in the following, and hence the SPN round-function takes on the form:

$$f = \mathcal{L} \circ \mathcal{S},$$

where $\mathcal{L}$ is a linear transformation of the state and $\mathcal{S}$ is the parallel application of t independent S-boxes to the state. Let $\mathcal{S}_i$ be the i'th S-box, i.e.

$$\mathcal{S} = \mathcal{S}_0 \| \cdots \| \mathcal{S}_{t-1}.$$

Then, in the usual way, the correlation of an approximation (α, β) of f is entirely determined by the approximation $(\alpha, \mathcal{L}^{-1}(\beta))$ of $\mathcal{S}$. This is in turn entirely determined by the component approximations of the individual S-boxes so that

$$(C_{(\alpha,\beta)}(f))^2 = \prod_{i=0}^{t-1} (C_{(\alpha_i, \mathcal{L}^{-1}(\beta)_i)}(\mathcal{S}_i))^2.$$

We can now reduce the problem of finding the k best approximations over f to the problem of finding certain classes of approximations over $\mathcal{S}$. To this end, we introduce the notion of an *S-box pattern.*

Definition 2 (S-box pattern)**.** Let $\mathcal{S} = \mathcal{S}_0 \| \cdots \| \mathcal{S}_{t-1}$ be the parallel application of t independent S-boxes to a cipher state. Then a pattern of $\mathcal{S}$ is a tuple $p \in \mathbb{R}^t$. The pattern represents a set of approximations of $\mathcal{S}$ such that the squared correlation of $\mathcal{S}_i$ is equal to p_i. We associate a value to a pattern p, namely $C(p) = \prod p_i$, i.e. the squared correlation of any approximation represented by p.

We say that a pattern *expands* into a number of approximations, and denote this set of approximations by $\mathrm{Ex}(p)$. As an example, consider an $\mathcal{S}$ function consisting of five copies of a 4-bit S-box which has two approximations with squared correlation 2^{-2}, namely $(\texttt{0x3}, \texttt{0xd})$ and $(\texttt{0x7}, \texttt{0x4})$. Then the pattern

$$p = (1, 2^{-2}, 1, 1, 2^{-2})$$

would have value $C(p) = 2^{-4}$ and expands into the set of four approximations

$$\begin{aligned}\mathrm{Ex}(p) = \{&(\texttt{0x03003}, \texttt{0x0d00d}), (\texttt{0x03007}, \texttt{0x0d004}),\\ &(\texttt{0x07003}, \texttt{0x0400d}), (\texttt{0x07007}, \texttt{0x04004})\}.\end{aligned}$$

We note that this expansion can be done in amortized linear time in the size of $\mathrm{Ex}(p)$, independent of t. Moreover, if we just desire to know the input or output masks of the approximations in $\mathrm{Ex}(p)$, these can also be generated in amortized linear time in the number of inputs/outputs.

Now, if we can determine the set $\mathcal{P}$ of patterns with the k' largest values, then clearly $\mathrm{Ex}(\mathcal{P}) = \mathcal{A}$ contains approximations over f with the $|\mathcal{A}|$ largest correlations. This problem can be efficiently solved using the approach of finding critical paths presented in [45]. We briefly outline the idea of the algorithm here:

1. Compute lists L_i of unique values in the LAT of each $\mathcal{S}_i$ and sort them in descending order.
2. Maintain a max-heap of partially determined patterns sorted by their current value. Add a fully undetermined pattern $p = (?, ?, \dots, ?)$ to the heap.
3. Create an empty set $\mathcal{P}$. Repeat the following until $\mathcal{P}$ has the desired size:
   a) Pop the top pattern p off the heap. If it was fully determined, add it to $\mathcal{P}$.
   b) Find the last determined position of p, say p_i, and generate two new patterns:
      i. Replace p_i with the next value in the list L_i and insert the pattern in the heap.
      ii. Replace the undetermined value p_{i+1} with the first value on the list L_{i+1} and insert the pattern in the heap.

Note that this pattern representation, aside from making it easy to find approximations sorted by their correlation, is a very useful time-memory trade-off: each pattern can represent a large number of approximations, allowing us to select a large number of candidate edges for the graph $\bar{G}_{\mathcal{E}}$ without storing them explicitly. However, we need to spend time expanding each pattern whenever we explicitly need the approximations.

About Feistel constructions and other designs The process described here for selecting edges is very efficient for SPN designs. However, it is less clear how to perform the edge selection for other types of designs. For Feistel designs with SPN-like F-functions, we can use the same approach with a slight modification: We let an S-box pattern describe approximations over the F-functions of two consecutive rounds and then derive approximations over two rounds from this pattern. The resulting two-round approximation is shown in Figure 1. This concept can be extended to generalised Feistel constructions.

Concerning radically different design approaches, i.e. ARX and AND-RX designs, we note that [11] and [33] present formulas for the differential probabilities of SPECK and SIMON-like round-functions, respectively. The latter work also gives a method for determining linear correlations of SIMON-like round-functions. These results could potentially be used to perform efficient edge selection for these types of designs.

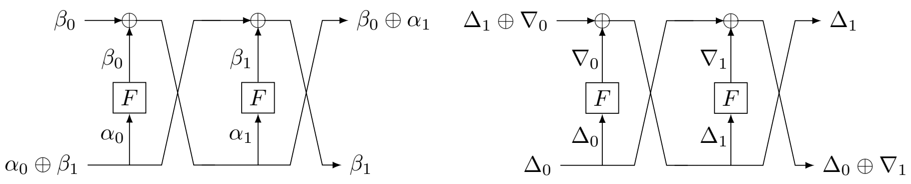


Figure 1: An illustration of how linear approximations/differentials over the F-functions of two consecutive Feistel rounds determine a linear approximation/differential over those rounds.

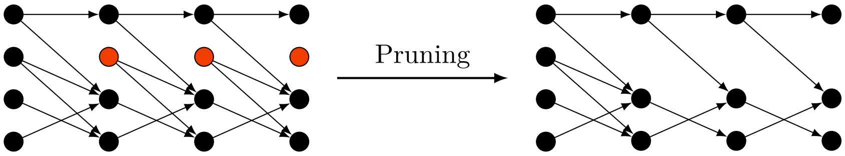


Figure 2: Left: A graph representing parts of linear/differential trails over three rounds of a cipher. Right: The graph after all edges and vertices which are not part of a full trail have been removed.

Graph Pruning

By using the pattern representation introduced above, we can store a large set of interesting edges in a space efficient way. However, not all edges in $\mathcal{A}$ might be relevant when added to the graph $\bar{G}_{\mathcal{E}}$. Consider Figure 2. On the left we show a graph which was generated from a set of patterns, i.e. each edge represents an approximation over the round-function f. The vertices marked in red cannot be a part of a path from a vertex in the first stage to a vertex in the last stage. Hence, we can remove these vertices and all their edges, leaving us with the second, smaller graph, which only contains the information we are interested in. In other words:

- Remove any vertex in S_0 with no outgoing edges.
- Remove any vertex in S_1 to S_{r-1}, if it does not have at least one incoming and one outgoing edge, remove it.
- Remove any vertex in S_r with no incoming edges.

We potentially have to repeat this process until no more vertices can be removed. There is one major problem with naively generating the graph $\bar{G}_{\mathcal{E}}$ in this way, namely that we have to store the initial graph before pruning (which takes roughly $r \cdot |\mathcal{A}|$ space), which can be many times larger than the pruned graph. This essentially limits the number of single round approximations we can consider, i.e. it limits the size of the search space we can cover. In Section 5 we present a number of improvements that alleviate this problem.

4.2 Finding Linear Hulls and Differentials

Once the graph $\bar{G}_{\mathcal{E}}$ has been generated, we can quite easily calculate $w(\alpha \diamond \beta)$ for all pairs $(\alpha, \beta) \in S_0 \times S_r$ by essentially performing a breadth first traversal of the graph for each α while doing some bookkeeping. The idea is the following:

1. Let $\mathcal{H}$ be an empty hash table. Choose an $\alpha \in S_0$ and let $\mathcal{H}(\alpha) = 1$.

2. For each stage S_0 to S_{r-1} of $\bar{G}_{\mathcal{E}}$, do the following:
    a) Create an empty hash table $\mathcal{H}'$.
    b) For each key of $\mathcal{H}$, let u be the corresponding vertex in $\bar{G}_{\mathcal{E}}$. Let $c = \mathcal{H}(u)$. Then, for each edge $u \to v$, if $\mathcal{H}'(v)$ does not exists, let $\mathcal{H}'(v) = c \cdot l(u \to v)$. Otherwise, let $\mathcal{H}'(v) = \mathcal{H}'(v) + c \cdot l(u \to v)$.
    c) Let $\mathcal{H} = \mathcal{H}'$.

3. $\mathcal{H}(\beta)$ now contains $w(\alpha \diamond \beta)$.

4. Repeat for a new value of α.

Note that the number of paths in any $\alpha \diamond \beta$ can also be calculated with a bit of extra bookkeeping in step 2.b. The time complexity of this algorithm is $\mathcal{O}(|S_0| \cdot |\bar{G}_{\mathcal{E}}|)$ and the memory complexity is $\mathcal{O}(|S_0| \cdot |S_r|)$. The memory complexity can be reduced to a constant by only storing the hulls with highest weight calculated so far in Step 3. The time complexity can be reduced to $\mathcal{O}(|\bar{G}_{\mathcal{E}}|)$ by considering all $\alpha \in S_1$ simultaneously. However, this will increase the memory complexity, and in practice we find that this slows down the search due to a poorer cache locality. Moreover, the procedure outlined above is trivially parallelisable over different α values.

It is interesting to note that when the paths contained in $\alpha \diamond \beta$ are not completely edge disjoint, the number of paths can be many orders of magnitude larger than the size of $\bar{G}_{\mathcal{E}}$. Thus, this way of computing $w(\alpha \diamond \beta)$ can be much more efficient than explicitly finding each possible path of $\alpha \diamond \beta$ and adding their lengths. This graph representation of linear hulls therefore allows us to compactly express a large number of trails for a linear approximation, and potentially enables us to capture a much larger part of the linear hull than if we had used a more direct trail search.

5 Improvements

The graph generation algorithm presented in Section 4.1 has two main limitations: First, the time we spend generating the graph is proportional to the number of single round approximations we consider, and second, we initially have to store a much larger graph than we ultimately need. In the following, we present some improvements to the algorithm which prevent this, as well as some additional useful techniques.

5.1 Vertex Generation

We first note that we can perform the pruning step of Section 4.1 without initially storing all $r \cdot |\mathcal{A}|$ edges. Let us denote by $\mathrm{Ex}_{\mathrm{in}}(\mathcal{P})$ and $\mathrm{Ex}_{\mathrm{out}}(\mathcal{P})$ the expansion of $\mathcal{P}$ into only input masks and output masks, respectively. As noted in Section 4.1, for SPN ciphers we can generate $\mathrm{Ex}_{\mathrm{in}}(\mathcal{P})$ or $\mathrm{Ex}_{\mathrm{out}}(\mathcal{P})$ in linear time in the number of inputs or outputs. Moreover, observe that if a vertex in any of the stages S_1 to S_{r-1} does not correspond to a mask contained in $\mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) \cap \mathrm{Ex}_{\mathrm{out}}(\mathcal{P})$, then it will be removed during the pruning process. Thus, we can initially set

$$S_i = \begin{cases} \mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) & i = 0, \\ \mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) \cap \mathrm{Ex}_{\mathrm{out}}(\mathcal{P}) & 1 \le i \le r-1, \\ \mathrm{Ex}_{\mathrm{out}}(\mathcal{P}) & i = r. \end{cases}$$

Then, when adding edges, we generate $\mathcal{A}$ and only add an edge if the corresponding vertices already exists in the graph. Since usually $\mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) \cap \mathrm{Ex}_{\mathrm{out}}(\mathcal{P}) \lll \mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) \cup \mathrm{Ex}_{\mathrm{out}}(\mathcal{P})$, the memory usage is greatly reduced. In practice, the time taken to generate the graph is also reduced, even though we still have to generate the entire set $\mathcal{A}$, as inserting edges and vertices is much more expensive than checking set membership. Finally, we note that we still have to prune the resulting graph to obtain the desired $\bar{G}_{\mathcal{E}}$.

5.2 Graph Compression and Pattern Elimination

While vertex generation somewhat improves memory and running time, it still might be the case that some patterns in $\mathcal{P}$ ultimately did not contribute to $\bar{G}_{\mathcal{E}}$, i.e. all edges expressed by the pattern are removed during pruning. We will call such a pattern a *dead pattern*. Clearly, it would be preferable if we ignored dead patterns completely. We now present a technique for finding dead patterns quickly while using little memory.

We first introduce the notion of a compression function $g_j(x) : \mathbb{F}_2^n \to \mathbb{F}_2^{n/j}$. Let $y = g_j(x)$. Then for $0 \le i < j$, $y_i = 1$ iff $(x_{j \cdot i}, \dots, x_{j \cdot (i+1)-1})$ is non-zero. For example,

$$g_4(\texttt{0xf1a0000500705200}) = \texttt{0b1110000100101100} = \texttt{0xe12c},$$

i.e. the value `0xf1a0000500705200` $\in \mathbb{F}_2^{64}$ is compressed to the value `0xe12c` $\in \mathbb{F}_2^{16}$. Note that this is similar to the concept of truncated differentials/approximations. With some abuse of notation, for a graph G will say that $g_j(G)$ is the graph obtained by applying g_j to all vertices, identifying vertices in the same stage that have the same compressed value, and then removing multiple edges. An example of this process is shown in Figure 3.

We can use compression to find dead patterns in a space efficient way. Instead of generating $\bar{G}_{\mathcal{E}}$, we first generate an approximation to $g_j(\bar{G}_{\mathcal{E}})$, say $\hat{g}_j(\bar{G}_{\mathcal{E}})$, by applying g_j to all values when generating the graph. Note that this does not yield

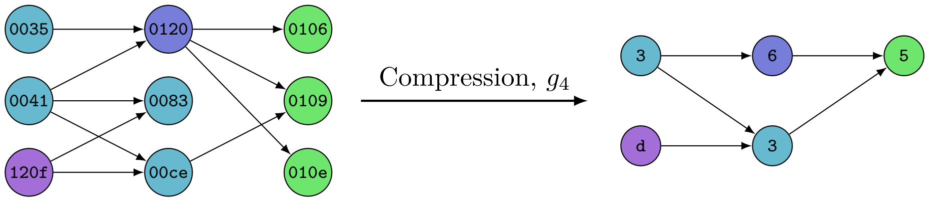


Figure 3: An example of graph compression using the compression function g_4. The values of the vertices are shown in hexadecimal notation. Vertices in the same stage with non-zero nibbles in the same position are identified, and any multiple edges are removed.

$g_j(\bar{G}_{\mathcal{E}})$; any path between two vertices in $\bar{G}_{\mathcal{E}}$ is preserved in $\hat{g}_j(\bar{G}_{\mathcal{E}})$, but there might be some additional false paths. As a result, when applying pruning to the compressed graph, some vertices might not be removed, although they would have been removed in the actual graph.

Note now that if a pattern is dead when considering $\hat{g}_j(\bar{G}_{\mathcal{E}})$, it is also dead when considering $\bar{G}_{\mathcal{E}}$ (although the converse does not hold). Thus, we can use $\hat{g}_j(\bar{G}_{\mathcal{E}})$ to remove some dead patterns. We propose the following approach to removing dead patterns while conserving memory:

1. Generate a set of patterns $\mathcal{P}$.
2. Pick a $j > 1$ such that j is a power of two, and do the following:
   a) Generate the graph $\hat{g}_j(\bar{G}_{\mathcal{E}})$ as described above.
   b) Remove any patterns from $\mathcal{P}$ which are dead according to $\hat{g}_j(\bar{G}_{\mathcal{E}})$.
   c) If $j = 2$ then stop. Otherwise set $j = j/2$ and repeat.

The main insight here is that initially $\mathrm{Ex}(\mathcal{P})$ can be many times larger than what we can store in memory. By gradually using a finer compression, we decrease the size of $\mathrm{Ex}(\mathcal{P})$, while still keeping the intermediate graphs manageable and without losing any information from the original search space. In practice, for ciphers with a block size of 64 bits and 4-bit S-boxes, we find that starting with $j = 4$ works well. How many dead patterns occur varies between different designs, but we find that in general, if there are few dead patterns, we also rarely need to use a large set $\mathcal{A}$ to get good results.

Speedup for SPN Ciphers While the above processes has the potential to greatly reduce memory usage, we still need to calculate the initial set $\mathcal{A}$ at least once, and potentially multiple times if few patterns are eliminated. For SPN ciphers we can improve this by observing that if j is a multiple of the width of the S-box, we can calculate the compression of an approximation (α, β) simply by calculating $\mathcal{L}^{-1}(\beta)$. This is true, since if the output mask of any S-box is non-zero, then so is the

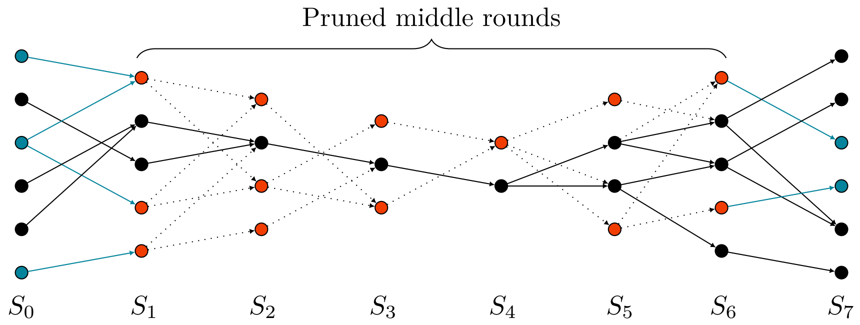


Figure 4: An illustration of vertex anchoring. The black and red graph represents trails built from the set of single round approximations $\mathcal{A}$. The red subgraph would be removed if the red/black graph was pruned. The blue anchor vertices are added to prevent the red subgraph from being removed, increasing the number of trails found.

corresponding input mask of that S-box, which is all we need to know to calculate the compressed value of α. In this case we can therefore generate $\hat{g}_j(\bar{G}_\mathcal{E})$ in time $\mathcal{O}(|\mathrm{Ex}_{\mathrm{in}}(\mathcal{P})| + |\mathrm{Ex}_{\mathrm{out}}(\mathcal{P})|)$ (recall that vertex generation has this time complexity for SPN ciphers). This greatly improves the running time of the algorithm for this type of ciphers.

5.3 Vertex Anchoring

One big limitation with the algorithm presented here is that the search space is limited by how big a set $\mathcal{A}$ the available computing power allows us to consider. While the improvements presented so far increase the possible size of $\mathcal{A}$, we wont be able to find paths that locally have very short edges. Note that such a path might still be comparatively long, if all other edges of the path are long.

Without having cipher specific insight, it seems difficult to know when it is beneficial to add a locally bad edge, and especially which edge to add. This problem is exacerbated by the fact that short edges represent approximations which usually activate many S-boxes, and so the number of short edges is usually much larger than the number of long edges.

We propose a partial solution to this problem by introducing a technique called *vertex anchoring*. Consider the example given in Figure 4. Here, the red and black subgraph is the graph we might obtain from a set of approximations $\mathcal{A}$, before pruning. Note that all the red vertices would be removed from this graph during pruning, as they are not part of a path from a vertex in S_0 to a vertex in S_7. Nevertheless, the red paths might be quite long paths and it is therefore potentially wasteful to discard such nearly complete paths. Instead, note that we can freely add a vertex to S_0, as long there exists an edge between this vertex and any vertex of S_1. Such an edge would be outside the set $\mathcal{A}$, and including it will effectively increase our search space. These edges are shown in blue in Figure 4, and they ensure that the red subgraph is not removed during any subsequent pruning. As the result of these observations, we propose the following approach:

1. Generate the graph $\bar{G}_{\mathcal{E}}$ for $r-2$ rounds. Denote the stages $S_1, \dots, S_{r-1}$.
2. Add a stage S_0 at the start (respectively S_r at the end) of $\bar{G}_{\mathcal{E}}$ consisting of all vertices and edges in $\mathcal{A}$ which are incident to a vertex in S_1 (S_{r-1}).
3. For any vertex in S_1 (respectively S_{r-1}) which does not have an incoming (outgoing) edge, find the longest possible edge going into (out of) this vertex, and add this edge and its start (end) vertex to S_0 (S_r).

For SPN ciphers, Step 3 can be done efficiently simply by finding the output (input) mask to the S-box layer represented by each vertex, and then choosing the best possible input (output) masks for each S-box. In practice, we limit the number of anchor vertices added so as to not increase the search time too much. We find that this method dramatically improves the results for some ciphers.

5.4 Parallelisation

As a practical improvement to the algorithm, we note that most aspects can be parallelised. In particular, whenever we need to calculate $\mathrm{Ex}(\mathcal{P})$, $\mathcal{P}$ can be split into parts and distributed across different threads. Often a main thread will have to collect the results form each of the worker threads, e.g. when calculating $\mathrm{Ex}_{\mathrm{in}}(\mathcal{P}) \cap \mathrm{Ex}_{\mathrm{out}}(\mathcal{P})$ during vertex generation, but this work is quite minimal. Moreover, as mentioned in Section 4.2, the search for hulls can easily be parallelised by distributing different α values across threads. Thus, while the scaling is not perfect, the algorithm benefits quite heavily from increasing the number of threads, especially when $\mathcal{A}$ is large, which is often the case since we want to cover as large a search space as possible.

6 Searching for Linear Approximations and Differentials

We applied the algorithm described here to 17 different SPN ciphers. The types of designs vary from ciphers with very lightweight permutation layers, such as PRESENT, to ciphers with very heavy permutation layers, such as KLEIN. While we did also apply the algorithm to some Feistel designs (i.e. TWINE and MIBS), the main improvements over the basic algorithm presented in Section 5 apply to strict SPN designs, and we were unable to obtain any interesting results for these ciphers due to the increased running time. Moreover, the current implementation of the algorithm only supports ciphers with identical S-boxes, although adding this functionality would not slow down the implementation noticeably.

Table 1: Results for linear cryptanalysis obtained using the algorithm presented in this work. $\mathcal{A}$ is the set of single round approximations, a is the number of anchor vertices, $\alpha \diamond \beta$ is the set of trails found for the best approximation, ELP is the expected squared correlation, and T_g and T_s is the time in hours to generate and search through the graph, respectively. Entries annotated by † indicate an improvement over a previously published result.

Cipher (Total rounds, block size)	**Rounds**	$\lvert\mathcal{A}\rvert$	a	$\lvert\alpha \diamond \beta\rvert$	**ELP**	T_{g}	T_{s}
AES [41]	3	$2^{29.9}$	$2^{24.0}$	2^{1}	$2^{-53.36}$	0.0	0.0
(10, 128)	4	$2^{38.8}$	$2^{24.0}$	2^{4}	$2^{-147.88}$	2.5	20.0
EPCBC-48 [44]	15 † [19]	$2^{26.1}$	–	$2^{31.3}$	$2^{-43.74}$	0.0	0.4
(32, 48)	16 † [19]	$2^{26.1}$	–	$2^{34.0}$	$2^{-46.77}$	0.0	0.4
EPCBC-96 [44]	31	$2^{27.6}$	–	$2^{63.6}$	$2^{-94.47}$	0.0	0.4
(32, 96)	32	$2^{27.6}$	–	$2^{63.6}$	$2^{-97.59}$	0.0	0.4
FLY [32]	8	$2^{32.5}$	–	$2^{6.5}$	$2^{-54.83}$	0.1	6.0
(20, 64)	9	$2^{32.5}$	–	$2^{6.1}$	$2^{-63.00}$	0.2	8.8
GIFT-64 [7]	11	$2^{31.8}$	–	$2^{5.1}$	$2^{-55.00}$	0.1	8.0
(28, 64)	12	$2^{32.7}$	–	$2^{3.6}$	$2^{-64.00}$	0.2	41.5
KHAZAD [8]	2	$2^{18.3}$	$2^{25.0}$	2^{0}	$2^{-37.97}$	0.0	0.0
(8, 64)	3	$2^{30.1}$	$2^{25.0}$	2^{0}	$2^{-68.01}$	0.2	0.2
KLEIN [29]	5	$2^{30.8}$	$2^{17.0}$	2^{0}	$2^{-46.0}$	0.0	0.0
(12, 64)	6	$2^{39.6}$	$2^{16.9}$	2^{0}	$2^{-66.0}$	0.3	0.0
LED [30] (32, 64)	4	$2^{24.7}$	2^{25}	2^{2}	$2^{-48.68}$	0.0	0.9
MANTIS$_7$ [9] (2 · 8, 64)	2 · 4	$2^{34.3}$	$2^{24.0}$	$2^{15.0}$	$2^{-49.05}$	0.1	0.0
Midori64 [6]	6	$2^{44.3}$	–	$2^{19.0}$	$2^{-53.02}$	25.9	0.8
(16, 64)	7	$2^{46.5}$	–	$2^{21.9}$	$2^{-62.88}$	53.1	5.5
PRESENT [14]	23 † [40]	$2^{31.1}$	–	$2^{55.0}$	$2^{-61.00}$	0.1	6.8
(31, 64)	24 † [40]	$2^{31.1}$	–	$2^{57.9}$	$2^{-63.61}$	0.1	6.9
	25 † [40]	$2^{31.1}$	–	$2^{60.7}$	$2^{-66.21}$	0.1	6.9
PRIDE [3]	15	$2^{27.1}$	–	2^{0}	$2^{-58.00}$	0.0	0.0
(20, 64)	16	$2^{37.4}$	–	2^{3}	$2^{-63.99}$	1.8	0.0
PRINCE [18]	2 · 3	$2^{18.1}$	–	$2^{2.0}$	$2^{-54.00}$	0.0	0.0
(2 · 6, 64)	2 · 4	$2^{38.3}$	–	$2^{6.8}$	$2^{-63.82}$	2.1	0.4
PUFFIN [23] (32, 64)	32	$2^{26.8}$	–	$2^{112.4}$	$2^{-51.90}$	0.0	0.0
QARMA [5] (2 · 8, 64)	2 · 3	$2^{24.8}$	$2^{24.0}$	$2^{5.0}$	$2^{-53.71}$	0.0	0.0
RECTANGLE [47]	12 † [47]	$2^{31.1}$	–	$2^{15.0}$	$2^{-52.27}$	0.1	21.1
(25, 64)	13 † [47]	$2^{31.1}$	–	$2^{15.9}$	$2^{-58.14}$	0.1	25.9
	14 † [47]	$2^{31.1}$	–	$2^{18.3}$	$2^{-62.98}$	0.1	31.1
SKINNY-64 [9]	8	$2^{41.4}$	$2^{23.7}$	$2^{34.4}$	$2^{-50.46}$	0.7	50.7
(32, 64)	9	$2^{41.4}$	$2^{23.9}$	$2^{31.3}$	$2^{-69.83}$	0.4	8.9

Table 2: Results for differential cryptanalysis obtained using the algorithm presented in this work. $\mathcal{D}$ is the set of single round differentials, a is the number of anchor vertices, $\Delta \diamond \nabla$ is the set of trails found for the best differential, EDP is the expected differential probability, and T_g and T_s is the time in hours to generate and search through the graph, respectively. Entries annotated by † indicate an improvement over a previously published result.

Cipher (Total rounds, block size)	**Rounds**	$\lvert\mathcal{D}\rvert$	a	$\lvert\Delta \diamond \nabla\rvert$	**EDP**	T_{g}	T_{s}
AES [41] (10, 128)	3	$2^{18.7}$	$2^{24.0}$	2^{0}	$2^{-54.00}$	0.0	0.0
	4	$2^{36.9}$	$2^{24.0}$	2^{0}	$2^{-150.00}$	0.7	0.3
EPCBC-48 [44] (32, 48)	13	$2^{28.4}$	–	$2^{21.2}$	$2^{-43.86}$	0.1	13.7
	14	$2^{28.4}$	–	$2^{20.4}$	$2^{-47.65}$	0.1	14.0
EPCBC-96 [44] (32, 96)	20	$2^{32.8}$	–	$2^{16.9}$	$2^{-92.73}$	1.1	21.6
	21	$2^{32.8}$	–	$2^{19.9}$	$2^{-97.78}$	1.1	22.6
Fly [32] (20, 64)	8	$2^{31.6}$	–	$2^{4.9}$	$2^{-55.76}$	0.1	2.6
	9	$2^{33.2}$	–	$2^{7.3}$	$2^{-63.35}$	0.2	17.8
GIFT-64 [7] (28, 64)	12 † [48]	$2^{22.4}$	–	$2^{3.3}$	$2^{-56.57}$	0.0	0.0
	13	$2^{22.4}$	–	$2^{3.6}$	$2^{-60.42}$	0.0	0.0
Khazad [8] (8, 64)	2	$2^{25.8}$	$2^{24.8}$	2^{0}	$2^{-45.42}$	0.0	0.0
	3	$2^{25.8}$	$2^{25.0}$	2^{0}	$2^{-81.66}$	0.0	0.0
KLEIN [29] (12, 64)	5	$2^{30.8}$	$2^{17.0}$	$2^{1.0}$	$2^{-45.91}$	0.0	0.0
	6	$2^{39.7}$	$2^{24.0}$	$2^{1.0}$	$2^{-69.00}$	0.3	6.4
LED [30] (32, 64)	4	$2^{37.7}$	$2^{24.0}$	2^{1}	$2^{-49.42}$	0.5	0.1
MANTIS$_7$ [9] ($2 \cdot 8$, 64)	$2 \cdot 4$	$2^{37.7}$	–	$2^{18.6}$	$2^{-47.98}$	0.9	0.1
Midori64 [6] (16, 64)	6	$2^{42.2}$	$2^{23.9}$	$2^{19.6}$	$2^{-52.37}$	1.6	1.0
	7	$2^{42.2}$	$2^{23.9}$	$2^{22.8}$	$2^{-61.22}$	1.0	0.9
PRESENT [14] (31, 64)	15	$2^{30.3}$	–	$2^{27.2}$	$2^{-58.00}$	0.1	16.2
	16 † [1]	$2^{30.3}$	–	$2^{28.9}$	$2^{-61.80}$	0.1	18.0
	17	$2^{30.3}$	–	$2^{32.9}$	$2^{-63.52}$	0.1	18.8
PRIDE [3] (20, 64)	15	$2^{35.9}$	$2^{23.6}$	$2^{5.0}$	$2^{-58.00}$	0.5	36.5
	16	$2^{35.9}$	$2^{23.6}$	$2^{17.4}$	$2^{-63.99}$	0.5	44.1
PRINCE [18] ($2 \cdot 6$, 64)	$2 \cdot 3$ † [20]	$2^{14.0}$	2^{19}	2^{1}	$2^{-55.91}$	0.0	0.0
	$2 \cdot 4$	$2^{38.7}$	–	$2^{9.0}$	$2^{-67.32}$	3.0	1.0
PUFFIN [23] (32, 64)	32	$2^{26.0}$	–	$2^{63.7}$	$2^{-59.63}$	0.0	0.0
QARMA [5] ($2 \cdot 8$, 64)	$2 \cdot 3$	$2^{24.8}$	$2^{26.0}$	$2^{7.3}$	$2^{-56.47}$	0.1	0.0
RECTANGLE [47] (25, 64)	13 † [47]	$2^{31.1}$	–	$2^{15.3}$	$2^{-55.64}$	0.1	32.2
	14 † [47]	$2^{31.1}$	–	$2^{15.9}$	$2^{-60.64}$	0.1	41.3
	15 † [47]	$2^{31.1}$	–	$2^{18.2}$	$2^{-65.64}$	0.1	50.2
SKINNY-64 [9] (32, 64)	8	$2^{39.4}$	$2^{24.0}$	$2^{31.0}$	$2^{-50.72}$	0.2	15.0
	9	$2^{41.7}$	$2^{23.8}$	$2^{31.2}$	$2^{-69.64}$	0.4	6.4

Note that we investigate three ciphers that use a PRINCE-like design, namely PRINCE, MANTIS, and QARMA. For these ciphers, we generate a graph for the first half of the rounds, as described above, reverse this graph, and then stitch these two graphs together through the central permutation layer.

6.1 Results for ELP and EDP

We ran the algorithm using an Intel Xeon E5-2650 v4 processor (24 threads at 2.2 GHz) with 256 GB of memory available. The results for linear cryptanalysis are shown in Table 1 and the results for differential cryptanalysis in Table 2. Note that the number of rounds stated here is the number of non-linear layers (i.e. S-box layers) applied.

The number of single round approximations or differentials considered when generating the graph is the smallest that gave the stated result – for most ciphers, we investigated larger search spaces without obtaining any improvements. In general, it is interesting to note that for the majority of ciphers, actually generating the graph is quite fast, while searching through the graph can take considerably longer. If one has an idea of what input/output masks/differences are good, the graph can be restricted to paths between these interesting values, which will greatly reduce the search time. A general strategy for using the algorithm could therefore be to find some preliminary interesting approximations/differentials using a small search space, and then increase the search space while restricting the graph to these approximations/differentials in order to improve the estimates.

Entries annotated with a † indicate improvements over previous best results. Entries that are not annotated are either new or do not improve on known results. For many of the ciphers, the search found multiple approximations/differentials that were equally good. It is therefore possible that multiple linear/differential attacks could be mounted on a larger number of rounds than stated here.

We highlight a few interesting results. For RECTANGLE, the designers did take into account multiple trails in [47], and estimated that over 14 rounds the best differential has EDP $2^{-62.83}$. We improve this to $2^{-60.64}$, demonstrating that being able to include a larger number of trails can improve estimates.

For GIFT-64, [48] used a MILP based tool to find a 12 round differential trail with probability 2^{-60}. By taking into account multiple trails, we improve this to $2^{-56.57}$ and find a 13 round differential with probability $2^{-60.42}$. Thus, we can potentially extend their attack by one round.

For PRESENT, we improve some results of [1]. In particular, we improve their result for 16 round differentials from $2^{-62.58}$ to $2^{-61.80}$ and furthermore find a 17 round differential with probability $2^{-63.52}$. For linear cryptanalysis, we match the results of [1], although interestingly we find fewer trails. This shows that the algorithm presented here can match or even improve the results obtained by the partial correlation/difference transition matrix method, all the while being more versatile.

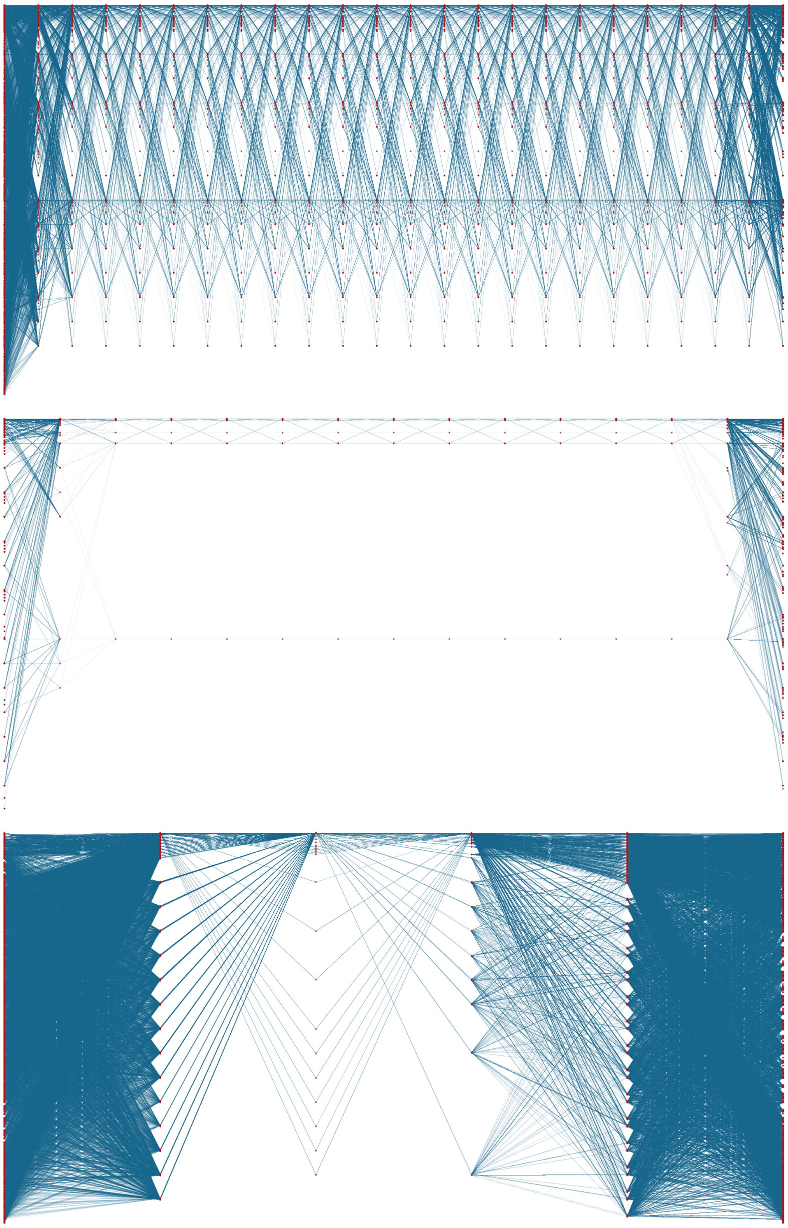


Figure 5: Examples of linear hull graphs generated by our algorithm. Top: 23 rounds of PRESENT using $|\mathcal{A}| = 2^{24.7}$ single round approximations. Middle: 14 rounds of PRIDE, also with $|\mathcal{A}| = 2^{24.7}$. Bottom: 5 rounds of KLEIN, with $|\mathcal{A}| = 2^{26.8}$ and using 2^{17} anchoring vertices.

6.2 Visualising Trail Graphs

An interesting side effect of applying our new algorithm is that we can visualise the linear/differential trails in order to get a better understanding of how the cipher's structure influences its resistance to linear and differential cryptanalysis. Figure 5 show the linear hull graphs that we generated for three different ciphers: PRESENT, PRIDE, and KLEIN. The vertices in each stage are ordered by their value as integers.

While the search spaces selected for the three ciphers are comparable in size, the resulting graphs have widely different structures. The graph for PRESENT show that each stage is identical, and that the stages are highly connected. Thus, as observed in [40], there exists a very large number of trails for many approximations of PRESENT that have similar structure and therefore similar correlation contribution. PRIDE also exhibits identical stages, and we can even observe iterative trails, but there are only very few vertices in each stage, preventing the number of trails from exploding. The graph for KLEIN (which has a very heavy linear layer), shows a very large number of edges in the graph, but the structure of the stages vary, resulting in no clustering of trails. Indeed, Table 1 shows that we only found one trail for the best approximations over 5 and 6 rounds.

7 Correlation Distributions

Determining the ELP and EDP of the best linear approximations and differentials is important when assessing the strength of a cipher against these attacks. However, these summary statistics do not paint to full picture: in reality, the linear correlation and differential probability vary over the key space, and more detailed knowledge about the distribution of these values can lead to stronger distinguishers. As an example, [17] demonstrated how asymmetries in the joint correlation distribution of multiple linear approximations of DES can be used to improve attacks.

For differentials, not much is known about how the differential probabilities vary as the key changes. For linear cryptanalysis, there has been an increased interest in developing more accurate models for the key dependent behaviour, see e.g. [12, 13, 15, 16, 31]. This line of research is in large part facilitated by the following useful result.

Theorem 3 ([25])**.** *Let (α, β) be a linear approximation of an SPN cipher and let $\bar{k}$ denote the concatenation of the cipher's round keys for the encryption key k. Then the linear correlation is given by*

$$C^k_{(\alpha,\beta)} = \sum_U (-1)^{s_U \oplus \langle U, \bar{k} \rangle} |C^k_U|,$$

where the sum is over trails $U = (\alpha, \ldots, \beta)$, s_U is the sign bit of U, and $|C^k_U|$ is independent of k.

The above theorem indicates that for an SPN cipher we can determine the key dependent correlation by adjusting the sign of each trail's correlation contribution.

Consequently, we can estimate the distribution over the key-space by doing this for a large number of keys. A similar result holds for Feistel ciphers with SPN like F-functions.

7.1 Finding Key-Dependent Distributions

Our algorithm for estimating ELP can easily be adapted to efficiently calculate key dependent correlations instead. The main idea is simply to construct the graph $\bar{G}_{\mathcal{E}}$, but using the signed correlation values instead of the squared correlation as edge weights, and then adjust the sign of the edges for each different key. Note that we can easily find the signs of each edge after we have generated $\bar{G}_{\mathcal{E}}$, as we know the input and output masks each edge represents. Thus, we can find the signed correlation of an approximation by using a slightly adapted version of the algorithm presented in Section 4.2 (we assume that a pre-whitening key k_0 is used):

1. Choose an encryption key k.
2. Let $\mathcal{H}$ be an empty hash table. Choose an $\alpha \in S_1$ and let $\mathcal{H}(\alpha) = (-1)^{\langle \alpha, k_0 \rangle}$.
3. For each stage S_0 to S_{r-1} of $\bar{G}_{\mathcal{E}}$, do the following:
   a) Let k_i be the current round-key.
   b) Create an empty hash table $\mathcal{H}'$.
   c) For each key of $\mathcal{H}$, let u be the corresponding vertex in $\bar{G}_{\mathcal{E}}$. Let $c = \mathcal{H}(u)$. Then, for each edge $u \to v$, if $\mathcal{H}'(v)$ does not exists, let $\mathcal{H}'(v) = c \cdot (-1)^{\langle v, k_i \rangle} \cdot l(u \to v)$. Otherwise, let $\mathcal{H}'(v) = \mathcal{H}'(v) + c \cdot (-1)^{\langle v, k_i \rangle} \cdot l(u \to v)$.
   d) Let $\mathcal{H} = \mathcal{H}'$.
4. $\mathcal{H}(\beta)$ now contains $C^k_{(\alpha,\beta)}$.
5. Repeat for as many encryption keys as desired.

Clearly, this procedure only calculates a partial sum of $C^k_{(\alpha,\beta)}$. To obtain a better approximation of the actual value, we use the signal/noise decomposition technique proposed in [15]. This technique is summarised the in the following lemma.

Lemma 4 ([15]). *Let $\mathcal{S}$ be a set of strong linear trails for an approximation (α, β). Then $C^k_{(\alpha,\beta)}$ can be approximated by*

$$C^k_{(\alpha,\beta)} = \left(\sum_{U \in \mathcal{S}} (-1)^{s_U \oplus \langle U, \bar{k} \rangle} |C^k_U| \right) + \mathcal{N}(0, 2^{-n}),$$

where $\mathcal{N}(0, 2^{-n})$ denotes the normal distribution with mean 0 and variance 2^{-n}.

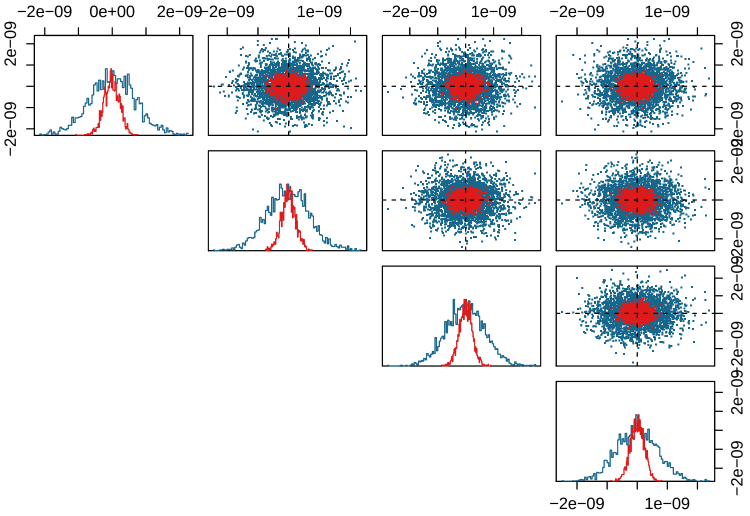


Figure 6: Shown in blue, the pairwise joint linear correlation distributions for four linear approximations of 23 rounds of PRESENT. The correlation distribution of an ideal cipher is shown in red. The plot shows that the joint correlation distribution for PRESENT is close to normal.

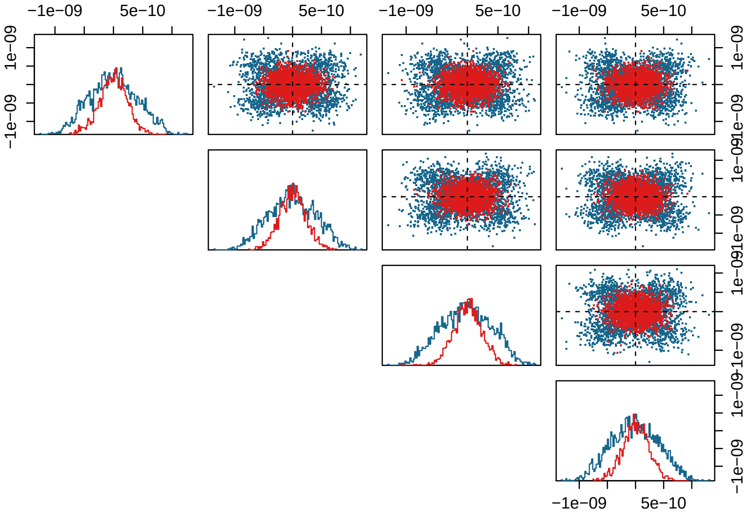


Figure 7: Shown in blue, the pairwise joint linear correlation distributions for four linear approximations over 9 rounds of FLY. The correlation distribution of an ideal cipher is shown in red. For each pair of approximations we observe four distinct clusters in the distributions.

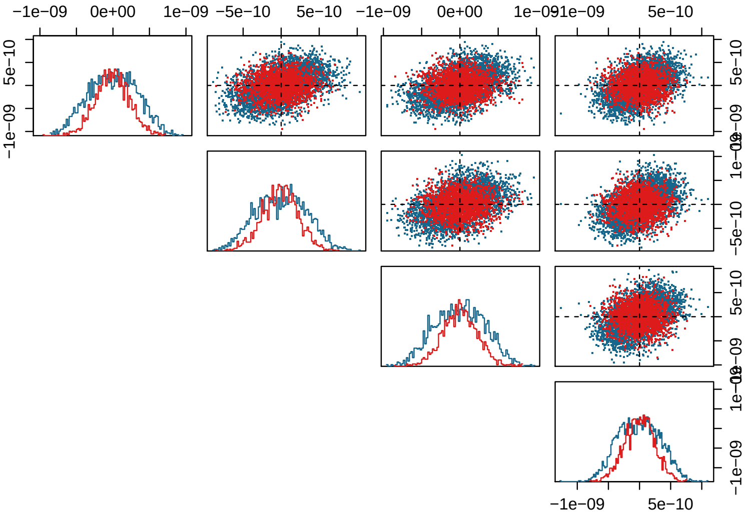


Figure 8: Shown in blue, the pairwise joint linear correlation distributions for four linear approximations over 12 rounds of `GIFT-64`. The correlation distribution of an ideal cipher is shown in red. For each pair of approximations we observe two distinct clusters in the distributions. This indicates a dependence between the approximations.

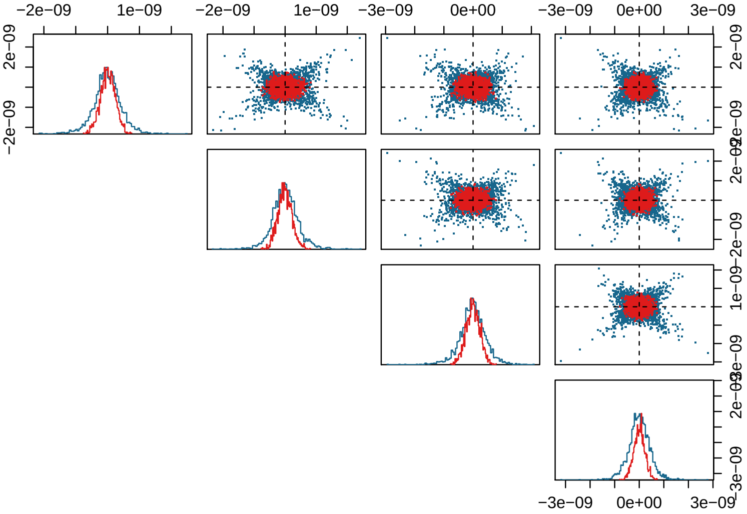


Figure 9: Shown in blue, the pairwise joint linear correlation distributions for four linear approximations over 14 rounds of RECTANGLE. The correlation distribution of an ideal cipher is shown in red. For each pair of approximations we observe a significant deviation from normality, manifested by very long tails of the distributions.

7.2 Results

We have applied the above technique to some of the ciphers we investigated in Section 6. That is, we calculated the partial sum of $C^k_{(\alpha,\beta)}$ for 10 000 randomly chosen encryption keys, and then added the noise distribution $\mathcal{N}(0, 2^{-n})$ to the resulting data sets. We note that when doing this for only a few approximations, the process takes at most a few minutes, depending on the cipher. In light of the results of [17] we consider the joint distributions of four different ciphers.

Figure 6 shows the pairwise joint distributions of four linear approximations over 23 rounds of PRESENT. As a reference, the correlation distribution of an ideal cipher is shown, i.e. a bivariate normal distribution with marginals $\mathcal{N}(0, 2^{-n})$. In this case, the correlation distributions appear to be close to normal and entirely independent, resulting in a joint normal distribution. This matches the observations made in [16].

Figure 7 shows the same picture but for 9 rounds of Fly. However, in this case, while the marginal correlation distributions appear the be close to normal, when considering the joint distributions, we can see that there are four clusters of observations for each pair of approximations. A similar situation occurs over 12 rounds of `GIFT-64`, as shown in Figure 8, only here we only observe two clusters for each pair. As in [17], this would indicate that there is a heavy overlap in the trails of the approximations, resulting in a strong dependence between the signs of the correlations.

Finally, we consider approximations over 14 rounds of RECTANGLE in Figure 9. Here, we observe even stranger behaviour, as the marginal distributions do not even appear to be normal. In fact, the distributions have much longer tails than expected, which would indicate that there is a large percentage of weak keys for which a linear attack would work better than expected.

The last three examples show that even if the ELP is close to the value expected from an ideal block cipher, the actual correlation distributions might exhibit additional behaviour which can be exploited in an attack. Attacks of this type warrant further investigation, and hopefully the algorithm presented in this work will make this line of research easier.

8 Future Work

The algorithm presented in this work has much potential for further extensions and improvements. First and foremost, it would be very useful to find improvements similar to those of Section 5 that apply to other types of ciphers, in particular Feistel designs and designs that are not based on S-boxes. This is closely related to the strategy for selecting edges, discussed in Section 4.1. As also pointed out there, it would be interesting to use the results of [11, 33] to develop an edge selection strategy for ARX and AND-RX designs.

In more general terms, it would also be highly interesting to explore different heuristics for the edge selection, as selecting the longest edges is not necessarily the best strategy. This consideration has two aspects: First, we might obtain globally better results by including very bad edges locally, and second, for all the ciphers we investigated, we end up only using a very small subset of the single round approximations/differentials we initially consider. As such, we waste much time and memory considering edges we are ultimately not interested in. A better heuristic that can filter out (some) of these edges early would potentially improve the algorithm.

Finally, we entertain the possibility that the general graph framework could be extended to other types of cryptanalysis. Indeed, we could describe any property that propagates through the round-function of a cipher as a path through a graph. As such, it might be possible to apply the technique to search for e.g. division properties.

References

[1] Mohamed Ahmed Abdelraheem. “Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers”. In: *Information Security and Cryptology - ICISC 2012*. 2012, pp. 368–382.

[2] Mohamed Ahmed Abdelraheem, Javad Alizadeh, Hoda A. AlKhzaimi, Mohammad Reza Aref, Nasour Bagheri, and Praveen Gauravaram. “Improved Linear Cryptanalysis of Reduced-Round SIMON-32 and SIMON-48”. In: *Progress in Cryptology - INDOCRYPT 2015*. 2015, pp. 153–179.

[3] Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin. “Block Ciphers - Focus on the Linear Layer (feat. PRIDE)”. In: *Advances in Cryptology - CRYPTO 2014*. 2014, pp. 57–76.

[4] Ralph Ankele and Stefan Kölbl. “Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis”. In: *Selected Areas in Cryptography - SAC 2018*. 2018.

[5] Roberto Avanzi. “The QARMA Block Cipher Family. Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes”. In: *IACR Transactions on Symmetric Cryptology* 2017.1 (2017), pp. 4–44.

[6] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. “Midori: A Block Cipher for Low Energy”. In: *Advances in Cryptology - ASIACRYPT 2015*. 2015, pp. 411–436.

[7] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. “GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption”. In: *Cryptographic Hardware and Embedded Systems - CHES 2017*. 2017, pp. 321–345.

[8] Paulo S.L.M. Barreto and Vincent Rijmen. “The Khazad Legacy-Level Block Cipher”. In: *Primitive submitted to NESSIE* 97 (2000).

[9] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS". In: *Advances in Cryptology - CRYPTO 2016*. 2016, pp. 123–153.

[10] Eli Biham and Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In: *Advances in Cryptology - CRYPTO '90*. 1990, pp. 2–21.

[11] Alex Biryukov and Vesselin Velichkov. "Automatic Search for Differential Trails in ARX Ciphers". In: *Topics in Cryptology - CT-RSA 2014*. 2014, pp. 227–250.

[12] Céline Blondeau and Kaisa Nyberg. "Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis". In: *IACR Transactions on Symmetric Cryptology* 2016.2 (2016), pp. 162–191.

[13] Céline Blondeau and Kaisa Nyberg. "Joint Data and Key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and Its Impact to Data Complexity". In: *Design, Codes and Cryptography* 82.1-2 (2017), pp. 319–349.

[14] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. "PRESENT: An Ultra-Lightweight Block Cipher". In: *Cryptographic Hardware and Embedded Systems - CHES 2007*. 2007, pp. 450–466.

[15] Andrey Bogdanov and Elmar Tischhauser. "On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2". In: *Fast Software Encryption, FSE 2013*. 2013, pp. 19–38.

[16] Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. "Multivariate Profiling of Hulls for Linear Cryptanalysis". In: *IACR Transactions on Symmetric Cryptology* 2018.1 (2018), pp. 101–125.

[17] Andrey Bogdanov and Philip S. Vejre. "Linear Cryptanalysis of DES with Asymmetries". In: *Advances in Cryptology - ASIACRYPT 2017*. 2017, pp. 187–216.

[18] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. "PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract". In: *Advances in Cryptology - ASIACRYPT 2012*. 2012, pp. 208–225.

[19] Stanislav Bulygin. "More on Linear Hulls of PRESENT-like Ciphers and a Cryptanalysis of Full-Round EPCBC-96". In: *IACR Cryptology ePrint Archive* 2013 (2013), p. 28.

[20] Anne Canteaut, Thomas Fuhr, Henri Gilbert, María Naya-Plasencia, and Jean-René Reinhard. "Multiple Differential Cryptanalysis of Round-Reduced PRINCE". In: *Fast Software Encryption, FSE 2014*. 2014, pp. 591–610.

[21] Jiageng Chen, Atsuko Miyaji, Chunhua Su, and Jesen Teh. “Accurate Estimation of the Full Differential Distribution for General Feistel Structures”. In: *Information Security and Cryptology, 2015.* 2015, pp. 108–124.

[22] Jiageng Chen, Atsuko Miyaji, Chunhua Su, and Jesen Teh. “Improved Differential Characteristic Searching Methods”. In: *IEEE 2nd International Conference on Cyber Security and Cloud Computing, CSCloud 2015.* 2015, pp. 500–508.

[23] Huiju Cheng, Howard M. Heys, and Cheng Wang. “PUFFIN: A Novel Compact Block Cipher Targeted to Embedded Digital Systems”. In: *11th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2008.* 2008, pp. 383–390.

[24] Joan Daemen and Vincent Rijmen. “Probability distributions of correlation and differentials in block ciphers”. In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 221–242.

[25] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002.

[26] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. “Heuristic Tool for Linear Cryptanalysis with Applications to CAESAR Candidates”. In: *Advances in Cryptology - ASIACRYPT 2015.* 2015, pp. 490–509.

[27] Pierre-Alain Fouque, Gaëtan Leurent, and Phong Q. Nguyen. “Automatic Search of Differential Path in MD4”. In: *IACR Cryptology ePrint Archive* 2007 (2007), p. 206.

[28] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. “MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck”. In: *Fast Software Encryption, FSE 2016.* 2016, pp. 268–288.

[29] Zheng Gong, Svetla Nikova, and Yee Wei Law. “KLEIN: A New Family of Lightweight Block Ciphers”. In: *RFID. Security and Privacy, RFIDSec 2011.* 2011, pp. 1–18.

[30] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. “The LED Block Cipher”. In: *Cryptographic Hardware and Embedded Systems - CHES 2011.* 2011, pp. 326–341.

[31] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. “Capacity and Data Complexity in Multidimensional Linear Attack”. In: *Advances in Cryptology - CRYPTO 2015.* 2015, pp. 141–160.

[32] Pierre Karpman and Benjamin Grégoire. “The LITTLUN S-box and the FLY block cipher”. In: *Lightweight Cryptography Workshop.* 2016, pp. 17–18.

[33] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. “Observations on the SIMON Block Cipher Family”. In: *Advances in Cryptology - CRYPTO 2015.* 2015, pp. 161–185.

[34] Xuejia Lai, James L. Massey, and Sean Murphy. “Markov Ciphers and Differential Cryptanalysis”. In: *Advances in Cryptology - EUROCRYPT ’91.* 1991, pp. 17–38.

[35] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology - EUROCRYPT ’93,* 1993, pp. 386–397.

[36] Mitsuru Matsui. “On Correlation Between the Order of S-boxes and the Strength of DES”. In: *Advances in Cryptology - EUROCRYPT ’94.* 1994, pp. 366–375.

[37] Nicky Mouha and Bart Preneel. “Towards Finding Optimal Differential Characteristics for ARX: Application to Salsa20”. In: *IACR Cryptology ePrint Archive* 2013 (2013), p. 328.

[38] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. “Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming”. In: *Information Security and Cryptology, Inscrypt 2011.* 2011, pp. 57–76.

[39] Kaisa Nyberg. “Linear Approximation of Block Ciphers”. In: *Advances in Cryptology - EUROCRYPT ’94.* 1994, pp. 439–444.

[40] Kenji Ohkuma. “Weak Keys of Reduced-Round PRESENT for Linear Cryptanalysis”. In: *Selected Areas in Cryptography, SAC 2009.* 2009, pp. 249–265.

[41] National Institute of Standards and Technology. “197: Advanced encryption standard (AES)”. In: *Federal information processing standards publication* 197.441 (2001), p. 0311.

[42] Marc Stevens. “New Collision Attacks on SHA-1 Based on Optimal Joint Local-Collision Analysis”. In: *Advances in Cryptology - EUROCRYPT 2013.* 2013, pp. 245–261.

[43] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. “Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers”. In: *Advances in Cryptology - ASIACRYPT 2014.* 2014, pp. 158–178.

[44] Huihui Yap, Khoongming Khoo, Axel Poschmann, and Matt Henricksen. “EPCBC - A Block Cipher Suitable for Electronic Product Code Encryption”. In: *Cryptology and Network Security, CANS 2011.* 2011, pp. 76–97.

[45] S. H. Yen, David Hung-Chang Du, and Subbarao Ghanta. “Efficient Algorithms for Extracting the K most Critical Paths in Timing Analysis”. In: *Proceedings of the 26th ACM/IEEE Design Automation Conference, 1989.* 1989, pp. 649–654.

[46] Jun Yin, Chuyan Ma, Lijun Lyu, Jian Song, Guang Zeng, Chuangui Ma, and Fushan Wei. “Improved Cryptanalysis of an ISO Standard Lightweight Block Cipher with Refined MILP Modelling”. In: *Information Security and Cryptology, 2017.* 2017, pp. 404–426.

[47] WenTao Zhang, ZhenZhen Bao, DongDai Lin, Vincent Rijmen, BoHan Yang, and Ingrid Verbauwhede. “RECTANGLE: A Bit-Slice Lightweight Block Cipher Suitable for Multiple Platforms”. In: *Science China Information Sciences* 58.12 (2015), pp. 1–15.

[48] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu. “MILP-based Differential Attack on Round-reduced GIFT”. In: *IACR Cryptology ePrint Archive* 2018 (2018), p. 390.

Publication 4

On Linear Correlation Distributions: More Instructive Examples

Publication Information

Andrey Bogdanov and Philip S. Vejre. *On Linear Correlation Distributions: More Instructive Examples.* In submission. 2018

Contribution

- Main author.

Remarks

None.

On Linear Correlation Distributions: More Instructive Examples

Andrey Bogdanov and Philip S. Vejre

Technical University of Denmark

Abstract. Despite the fact that linear cryptanalysis is one of the most prominent cryptanalysis techniques, our understanding of distinguishers made from linear approximations is still developing. This is especially true for advanced variants that use multiple linear approximations. Using recent models for expressing multivariate linear correlation distributions, this work takes a closer look at these distributions for a range of different ciphers.

Our main finding is that the shapes of the correlation distributions are very diverse, ranging from normal distributions, to mixtures of different types, to highly non-normal distributions. A consequence of this is that the cryptanalyst should be careful when making estimates of distinguisher advantage. Indeed, when considering the χ^2 and the likelihood distinguishers, we find that estimates made solely from ELP can be misleading.

While ELP estimates prove to be accurate for PRESENT, a cipher which has been studied intensely in this context and has approximations with approximately normally distributed correlations, this is not the case for other distribution types. For normal mixtures, we observe that the advantage is highly dependent on the configuration of the components. For 15 rounds of PRIDE the lack of a central component results in a higher advantage, while a dense central component for 11 rounds of GIFT makes distinguishing very difficult. The highly non-normal correlation distribution we observe for 13 rounds of RECTANGLE likewise results in a lower advantage than what we would expect under assumptions of normality. For these ciphers, we observe a discrepancy in advantage compared to a well behaved normal distribution of about 6 bits.

We reaffirm previous work by Hermelin et al., now in a key dependent model, and show that the χ^2 distinguisher is not robust to the introduction of noise, i.e. its advantage decreases if we add uninformative approximations to our distinguishing set. The likelihood distinguisher does not exhibit this behaviour, and its advantage is unaffected. Both of these results demonstrate the importance of closely examining the probability distributions used in a distinguisher, as variation between ciphers can make it hard to rely on e.g. summary statistics.

1 Introduction

New types of attacks on symmetric key primitives, such as block ciphers and hash functions, continue to be developed. At their core, many of these attacks rely on some kind of distinguisher; a function which determines whether the attacker is interacting with the target primitive or a random function. However, for some of these distinguishers, our understanding of exactly how they behave is not perfect. Surprisingly, this is perhaps most true for some of the oldest statistical distinguishers, namely differential [6] and linear distinguishers [22]. While theoretical results on the statistical behaviour of these distinguishers do exist for ideal primitives [15], the behaviour for e.g. practical block cipher designs is not as set in stone. As in example, [3] explores the distributions of differential probabilities for different block ciphers, and find that their distributions deviate from the expected Poisson shape.

This picture only becomes more complicated for advanced variants of these attacks, such as truncated differential attacks and multiple linear attacks. Nevertheless, the power of a differential or linear attack entirely depends on how well the underlying distinguishers perform, and so if we cannot accurately assess this, we also cannot determine how well e.g. a key recovery attack based on these distinguishers will work. It is therefore important to keep refining our understanding of these statistical distinguishers. To this end, this work will take a closer look at the behaviour of linear correlation distributions.

Previous Work

Since the introduction of linear cryptanalysis by Matsui in 1993 [22], a great number of publications have improved and expanded on the idea. Perhaps the most influential extensions are multiple [4, 7] and multidimensional [14, 18, 19] linear cryptanalysis, and both for the one-dimensional case and the multidimensional extensions a lot of work has been done recently to further our understanding of how linear correlation distributions behave [8, 9, 11, 20]. In particular, [12] proposes the *multivariate profiling* model that in principle is able to describe arbitrary (multivariate) distributions of linear correlations over the space of encryption keys, provided that a set of good linear trails is known. Moreover, tools have been developed for finding such trails, e.g. [3, 17], allowing us to efficiently sample from these distributions. In one case, an attack improvement was facilitated by the unusual shape of correlation distributions [13]. This begs the question if other such cases exist, facilitating more powerful attacks, or whether our assessment of current attacks might be flawed in some cases. This motivates us to take a closer look at correlations distributions of various different block ciphers.

Contributions

For multiple and multidimensional linear cryptanalysis, estimates of a particular distinguisher's effectiveness is often made under simplifying assumptions. It is common practice to find a set of good trails for each approximation, and then use the sum of squared correlation contributions of these trails to estimate the expected capacity of the approximations. Assuming a χ^2 shape of the capacity, an estimate of the advantage can then be made. Indeed, this is a reasonable approach if the distribution we are distinguishing is well behaved, i.e. it is close to a multivariate normal distribution. This is the case for PRESENT [10], which has been analysed in many different works, and many models have been proposed that give good results in this case using ELP as the main estimator [8, 9, 20]. Thus, while the normal case is well understood, little work has been done on other types of correlation distributions.

When ELP Succeeds and When it Fails In this work, we consider examples of correlation distributions for which ELP is a not a good estimate of distinguishing power. We utilise the multivariate profiling model of [12], in which the shape of the correlation distribution is estimated directly from known linear trails of the cipher using no assumptions about e.g. the key-schedule. In order to find such trails, we use the recently published tool from [17]. This tool allows us to efficiently find a large number of trails for each approximation, hopefully resulting in accurate estimates, and also allows us to sample from the corresponding signal distributions, which directly enables the use of the multivariate profiling model.

Under this model the correlation distributions we observe for different ciphers designs vary widely in shape, and as a result, an estimate of advantage based on ELP alone can deviate significantly from an advantage observed using this model. Indeed, this deviation boils down to the exact shape of the correlation distribution compared to the shape of an ideal distribution. Since the ideal distribution is concentrated around zero, distributions that have a very low density around zero are easier to distinguish, while distributions that have a high density around zero can be difficult to distinguish. This is despite the fact that such distributions can have components far away from zero, giving them a high ELP.

We explore this discrepancy for two commonly discussed distinguishers, the χ^2 and the likelihood distinguishers, for a range of different types of distributions. Our findings are summarised in Table 1 and described in more detail below.

- **Mixtures of Normal Distributions** We first consider correlation distributions of various ciphers which can be described by the mixture model of [13]. We demonstrate that if the distribution has a single component, estimating the advantage solely from ELP is a sound approach. This is for example the case for PRESENT [10]. We apply the profiling model using about 2^{52} trails per approximation over 22 rounds, and the observed advantage closely matches the estimate.

  For mixtures with several components, the picture is more complex. Indeed, while the distributions for the ciphers PRIDE [2] and `GIFT` [5] can both be described as mixtures, the simple advantage estimate deviates in different directions from the one we observe. In the case of PRIDE, the approximations have a single dominant trail over 15 rounds, causing the distribution to have a low density around zero. For GIFT however, several strong trails exist over 11 rounds, creating a central component in the correlation distribution, decreasing the advantage. For PRIDE, the observed advantage is about 6 bits higher than estimated, while for GIFT it is about 6 bits lower.

Moreover, while it was shown in [13] that asymmetric mixture distributions can improve distinguishing advantage, we show that this is not always the case. Indeed, GIFT exhibits both symmetric and asymmetric distributions with the same ELP values, but the distinguisher works no better in the asymmetric case due to the persistence of the central component. These observations demonstrate the importance of carefully inspecting the correlation distributions used in an attack when deriving attack complexities, as small variations in these can have a large impact.

- **Non-Normal Distributions** While the examples explored above have components which are individually normal, it is not guaranteed that the correlation distributions of a cipher can be expressed in this model. As an example of this, we show that the block cipher RECTANGLE [28] exhibits very non-normal correlation distributions, even in the one-dimensional case. In particular, despite finding 6 million trails over 13 rounds of the cipher, the distributions we

Table 1: Summary of cases studied in this work. We considered how distinguishing advantage estimated assuming a simple normal shape compares to the advantage estimated using the more advanced multivariate profiling (MP) model of [12]. The number of trails used for the MP model is shown. In most cases, there is a significant difference in the two estimates, caused by the more complex shape of the correlation distributions.

Cipher	**Rounds**	**Trails**	**Distribution Type**	**Symmetry**	**Advantage Estimate**	
					ELP	MP
Known results						
PRESENT Section 3.1	22	2^{53}	Normal	Symmetric	5.85	5.85
New results						
PRIDE Section 3.2	15	1	Normal mixture	Symmetric	2.49	8.83
GIFT-64 Section 3.2	11	34		Symmetric	7.25	0.53
GIFT-64 Section 3.3	11	34		Asymmetric	7.25	0.52
RECTANGLE Section 4	13	2^{23}	Non-normal	Symmetric	7.98	1.85

observe have much longer tails than a normal distribution and show a strong dependence structure in the case of multiple approximations. We rule out the key-schedule as the source of this non-normality, but it remains an open question exactly what about the structure of RECTANGLE results in this deviation from normality.

One consequence of this complicated distribution shape is that using the likelihood distinguisher is impractical. Moreover, for the χ^2 distinguisher we observe a much lower advantage than the estimate obtained from the ELP, again due to the distribution being dense around zero. We leave it as an open question whether the deviation from normality itself can be used as a distinguisher.

- **Distinguishing with Uninformative Approximations** Finally, we consider the performance of the two distinguishers in a scenario where a proportion of approximations behave like noise. The two distinguishers were previously compared in [18] for multidimensional linear cryptanalysis, where it was shown that for a fixed capacity, the χ^2 advantage is inversely proportional to the number of approximations M, while the LLR advantage is inversely proportional to $\log(M)$. However, that work does not consider how the multidimensional probability distribution varies over the key space. Thus, we reconsider this comparison in a key dependent model. To eliminate the factor of distribution shape, we again consider PRESENT, and demonstrate that the χ^2 distinguisher is not *robust*, that is, its distinguishing advantage decreases dramatically when "bad" approximations are added to a set of "good" approximations. For a set of four approximations, adding four noisy approximations decreases the advantage from about 11 bits to about 9 bits. We also show that the likelihood distinguisher does not suffer from this problem, although it can be harder to apply in practice, reaffirming the findings of [18]. This further shows that if a cryptanalyst decides to use the unstable χ^2 distinguisher, she must carefully analyse the approximations used to obtain the optimal advantage.

The rest of this work is structured as follows: Section 2 describes the preliminaries of linear distinguishers. Section 3 considers distinguishing distributions that can be described as normal mixtures, while Section 4 describes non-normal distributions. Finally, Section 5 analyses distinguishing in the presence of uninformative approximations.

2 Linear Distinguishers

Throughout this work, we will consider the challenge of distinguishing block ciphers. We define a *block cipher* as a function

$$\mathcal{E}(x, k) : \mathbb{F}_2^n \times \mathbb{F}_2^\kappa \to \mathbb{F}_2^n.$$

For each key k in the key-space $\mathbb{F}_2^\kappa$, $\mathcal{E}$ is a permutation on $\mathbb{F}_2^n$. As such, $\mathcal{E}$ is a family of 2^κ permutations. To $\mathcal{E}$ we associate an *ideal* version, $\tilde{\mathcal{E}}$. Each instance of $\tilde{\mathcal{E}}$ is also a permutation on $\mathbb{F}_2^n$, but the permutations are chosen uniformly at random from the space of all permutations on $\mathbb{F}_2^n$. Thus, we can view $\tilde{\mathcal{E}}$ as a "perfect" block cipher with a key-space of size $2^n!$.

2.1 Distinguishers in General

Assume now that an attacker is either given or chooses a list of N inputs and is then given the corresponding list of outputs, encrypted with either an instance of $\mathcal{E}$ or an instance of $\tilde{\mathcal{E}}$. Let $\mathcal{T}$ denote the list of input/output pairs. The goal for the attacker is to determine if the ideal or non-ideal block cipher was used. To this end, we define a *distinguisher*:

$$\mathcal{D}(\mathcal{T}) : (\mathbb{F}_2^n)^N \times (\mathbb{F}_2^n)^N \to \{\text{Ideal, Not ideal}\}.$$

The distinguisher simply computes some function of $\mathcal{T}$, and outputs either "Ideal" or "Not ideal". We are mainly interested in two properties of the distinguisher: its *success probability* and its *advantage*. We define the success probability as

$$p_S = \Pr(\mathcal{D}(\mathcal{T}) = \text{Not ideal} \mid \mathcal{E})$$

and the advantage as

$$a = -\log_2 \left(\Pr(\mathcal{D}(\mathcal{T}) = \text{Not ideal} \mid \tilde{\mathcal{E}})\right),$$

where the probabilities are taken over the respective sets of permutations, as well as any other randomness used in the choice of inputs or by the distinguisher. The situation is illustrated in Figure 1. We will often fix the success probability, and consider the resulting advantage, as this value is directly associated with how well a distinguisher can be used as part of a key recovery attack on $\mathcal{E}$. For further details, see e.g. [25].

2.2 Linear Distinguishers

The general idea of a *linear distinguisher* is to find linear relationships between the bits of elements in $\mathcal{T}$ which exhibit a larger correlation than one would expect from an ideal cipher. To this end, we define a linear approximation $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ and associate to it a *linear correlation*:

$$C^k_{(\alpha,\beta)} = 2 \cdot \Pr_{x \in \mathbb{F}_2^n}(\langle \alpha, x \rangle = \langle \beta, \mathcal{E}(x, k) \rangle) - 1,$$

where $\langle \cdot, \cdot \rangle$ denotes the canonical inner product on $\mathbb{F}_2^n$. Note that $C^k_{(\alpha,\beta)}$ is a random variable over the key-space. We will denote the correlation measured for a specific

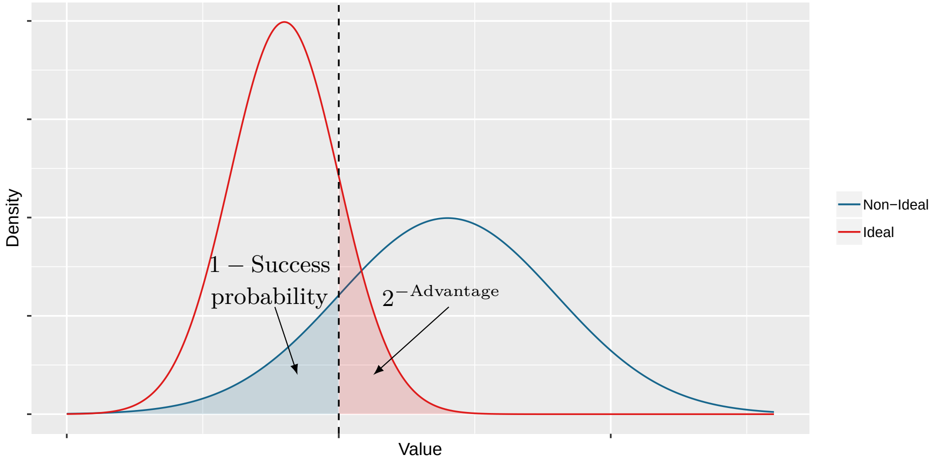


Figure 1: Illustration of success probability and advantage when applying a statistical distinguisher. The threshold value is denoted by the dashed line.

set $\mathcal{T}$ as $C^k_{(\alpha,\beta)}(\mathcal{T})$. Thus, a very simple linear distinguisher could be defined as

$$\mathcal{D}(\mathcal{T}) = \begin{cases} \text{Not ideal} & \text{if } |C^k_{(\alpha,\beta)}(\mathcal{T})| > \tau \\ \text{Ideal} & \text{otherwise} \end{cases},$$

for some predetermined value of the *threshold value* τ. While this distinguisher is essentially that proposed originally by Matsui [22], a variety of other linear distinguishers have been proposed. Amongst these is the natural extension of *multiple linear cryptanalysis* [4, 7] (and the related *multidimensional linear cryptanalysis* [14, 18, 19]). Let us denote a vector of M approximations by $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$, and its associated vector of simultaneous correlations by $C^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$. The idea of multiple linear cryptanalysis is then that the distinguisher can make a better decision based on $C^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ as opposed to just the correlation of a single approximation.

Common to all of these methods (and analogously any statistical distinguisher) is that in order to determine the success probability and advantage, we need to know the distribution of $C^k_{(\alpha,\beta)}$ over the key-space for both $\mathcal{E}$ and $\tilde{\mathcal{E}}$.

2.3 Correlation Distributions

The Ideal Case For the ideal cipher $\tilde{\mathcal{E}}$, [15] shows that $C^k_{(\alpha,\beta)} \sim \mathcal{N}(0, 2^{-n})$, i.e. a normal distribution with mean zero and variance 2^{-n}. For a specific set of input/output pairs $\mathcal{T}$, with inputs sampled randomly with replacement, we moreover have that $C^k_{(\alpha,\beta)}(\mathcal{T}) \sim \mathcal{N}(0, 2^{-n} + N^{-1})$. However, in the case of multiple linear approximations, it seems difficult to determine the exact shape of the joint linear correlations. Indeed, even if statistical dependencies of correlations are likely to occur, see e.g. the discussion in [24], we take the same approach as in [12] and make the simplifying assumption of statistical independence in the case of linearly independent approximations. Thus, in this case, we have that $C^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\mathcal{T}) \sim \mathcal{N}_M(\mathbf{0}, \mathrm{diag}(2^{-n} + N^{-1}))$.

The Non-Ideal Case For the cipher $\mathcal{E}$ useful results are known in the case of *key-alternating* ciphers, i.e. ciphers of the form

$$\mathcal{E} = f_r \circ \cdots \circ f_1,$$

where each *round function* f_i has the form

$$f_i(x, k_i) = g_i(x) \oplus k_i,$$

and the k_i's are round keys derived deterministically from k. Usually, a pre-whitening key k_0 is added to the input. For this construction, we define a *linear trail* as the tuple $U = (u_0, \ldots, u_r)$, and its associated *correlation contribution* as

$$C^k_U = \prod_{i=0}^{r} C_{u_i,u_{i+1}}(f_i)$$

i.e. the product of the correlations of each round function. Then the following result can the be shown.

Theorem 1 ([16]). *Let $\bar{k}$ denote the concatenation of a key-alternating cipher's round keys for the encryption key k. Then the linear correlation of an approximation (α, β) of the cipher can be calculated as*

$$C^k_{(\alpha,\beta)} = \sum_U (-1)^{s_U \oplus \langle U, \bar{k}\rangle} |C^k_U|,$$

where the sum is over trails $U = (\alpha, \ldots, \beta)$, s_U is the sign bit of U, and $|C^k_U|$ is independent of k.

While it is infeasible to calculate the full sum in Theorem 1, [11] proposed to split the sum into a set of *signal trails* $\mathcal{S}$ and a set of *noise trails*. The sum for a specific set $\mathcal{T}$ (with inputs sampled randomly with replacement) is then approximated by

$$C^k_{(\alpha,\beta)}(\mathcal{T}) \approx \left(\sum_{U \in \mathcal{S}} (-1)^{s_U \oplus \langle U, \bar{k}\rangle} |C^k_U| \right) + \mathcal{N}(0, 2^{-n} + N^{-1}). \tag{1}$$

A generalisation of this result to the case of multiple linearly independent approximations was given in [12]. In both cases, we can easily sample from $C^k_{(\alpha,\beta)}(\mathcal{T})$, respectively $C^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$, by calculating the above sum for different keys k. However, this requires that a suitable set of signal trails $\mathcal{S}$ is known. The work [17] gives an algorithm both for finding good trails and for efficiently sampling from the signal distribution. We will use this tool in the following to generate correlation distributions and derive the corresponding success probabilities and advantages.

2.4 On ELP and Capacity

While using the above model to derive success probabilities and advantage is somewhat involved, other methods for determining the effectiveness of a linear distinguisher have often been used in the literature, namely using ELP and capacity. Consider Theorem 1: It is known that if the round keys are statistically independent, then

$$\mathrm{E}((C^k_{(\alpha,\beta)})^2) = \sum_U (C^k_U)^2.$$

The sum on the right is usually denoted the *expected linear potential* (ELP). Moreover, under this independence assumption, and by the central limit theorem[1], $C^k_{(\alpha,\beta)} \sim \mathcal{N}(0, ELP)$. Thus, it is very common for works on linear cryptanalysis to find one or more trails of a linear approximation, calculate the ELP from their correlation contributions, and then estimate the strength of the attack from this value. Similarly, for multiple/multidimensional attacks with M approximations, we can define the *capacity* as

$$\mathcal{C}^k = \sum_{i=1}^{M} (C^k_{(\alpha_i,\beta_i)})^2.$$

Under the above independence assumption for correlations of multiple linearly independent linear approximations, the expected value of the capacity is simply the sum of the ELPs, and so this sum is often used as an estimate for the effectiveness of a multiple linear attack. Additionally, several works present results on estimating the variance of capacity for multidimensional distinguishers, in order to get a more precise estimate [8, 9].

As shown in [12], the fact that round keys are not independent in most ciphers can have an impact on the expected capacity, making it deviate from the sum of ELPs. Moreover, if the correlation distributions exhibit statistical dependence, the shape of the capacity distribution might make the expected value, or even the variance, a bad indicator of distinguishing power. Indeed, in Sections 3 and 4 we explore examples of correlation distributions where the sum of ELPs might give a misleading estimate, both positively and negatively. As a benchmark, we will compare to an estimate obtained using a "well behaved" correlation distribution, namely the distribution

$$C^k_B \sim \mathcal{N}_M(\mathbf{0}, \mathrm{diag}(ELP_1, \ldots, ELP_M)).$$

[1] Assuming that the difference in correlation contributions is not too large.

Indeed, this would be the best case for using the above method of summing ELPs as a way to estimate distinguishing power, and as such represents an optimal scenario for the cryptanalyst in terms of easy analysis. We note that when making comparisons, we still apply the signal/noise decomposition of [11], i.e. the estimate for each ELP value becomes

$$ELP_i = \left(\sum_{U \in \mathcal{S}_i} (C_U^k)^2 \right) + 2^{-n} + 1/N.$$

2.5 The χ^2 and LLR Distinguishers

Given the (potentially multivariate) correlation distributions of $\mathcal{E}$ and $\tilde{\mathcal{E}}$, we can formulate many different distinguishers. In this work, we will consider two distinguishers that have been proposed in the literature, namely the χ^2 distinguisher and the likelihood distinguisher.

The χ^2 Distinguisher The χ^2 distinguisher builds on the χ^2 method for hypothesis testing, and is perhaps the most used linear distinguisher. In the general case of M approximations, and for a given threshold value τ, the distinguisher is defined as

$$\mathcal{D}_{\chi^2}(\mathcal{T}) = \begin{cases} \text{Not ideal} & \text{if } N \sum_{i=1}^{M} (C_{(\alpha_i,\beta_i)}^k(\mathcal{T}))^2 > \tau \\ \text{Ideal} & \text{otherwise} \end{cases}.$$

Note that for a general multivariate distribution of $C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k$, the sum computed by the distinguisher is *not* χ^2 distributed, as the marginals need not be independent nor normally distributed. This makes it difficult to derive closed form expressions of the success probability and advantage. Nevertheless, as long as we can sample from $C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k$, we can estimate these values.

It is interesting to note that the distribution of the sum computed by $\mathcal{D}_{\chi^2}$ only depends on the distribution of $C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k$. That is, if we fix τ, changing the distribution of $C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k$ for $\mathcal{E}$ only affects the success probability, while changing the distribution for $\tilde{\mathcal{E}}$ only affects the advantage. On the other hand, the χ^2 distinguisher does not need any prior knowledge of these two distributions to work, making it very easily to apply in practice.

The Likelihood Distinguisher The likelihood distinguisher is more involved than the χ^2 distinguisher, in that we need to a priori have a good estimate of the two distributions we want to distinguish. For a given threshold value τ, it is defined as

$$\mathcal{D}_{LR}(\mathcal{T}) = \begin{cases} \text{Not ideal} & \text{if } \frac{\Pr\left(C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k(\mathcal{T}) | \mathcal{E}\right)}{\Pr\left(C_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}^k(\mathcal{T}) | \tilde{\mathcal{E}}\right)} > \tau \\ \text{Ideal} & \text{otherwise} \end{cases}.$$

For practical reasons, the logarithm of the likelihood ratio is often used, and so this distinguisher is also known as the log-likelihood ratio (LLR). If we have perfect knowledge of the two correlation distributions, this distinguisher is theoretically optimal. However, as previously pointed out in [18], it is not often used in practice due to the difficulty of calculating the required probabilities.

Using the models described in Section 2.3, one could e.g. use kernel density estimates to estimate $\mathcal{D}_{LR}$. However, this might become impractical for higher dimensions. In [13] it was proposed to use a mixture model, when this seems reasonable, and it is demonstrated that in this case the likelihood distinguisher performs better than the χ^2 distinguisher.

We finally note that the likelihood distinguisher does not have the behaviour described for the χ^2 distinguisher above. Namely, if we change either of the correlation distributions both the success probability and the advantage are likely to change. In other words, the distribution of $\mathcal{D}_{LR}$ depends both on the ideal and the non-ideal distribution simultaneously.

3 Distinguishing Normal Mixtures

In the following, we will compare advantage estimates obtained using the sum of ELP approach described in Section 2.4 to advantages observed under the model of [12] described in Section 2.3, for both the χ^2 and LLR distinguishers. We will first consider the case of distinguishing correlation distributions whose signal can be described by the model given in [13]. There, the signal is described as a normal mixture, i.e. a weighted sum of normal distributions. For a mixture with ℓ components, weights λ_i, mean vectors $\boldsymbol{\mu}_i$, and covariance matrices $\boldsymbol{\Sigma}_i$, $i = 1, \ldots, \ell$, the probability density of the distribution is given by, under the condition that $\sum \lambda_i = 1$,

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \lambda_i \phi_M(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where ϕ_M is the PDF of the M-variate normal distribution. Combining this with Equation 1 we can e.g. calculate the probabilities required for the likelihood distinguisher. Note however, that while algorithms for determining the mixture parameters do exist, a better result will likely be obtained if the cryptanalyst defines them herself, making the use of this model rather time consuming.

Testing Methodology In the following, we have used the tool published in [17] to search for linear trails as well as to sample from the signal distributions. For each set of approximations we consider, we have sampled 50 000 signal correlations using this tool, and then approximated the linear correlation by using the signal/noise model of [11]. Then, when calculating the advantage of each distinguisher, we split the data set randomly into two equal parts, i.e. a training set and a testing set. We use the training set to calculate the threshold value required for an 85% success

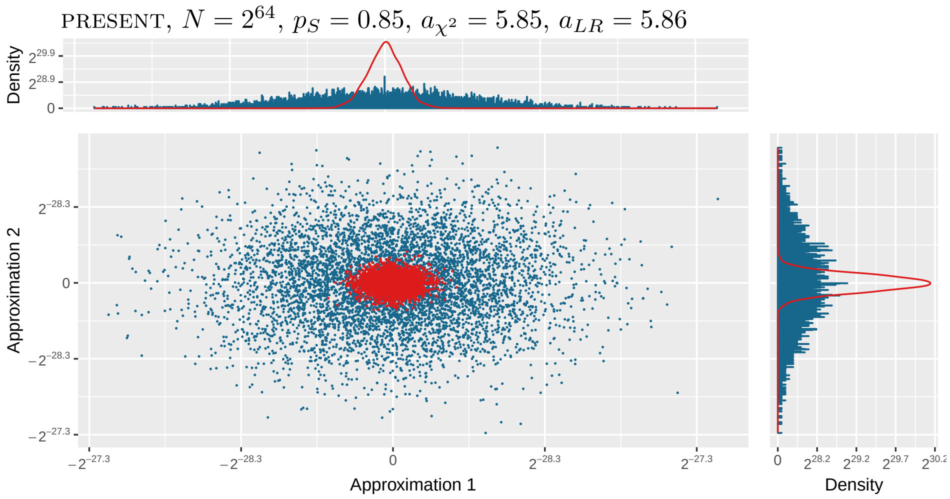


Figure 2: In blue, the joint correlation distribution of two linear approximations over 22 rounds of PRESENT. The distribution for an ideal cipher is shown in red. The advantage for the two distinguishers is given.

probability, and then estimate the advantage using the testing set. We repeat this process 1000 times to get a stable estimate of the advantage. In the following, we only consider pairs of approximations, in order to make visualisation easier. However, our observations generalise to higher dimensions.

3.1 The Normal Case: One Component

We first consider the case of a single mixture component, $\ell = 1$, i.e. a simple multivariate normal distribution. This case can be found for the 64-bit block cipher PRESENT [10]. PRESENT has been the target of many linear cryptanalysis publications due to the extreme linear hull effect it exhibits. It is also this presence of many equally good linear trails for each approximation that ensures that the correlation distributions are quite close to normal. We consider the two approximations

$$(\alpha_1, \beta_1) = (\texttt{0x000000000e000000}, \texttt{0x8000000080008000}),$$
$$(\alpha_2, \beta_2) = (\texttt{0x0000000000e00000}, \texttt{0x8000000080008000}),$$

over 22 rounds of the cipher. Using the tool of [17], we find about $2^{52.3}$ trails for each approximation, for a total ELP of $2^{-58.40}$ each. Applying the model of Equation 1 with $N = 2^{64}$, we obtain the correlation distribution shown in Figure 2. As previously observed in the literature, this distribution is quite close to normal [1, 12]. In this case, the two distinguishers perform equally well, both obtaining an advantage of

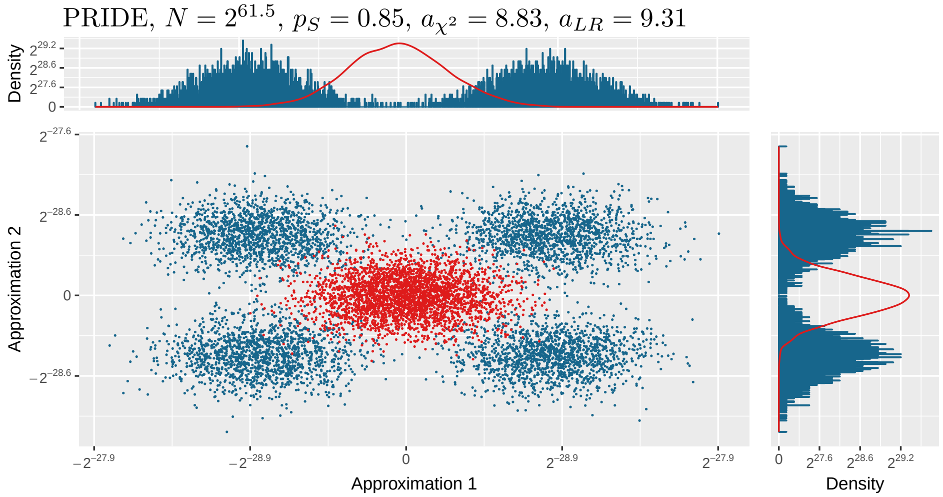


Figure 3: In blue, the joint correlation distribution of two linear approximations over 15 rounds of PRIDE. The distribution for an ideal cipher is shown in red. The advantage for the two distinguishers is given. A bivariate normal distribution with the same ELP values has a *lower* distinguishing advantage of 2.49 bits.

about 5.85 bits. Moreover, if we consider the idealised distribution C_B^k described in Section 2.4, we obtain a similar advantage. Thus, in this case, we can expect the sum of ELPs to give us a quite good estimate of our distinguishing power. Next, we will see that this is not always the case.

3.2 Several Components: ELP Can Be Misleading

We now consider two cases with $\ell > 1$ that demonstrate that the sum of ELPs is not necessarily a good indicator of distinguishing power. We first consider the 64-bit block cipher PRIDE [2]. While several works in differential cryptanalysis of PRIDE have been published [21, 26, 27, 29], there seems to be few results on linear cryptanalysis. Here, we consider the two approximations

$$(\alpha_1, \beta_1) = (\texttt{0x0000000000000100}, \texttt{0x0100000001000100}),$$
$$(\alpha_2, \beta_2) = (\texttt{0x0000000000000001}, \texttt{0x0001000000010001}),$$

over 15 rounds of the cipher. In this case, we were only able to find a single trail for each approximation, having squared correlation contribution 2^{-58}. This matches the initial analysis in [2]. Thus, each approximation has a single dominant trail, and as observed in [13], if the ELP is large enough, the distribution will therefore have

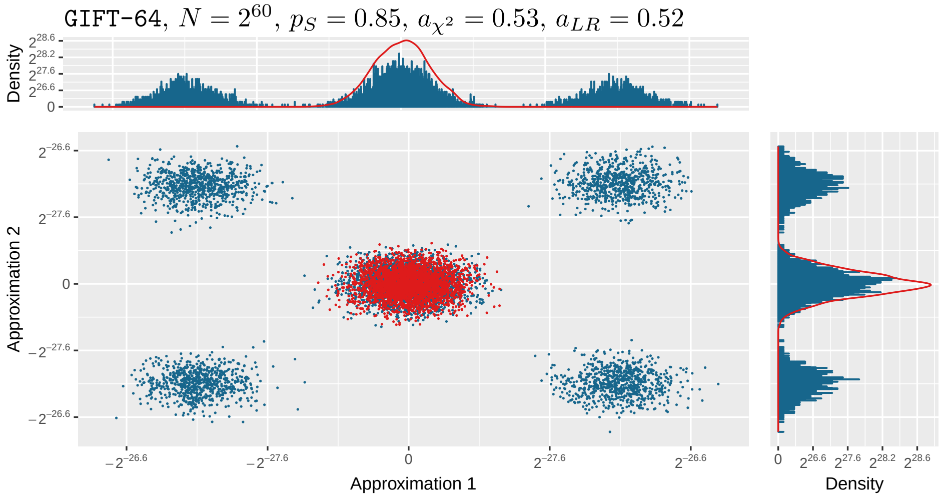


Figure 4: In blue, the joint correlation distribution of two linear approximations over 11 rounds of GIFT-64. The distribution for an ideal cipher is shown in red. The advantage for the two distinguishers is given. A bivariate normal distribution with the same ELP values has a *higher* distinguishing advantage of 7.25 bits.

more than one component. Indeed, the correlation distribution is shown in Figure 3, where we can clearly observe four distinct components.

If we set $N = 2^{61.5}$, and apply the χ^2 distinguisher, we obtain an advantage of 8.83 bits. A small improvement, i.e. about 0.5 bits, is obtained by applying the likelihood distinguisher, matching the observations made for DES in [13]. The interesting observation here, however, is that if we consider the benchmark distribution C_B^k with the same ELP values, we only obtain an advantage of 2.49 bits. Thus, if the cryptanalyst only considered the ELP when searching for suitable approximations, she would drastically underestimate her attack power, and maybe wrongly conclude that no linear attack on this number of rounds could be mounted. This could for example result in a designer choosing to use an insufficient number of rounds in the believe that no linear attack will be successful.

While the above case shows that one can easily underestimate the power of a distinguisher, we now consider the opposite case. The 64-bit block cipher GIFT-64 is a recent design that revisits the ideas of PRESENT, while trying to improve efficiency and security. For GIFT-64, we consider the two approximations

$$(\alpha_1, \beta_1) = (\texttt{0xe000090000600900}, \texttt{0x4014008210410028}),$$
$$(\alpha_2, \beta_2) = (\texttt{0xd000090000600900}, \texttt{0x4014008210400028}),$$

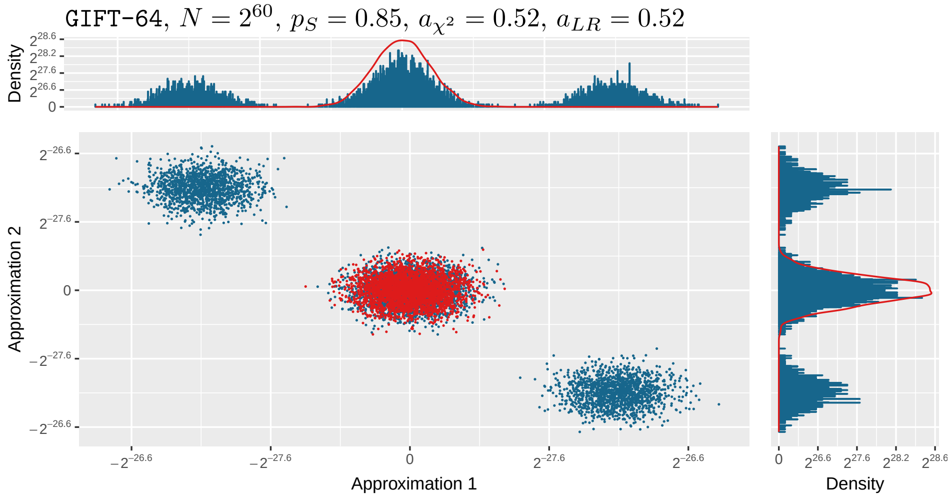


Figure 5: In blue, the joint correlation distribution of two linear approximations over 11 rounds of GIFT. The distribution for an ideal cipher is shown in red. The advantage for the two distinguishers is given. Compared to the distribution in Figure 4 we here see asymmetries, but the advantage is no better.

over 11 rounds of the cipher. For each of these approximations, we find 34 trails for a total ELP of approximately $2^{-55.0}$. Setting $= 2^{60}$, we obtain the very interesting correlation distribution shown in Figure 4.

While we do observe four components with non-zero mean vectors, there is also a fifth components with mean roughly zero. This could be explained by a pair of dominant trails which interact constructively and destructively. By inspecting the marginal distributions, it is evident that a large part of the probability density is contained in this central component. Indeed, when we estimate the mixture parameters, we find that the weight, λ_5, for this component is 0.5, i.e. it constitutes half the density. As a consequence, both distinguishers have a very low advantage of about 0.5 bits. This is compared to the benchmark distribution C_B^k for which one would get a 7.25 bit advantage. On the other hand, the shape of the distribution also implies that half the keys are weak keys, i.e. if we lower our success probability to 0.5, we obtain a perfect distinguisher. These two examples clearly show how important it is to consider the actual shape of the correlation distribution, as opposed to simply making estimates based on the sum of ELPs.

3.3 Asymmetry Cannot Save You

In [13], it was observed that DES exhibits asymmetrical correlation distributions, and the term *asymmetry factor* was proposed, defined as $\ell/2^M$. For DES it was observed that a smaller asymmetry factor increased the distinguishing power, and it was conjectured that this might be the case in general. However, we now demonstrate that this is not the case.

We again consider 11 rounds of `GIFT-64`, but this time we consider the two approximations

$$\begin{aligned}(\alpha_1, \beta_1) &= (\texttt{0x010000e00900e000}, \texttt{0x4401200011040002}),\\ (\alpha_2, \beta_2) &= (\texttt{0x010000e00900e000}, \texttt{0x4401200011048002}).\end{aligned}$$

These approximations have the same ELP as those described previously, but their input/output masks are chosen in a special way: their input masks are identical, and their output masks only differ in bit 15. `GIFT` is designed such that only part of the state is affected by the key addition step, and in particular only a constant is added to bit 15. In this case, the constant is 1, and thus these two approximations will likely always have the same absolute correlation but with opposite signs. Indeed, this is the case, as can be observed in Figure 5. Nevertheless, these approximations still have a central component, and even though we have an asymmetry factor of 3/4, we obtain no improvement in advantage over the symmetric case, for either of the distinguishers. Essentially, how well our distinguisher works is entirely determined by the central component. This further demonstrates that direct inspection of the correlation distributions is necessary in order to evaluate distinguishing power.

4 Distinguishing Non-Normal Distributions

The examples we have shown so far have all had correlation distributions that could be expressed as normal mixtures. Nonetheless, this type of shape is not a given. In particular, dependence between round-keys can influence the shape of the distribution, as demonstrated in both [1] and [12]. Moreover, it could happen that the structure of the linear trails is such that the resulting distribution is not normal. We will investigate such a case next. Since it it quite difficult to apply the likelihood distinguisher in this case, we only consider the χ^2 distinguisher in the following.

4.1 The Case of RECTANGLE

RECTANGLE [28] is yet another 64-bit block cipher inspired by the PRESENT design, but with a focus on bit-slicing friendly components. Over 13 rounds of the cipher, we found 6 242 685 trails for each of the two approximations

$$\begin{aligned}(\alpha_1, \beta_1) &= (\texttt{0x00000d000000a000}, \texttt{0x0000021000600084}),\\ (\alpha_2, \beta_2) &= (\texttt{0x00000d000000a000}, \texttt{0x0000021000610004}),\end{aligned}$$

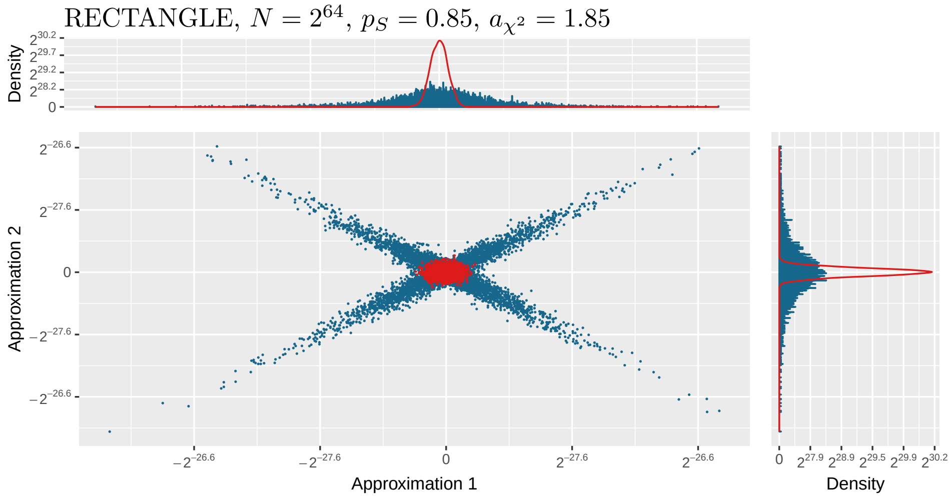


Figure 6: In blue, the joint correlation distribution of two linear approximations over 13 rounds of RECTANGLE. The distribution for an ideal cipher is shown in red. The advantage for the χ^2 distinguisher is given. The distribution cannot be described by a normal mixture. A bivariate normal distribution with the same ELP values has a *higher* distinguishing advantage of 7.98 bits.

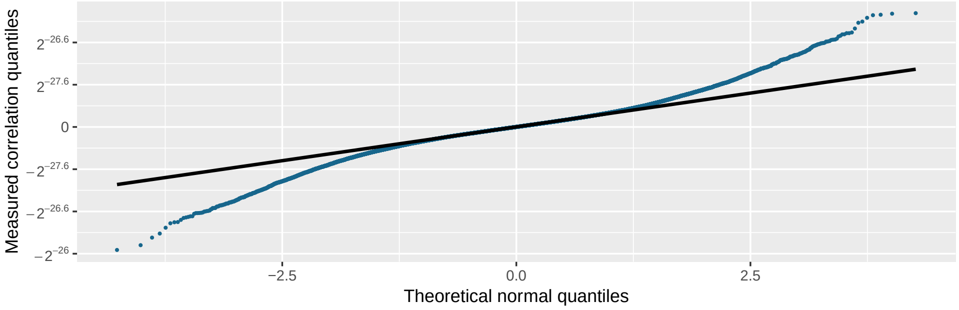


Figure 7: Theoretical normal quantiles compared to the observed correlation distribution quantiles for the block cipher RECTANGLE and the approximation $(\texttt{0x00000d000000a000}, \texttt{0x0000021000600084})$. We observe a clear deviation from normality.

for a total ELP of $2^{-58.01}$. The correlation distribution of these two approximations is shown in Figure 6 for $N = 2^{64}$. It is immediately clear that the joint distribution is not a normal mixture. Moreover, if we consider the marginal distributions, these are not normal either, as demonstrated by the quantile-quantile plot in Figure 7. Indeed, the observed distributions have much longer tails than a normal distribution with the same variance would have. Additionally, it is interesting to observe that this shape seems to be inherent to the structure of the round function of RECTANGLE, as replacing the key-schedule with e.g. that of PRESENT does not affect the shape of the distributions.

At a first glance, this extreme deviation from normality might suggest that these distributions would be easier to distinguish, but this is not the case. Indeed, the χ^2 distinguisher only obtains an advantage of 1.85 bits with $N = 2^{64}$, whereas distinguishing the benchmark distribution C_B^k would result in an advantage of 7.98 bits. The reason for this is similar to the case of `GIFT` discussed in Section 3.2, namely that the majority of the distribution density is concentrated close to zero. As such, only very few keys actually exhibit a large correlation.

While this non-normal distribution shape observed for RECTANGLE may not be beneficial in the case of single-key distinguishers, it would be interesting to explore in a multi-key setting. Indeed, if we could observe several points from the correlation distribution, a simple test of normality might work very well as a distinguisher. In any case, further exploration of these non-normal correlation distributions is warranted.

5 Distinguishing with Uninformative Approximations

In the previous sections we saw that it is essential to closely examine the correlation distributions of the approximations one intends to use for distinguishing. In the following, we eliminate the question of distribution shape by only considering distributions that are approximately multivariate normal. However, we now consider the composition of the set of approximations we use for distinguishing. Specifically, we consider the case of distinguishing when using a set of $M + \tilde{M}$ approximations, but where a subset of $\tilde{M}$ of the approximations are *uninformative*, i.e. their correlations are distributed as $\mathcal{N}(0, 2^{-n})$, independently from the other approximations.

This case could for example occur if a cryptanalyst chooses to include extra approximations in their attack with the hope of an added advantage, assuming that it has no adverse effect on the final attack complexity, but without carefully analysing the correlation distributions of these. Indeed, the general thinking here would be that adding these approximations cannot worsen the attack. This case was considered in [8, Theorem 4], where an expression of the variance of the capacity was given. Another case would be if the cryptanalyst wants to use the multidimensional model, in which case she needs to include all approximations in a full subspace, some of which are likely to have low correlations. We note that a recent work proposes a model in which some of these bad approximations can be discarded [23].

In this scenario, it is natural to expect that a distinguisher on all $M + \tilde{M}$ ap-

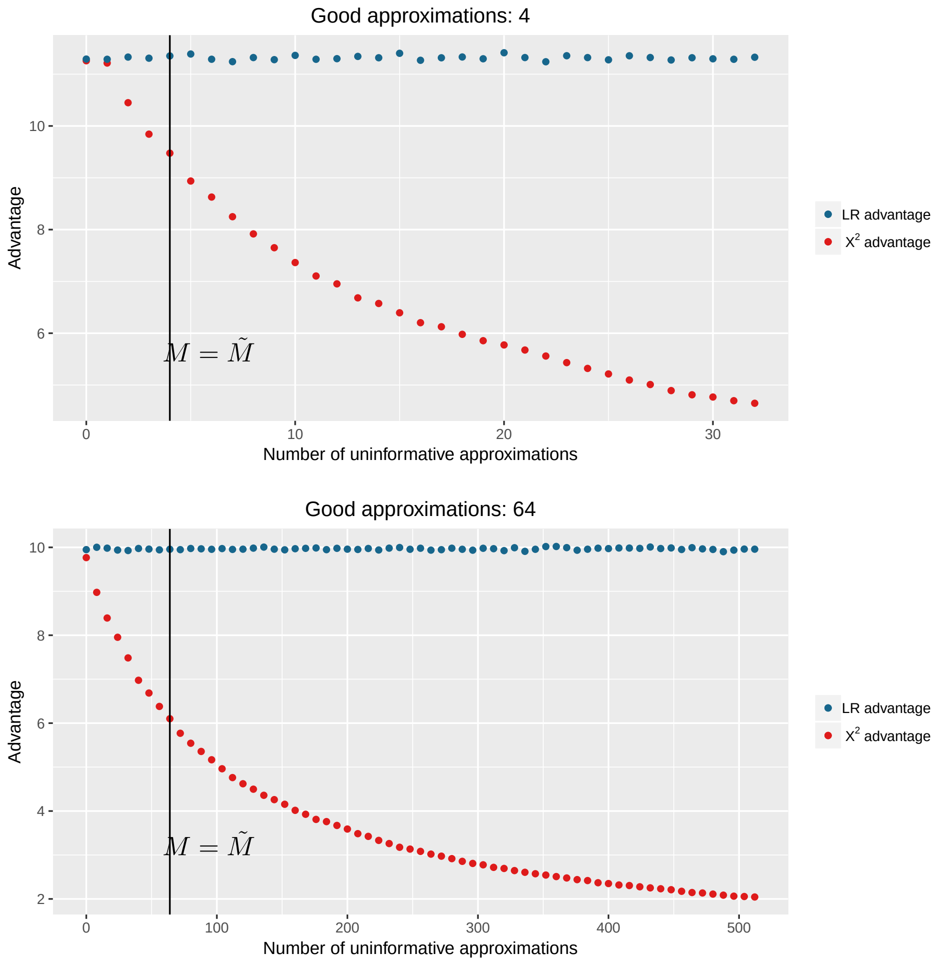


Figure 8: Distinguisher advantage as a function of uninformative approximations. The top plot uses a set of 4 good approximations ($N = 2^{63.5}$), whereas the bottom plot uses a set of 64 good approximations ($N = 2^{59.5}$). The advantage of the χ^2 distinguisher decreases as a function of the ratio of bad approximations, whereas the likelihood distinguisher is stable.

proximations performs no better than on the M "good" approximations, as the $\tilde{M}$ approximations add no information. On the other hand, we would also hope that adding these uninformative approximations does not have an adverse effect on our distinguisher, that is, the distinguisher is *stable*. Note that for multidimensional linear cryptanalysis, the previous work [18] shows that for a fixed capacity, the χ^2

advantage is inversely proportional to M, while the LLR advantage is only inversely proportional to $\log(M)$. However, the analysis made there assumed that the multidimensional probabilities were largely independent of the encryption key. In the following, we investigate whether this result also holds when the key dependence is taken into account.

5.1 Distinguisher Stability

As mentioned above, we will eliminate the variable of distribution shape by considering a joint correlation distribution which is close to normal. As demonstrated in Section 3.1, this is true for PRESENT. As our set of good approximations, we choose subsets of the approximations used in [12] to mount attacks on 26 and 27 rounds of the cipher. This set consists of 135 linearly independent approximations. We consider two subsets of these approximations, i.e. for $M = 4$ ($\mathcal{C}^k = 2^{-56.87}$) and $M = 64$ ($\mathcal{C}^k = 2^{-54.18}$), over 21 rounds of the cipher. We then investigate how the advantage behaves when we add uninformative approximations to these sets.

Advantage measurements are performed as in Section 3, i.e. we calculate the threshold required for an 85% success probability using a training set, and then calculate the advantage using a testing set, averaging the result over 100 repetitions. For both sets of approximations we consider up to $\tilde{M} = 8 \cdot M$ uninformative approximations. The result is shown in Figure 8. Interestingly, we observe that the χ^2 advantage declines quite rapidly when we increase the number of bad approximations. Moreover, comparing the two plots, the rate of decline seems to roughly be a function of the fraction of uninformative approximations. Thus, if M is relatively small even a quite low number $\tilde{M}$ of bad approximations can have a drastic negative impact on the advantage.

This observation is not so surprising in light of the discussion in Section 2.5; when adding uninformative approximations to the distributions for $\mathcal{E}$ and $\tilde{\mathcal{E}}$, the distinguishing distributions change independently of each other, and the larger the proportion of uninformative approximations get, the more they will resemble each other. For the likelihood distinguisher however, the ratio used for distinguishing stays constant, and so the distinguisher is unaffected, as can be seen in Figure 8. Thus, if it is possible to obtain a good estimate for the likelihood probabilities, e.g. when the correlation distributions are normal mixtures, this seems like a more robust choice for the cryptanalyst.

6 Conclusions

In this work, we have taken a closer look at different types of correlation distributions of multiple linear approximations, and our ability to distinguish these from the correlation distribution of an ideal cipher. We considered two types of distinguishers, the χ^2 distinguisher and the likelihood distinguisher, and how well the ELPs of a distribution can be used to predict the advantage of these distinguishers. Throughout, we have used the multivariate profiling model of [12] to estimate correlation distributions.

We first considered correlation distributions which can be described using the normal mixture model of [13]. We found that if the mixture has one component, i.e. it is a multivariate normal distribution, the ELP values gives a good estimate of advantage. However, we examined two examples where this is not the case: for the cipher PRIDE, whose correlation distribution has more than one component, we obtain a significantly larger advantage than expected from the ELP values alone due to the shape of the distribution. On the other hand, for the block cipher GIFT, we also observed multiple components, but the shape of the distribution is such that the advantage is adversely affected.

Considering correlation distributions that do not fit in the mixture model, we find that RECTANGLE exhibits distributions that are highly non-normal. As for GIFT, this has the effect of decreasing the distinguisher advantage compared to a normal distribution with the same expected capacity. We conclude that the cryptanalyst should closely examine the shape of the correlations distributions instead of relying on summary statistics in order to estimate distinguishing advantage.

Lastly, we observed that the two distinguishers behave differently if noisy approximations are added to the distinguishing set. In particular, the advantage of the χ^2 distinguisher decreases when the ratio of “good” to “bad” approximations decreases, but we found that the advantage of the more complex likelihood distinguisher is stable in this regard. We therefore urge the cryptanalyst to be careful when using the χ^2 distinguisher.

References

[1] Mohamed Ahmed Abdelraheem, Martin Ågren, Peter Beelen, and Gregor Leander. “On the Distribution of Linear Biases: Three Instructive Examples”. In: *Advances in Cryptology - CRYPTO 2012*. 2012, pp. 50–67.

[2] Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin. “Block Ciphers - Focus on the Linear Layer (feat. PRIDE)”. In: *Advances in Cryptology, CRYPTO 2014*. 2014, pp. 57–76.

[3] Ralph Ankele and Stefan Kölbl. “Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis”. In: *Selected Areas in Cryptography, SAC 2018*. 2018.

[4] Thomas Baignères, Pascal Junod, and Serge Vaudenay. “How Far Can We Go Beyond Linear Cryptanalysis?” In: *Advances in Cryptology - ASIACRYPT 2004*. 2004, pp. 432–450.

[5] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. “GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption”. In: *Cryptographic Hardware and Embedded Systems, CHES 2017*. 2017, pp. 321–345.

[6] Eli Biham and Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In: *Advances in Cryptology - CRYPTO '90*. 1990, pp. 2–21.

[7] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. "On Multiple Linear Approximations". In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 1–22.

[8] Céline Blondeau and Kaisa Nyberg. "Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis". In: *IACR Transactions on Symmetric Cryptology* 2016.2 (2016), pp. 162–191.

[9] Céline Blondeau and Kaisa Nyberg. "Joint Data and key Distribution of Simple, Multiple, and Multidimensional Linear Cryptanalysis Test Statistic and its Impact to Data Complexity". In: *Designs, Codes and Cryptography* 82.1-2 (2017), pp. 319–349.

[10] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. "PRESENT: An Ultra-Lightweight Block Cipher". In: *Cryptographic Hardware and Embedded Systems, CHES 2007*. 2007, pp. 450–466.

[11] Andrey Bogdanov and Elmar Tischhauser. "On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2". In: *Fast Software Encryption, FSE 2013*. 2013, pp. 19–38.

[12] Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. "Multivariate Profiling of Hulls for Linear Cryptanalysis". In: *IACR Transactions on Symmetric Cryptology* 2018.1 (2018), pp. 101–125.

[13] Andrey Bogdanov and Philip S. Vejre. "Linear Cryptanalysis of DES with Asymmetries". In: *Advances in Cryptology - ASIACRYPT 2017*. 2017, pp. 187–216.

[14] Joo Yeon Cho, Miia Hermelin, and Kaisa Nyberg. "A New Technique for Multidimensional Linear Cryptanalysis with Applications on Reduced Round Serpent". In: *Information Security and Cryptology, ICISC 2008*. 2008, pp. 383–398.

[15] Joan Daemen and Vincent Rijmen. "Probability Distributions of Correlation and Differentials in Block Ciphers". In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 221–242.

[16] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[17] Mathias Hall-Andersen and Philip S. Vejre. "Generating Graphs Packed with Paths: Estimation of Linear Approximations and Differentials". In: *IACR Transactions on Symmetric Cryptology* 2018.3 (2018), pp. 265–289.

[18] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. "Multidimensional Extension of Matsui's Algorithm 2". In: *Fast Software Encryption, FSE 2009*. 2009, pp. 209–227.

[19] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. “Multidimensional Linear Cryptanalysis of Reduced Round Serpent”. In: *Information Security and Privacy, ACISP 2008*. 2008, pp. 203–215.

[20] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. “Capacity and Data Complexity in Multidimensional Linear Attack”. In: *Advances in Cryptology - CRYPTO 2015*. 2015, pp. 141–160.

[21] Virginie Lallemand and Shahram Rasoolzadeh. “Differential Cryptanalysis of 18-Round PRIDE”. In: *Progress in Cryptology - INDOCRYPT 2017*. 2017, pp. 126–146.

[22] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology - EUROCRYPT ’93*. 1993, pp. 386–397.

[23] Kaisa Nyberg. “Affine Linear Cryptanalysis”. In: *Cryptography and Communications* (2018), pp. 1–11.

[24] Kaisa Nyberg. “Statistical and Linear Independence of Binary Random Variables”. In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 432.

[25] Ali Aydin Selçuk and Ali Biçak. “On Probability of Success in Linear and Differential Cryptanalysis”. In: *Security in Communication Networks, SCN 2002*. 2002, pp. 174–185.

[26] Cihangir Tezcan, Galip Oral Okan, Asuman Senol, Erol Dogan, Furkan Yücebas, and Nazife Baykal. “Differential Attacks on Lightweight Block Ciphers PRESENT, PRIDE, and RECTANGLE Revisited”. In: *Lightweight Cryptography for Security and Privacy, LightSec 2016*. 2016, pp. 18–32.

[27] Qianqian Yang, Lei Hu, Siwei Sun, Kexin Qiao, Ling Song, Jinyong Shan, and Xiaoshuang Ma. “Improved Differential Analysis of Block Cipher PRIDE”. In: *Information Security Practice and Experience, ISPEC 2015*. 2015, pp. 209–219.

[28] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. “RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms”. In: *SCIENCE CHINA Information Sciences* 58.12 (2015), pp. 1–15.

[29] Jingyuan Zhao, Xiaoyun Wang, Meiqin Wang, and Xiaoyang Dong. “Differential Analysis on Block Cipher PRIDE”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 525.