Higher-Order Cryptanalysis of Block Ciphers

Jakobsen, Thomas

Publication date:
1999

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

Higher-Order Cryptanalysis of Block Ciphers

Ph.D. thesis

Thomas Jakobsen

Department of Mathematics
Technical University of Denmark
1999

Abstract

The theme in this thesis is design and analysis of block ciphers. Specifically, new attacks are described that successfully break cryptosystems in which the ciphertext is expressible as evaluations of some low-degree polynomial in the plaintext with a low but non-negligible probability. The attacks are particularly efficient against certain ciphers that are provably secure against differential and linear cryptanalysis.

Several proposed ciphers from the cryptographic literature are either broken entirely by the new approach or shown to have theoretical weaknesses.

Also multiple links from the theory of error-correcting codes to the design of block ciphers are demonstrated and results from coding theory are successfully applied via this connection to the cryptographical setting.

A new, efficient decoding algorithm for Reed-Muller codes that go beyond half the minimum distance is described. Aside from its error-correcting applications, the algorithm is also useful for analysing bit-oriented block ciphers.

Resumé

Denne afhandling omhandler design og analyse af blockchifre. Specielt beskrives nye angreb, der kan bryde kryptosystemer, hvori chifferteksten kan udtrykkes som evalueringer af et polynomium af lav grad i klarteksten med en lille, men ikke-negligibel sandsynlighed. Angrebene er specielt effektive mod visse typer blokchifre som er beviseligt sikre mod differentiel og lineær kryptoanalyse.

Adskillige systemer, der er fremsat i den kryptografiske litteratur, brydes enten fuldstændigt v.h.a. de nye angreb eller det vises, at de har teoretiske svagheder.

Der demonstreres flere sammenhænge mellem blokchiffer-design og teorien om fejlkorrigerende koder og resultater fra kodningsteori udnyttes via disse forbindelser i den kryptografiske opsætning.

Herudover beskrives en ny effektiv, afkodningsalgoritme til Reed-Muller koder, som går over den halve minimumsafstand. Udover anvendelser indenfor fejlkorrektion kan algoritmen med fordel benyttes ved analyse af bit-orienterede blockchifre.

Preface

According to the over 1500 years old Kama Sutra by Vatsayana, cryptology – or "the art of understanding writing in cipher, and the writing of words in a peculiar way" – is the 44th of 64 arts, so-called yogas, that should be known and practised by both men and women. This thesis represents my attempts at mastering the 44th yoga.

The dissertation is submitted in partial fulfillment of the requirements for the Ph.D. degree at the Department of Mathematics, Technical University of Denmark. It is a collection of research papers based on results found in the period of my Ph.D. studies from February 1, 1996 to March 31, 1999 [1] with updated remarks and comments.

The research was funded by a Danish government grant which also supported a three and a half months stay in Cambridge, England at the Isaac Newton Institute of Mathematical Sciences with the Coding, Cryptology, and Computer Security Programme during February–May, 1996.

Most of the presented results deal with a new type of cryptanalysis of block ciphers. The probabilistic version of the attack demonstrates the existence of some quite interesting connections between cryptography and coding theory; in fact it was made possible only by recent progress in the area of decoding. It shows that previous design strategies for block ciphers are not always sufficient for obtaining high security. I hope that the results will contribute to the art and science of block cipher design and push the field of secret-key systems a bit closer towards a more complete understanding of the security concept.

I am greatly indebted to my supervisor Tom Høholdt who has taught me many valuable skills during the years and provided me with lots of intellectual stimuli. He is an extraordinary guide in the mathematical landscape. It is a pleasure to thank also Lars Knudsen, Agnes Heydtmann, and several other colleges with whom I have enjoyed many nice discussions.

Last but certainly not least, warm thanks go to my wife Karin and daughter Matilde for their incredible love, help, and extraordinary patience, to all my friends and family, and especially to my parents for their never-ending support and belief in what I do.

Thomas Jakobsen, Lyngby, March 31, 1999

[1] This period included absence during half a year's leave.

Contents

Chapter 1

Introduction

"We say again deliberately that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve." - Edgar Allan Poe

In his article from the middle of the 19th century, Poe was treating simple alphabetical substitution ciphers [23] and as such his claim was correct [8]. Intriguingly, correctness of the more general claim that any modern (and practical) cipher is breakable is still not disproved today since no-one has come up with a provably secure, yet practical cryptosystem. The fact that there are no published results even hinting at a proof of computational security not involving some unproven assumptions shows that there is still a lot of hard work ahead in trying to establish the foundations of cryptography. Much of the difficulty lies in the fact that these foundations are strongly linked to deep, complexity theoretical questions like whether P equals NP. As such, we may be a long way from proving anything in an absolute sense.

Today's notion of a strong block cipher is one that withstands all known attacks and several years of attempts to break it. Surely, this does not constitute mathematical proof, it just shows that breaking the cipher is at least a hard, intellectual challenge. Examples of well-known classes of attacks are the differential cryptanalysis of Biham and Shamir [1] and the linear cryptanalysis of Matsui [14]. These methods are now well understood and it is known how to build block ciphers that are provably security against them [17]. This thesis develops a new kind of attack on block ciphers. It is particularly useful for attacking some of the ciphers that are constructed to be secure against linear and differential cryptanalysis. Like its predecessors, the attack exploits certain weaknesses in the round function of iterative block ciphers.

The new attack deals with ciphers in which the ciphertext is expressible as evaluations of unknown, low-degree polynomials of the plaintext, either always or with some small but non-negligible probability. Intuition strongly suggests that such ciphers are weak but until recently there were no efficient means of actually exploiting such probabilistic, low-degree relations. In fact, it is quite

easy to show that attacking such a cipher corresponds to decoding of a certain Reed-Solomon code in a setting with a very low information rate and a correspondingly high error rate. A feat that has not been possible to do efficiently until the recent introduction of Sudan's algorithm [24].

Other links to error-correcting codes will be pursued in the thesis. For example, the existence of an algorithm for decoding Reed-Muller codes beyond half the minimum distance leads to another, more general attack. There are constructive results from coding theory, too. For instance, it is possible to build round functions that are immune to linear cryptanalysis by using Carlitz-Uchiyama's bound for exponential sums (known from coding theory) and results related to the distances and covering radii found in higher-order Reed-Muller codes also tell us how to construct ciphers for which it is hard to find nonlinear approximations.

The following provides an overview of the chapters.

Chapter 2 describes the general cryptographic model, gives a classification of attacks, and presents some block cipher preliminaries. Previously known attacks on block ciphers are also mentioned here together with a brief account of the new interpolation attacks.

A summary of the collected research papers is given in Chapter 3 together with updates and new results. The introductions from each article are not repeated in this overview. Instead a short description of each paper is given and new results are discussed.

The chapters 4–8 contain the research papers which all treat methods for the cryptanalysis of block ciphers. There is a development through the chapters towards more powerful attacks from generalized linear cryptanalysis to the interpolation attack, its probabilistic version and other variants.

The final Chapter 9 is an overview of the main results with conclusions and ideas for further research.

Chapter 2

Preliminaries

This chapter describes the cryptographic terminology and explains the mathematical model. We also give the usual classification of attacks into groups each with their own assumptions. Finally, the basic subjects of construction and analysis of block ciphers are treated together with a demonstration of some links to coding theory. Most of the explanations are kept at an informal and summary level, leaving the more detailed definitions to the articles in later chapters.

2.1 The Cryptographic Model

Cryptology is the composition of cryptography and cryptanalysis. Put briefly, *cryptography* is the science of constructing systems that by transformation of messages enables a transmitter to send confidential and authenticated information to a receiver over an otherwise insecure channel. Opposed to this, *cryptanalysis* is the science of trying, not minding the original intent to keep contents secret and authentic, to extract information from encrypted messages or to forge messages. Successful attempts of cryptanalysis are often called *attacks*. It is sometimes possible to develop attacks since no practical systems have ever been proven to be secure in an absolute sense, as previously mentioned.

The two contrasting fields, however, are intimately related sides of the same coin and should always be considered together. Indeed, construction of secure systems calls for knowledge in both areas since the currently best way of putting the security of a system to the test is by subjecting it to every known and previously unknown attack that one can come up with, and hope that they all fail.

The transformation of the original message is called *encryption* and the corresponding inverse transformation *decryption*. Together, the transforming algorithms are also known as a *cryptosystem* or a *cipher*. The original message is called the *plaintext* and the encrypted message is the *ciphertext*. Sometimes they are considered together as a plaintext/ciphertext-pair (*p/c-pair*). Usually, encryption is carried out under influence of a so-called *key* which is a piece

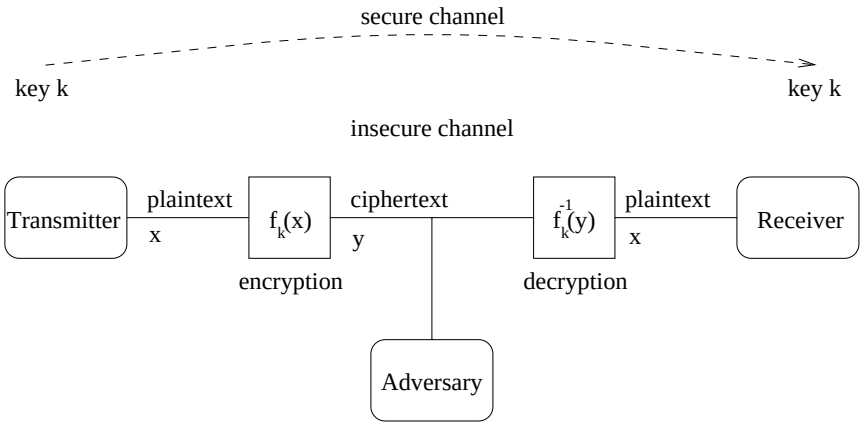

Figure 2.1: The cryptographic model.

of information that is usually secret. In a *secret-key* (or *symmetric*) system (opposed to public-key or asymmetric systems) the key is the same for both transmitter and receiver. For a meaningful decryption to exist, the encryption function must of course be injective, that is no two different plaintexts can have the same ciphertext.

The model is illustrated in Figure 2.1. Here we see that there is a third party involved, the adversary. The goal is getting the transmitter and the receiver to communicate securely even when the adversary has access to the communication channel. The figure illustrates how the secret key is distributed via some secure channel, e.g. a courier or even another encrypted channel. For a more thorough treatment of the cryptographic model, consult, e.g. Stinson [23] or Menezes *et al.* [16].

This thesis considers attacks on iterated block ciphers, and as such it deals only with secrecy issues in symmetric cryptosystems. Consequently, in the remainder we focus solely on the issues involved herein.

2.2 A Classification of Attacks

To further pinpoint and formalize the roles of cryptographer and cryptanalyst, Kerckhoff made the now common assumption that every detail of the cryptosystem is known to the adversary *except for the secret key*. This important assumption is known as *Kerckhoff's principle* and is quite realistic in distributed systems with several users and periodic key changes.

Generally, the goal for the adversary is to find any piece of previously unknown information about the cryptosystem; usually plaintext corresponding to some ciphertext or even the entire secret key. Under some definitions, a cipher is considered broken even if it is possible to guess correctly just one bit of secret

information with probability bounded away from $\frac{1}{2}$.

In addition to public information about the cryptosystem, an attacker might have access to varying amounts of plaintext and/or ciphertext. The following – more or less self-explanatory – classification of attack scenarios is useful:

- **Ciphertext-only**: Here, the adversary has access to some amount of ciphertext obtained, e.g., by passive eavesdropping on the communication link. By using the statistics of the underlying language or the possible messages, the system is then broken, and the plaintext or the key is obtained. The classical alphabet substitution [23] is an example of a cryptosystem which falls for a ciphertext-only attack. The ciphertext-only attack is the attack requiring the least number of assumptions about the adversary.

- **Known-plaintext**: In a known-plaintext attack, the adversary has obtained access to some amount of plaintext and the corresponding ciphertext. By analyzing this she hopes to be able to compute the secret key or further, previously unknown, plaintexts.

- **Chosen-plaintext**: As in the known-plaintext attack, the adversary has some amount of p/c-pairs, only here she gets to choose the values of the plaintext herself.

- **Chosen-ciphertext**: Similar to the chosen-plaintext attack, in this scenario the adversary gets to choose some fixed amount of ciphertext herself and then obtains the corresponding plaintext.

- **Adaptive attacks** allow the adversary to make new queries to the cryptosystem after having analyzed previously obtained p/c-pairs. All the attacks mentioned above have corresponding adaptive versions.

The above scenarios are increasingly restrictive in the assumptions made regarding the non-mathematical capability and behaviour of the adversary. For instance, in a chosen-ciphertext attack the adversary must obtain the plaintext corresponding to some ciphertext - this can be done only if there is some kind of access (direct or indirect) to the algorithm performing the decryption.

Most attacks are known-plaintext attacks or chosen-plaintext. It is often possible to turn a known-plaintext attack into a ciphertext-only attack by utilizing the statistics of the plaintexts (a common trick is making use of the observation that characters that are ascii-encoded into bytes have their most significant bit set to zero).

For a modern cipher to be considered secure, it must withstand all the above-mentioned types of attacks. Here, however, we shall adopt an even stronger notion of security. We will say that a cipher is *indistinguishable from random permutations* or just *indistinguishable* if it is impossible to decide with probability different from $1/2$ whether a given set of pairs consists of p/c-pairs or pairs describing a random permutation. This also agrees with the common perception that a cipher should be "random looking". We will call such a distinguishing method a *discriminator* and it will be defined more rigorously later.

It is easy to see that a cipher which is secure in this sense is also going to be secure against all the above-mentioned classes of attacks since success in any of these implies the existence of a discriminator. Indeed, the above attacks are almost examples of discriminators: Present to the attack algorithm a set which contains either p/c-pairs or random pairs. If the attack succeeds in finding the key or some other piece of information, then obviously the set does not contain random pairs (them having no structure) and vice versa. Of course, this holds true only when there is enough plaintext/ciphertext for the key to be uniquely determined - the number of p/c-pairs has to exceed the so-called *unicity distance* (which, however, is usually quite small compared to the number of possible p/c-pairs).

This more restrictive definition of security may seem unnecessary hard - one may ask what harm can possible be done if somebody is able to tell whether a piece of ciphertext has a chance of belonging to some plaintext - assuming we already know that some plaintext was encrypted in the first place. But actually, for most block ciphers, the existence of a discriminator also implies the existence of attacks that can recover hitherto unknown plaintext or even the entire key. This will be demonstrated in Section 2.4. In other words, the notion of an indistinguishable cipher is actually in some cases quite similar to that of, e.g., a cipher secure against known-plaintext attacks.

Since the notion of indistinguishability has a simple mathematical definition, is stronger than the previous notions of security, and is easier to work with, we will use it here. To recap, given a secret-key cipher, a discriminator is then an algorithm that can distinguish a set of p/c-pairs (from that cipher) from a set of random pairs. Note that this definition does not make any sense for public-key cryptosystems where everybody can encrypt by definition (and therefore trivially decide whether some given plaintext corresponds to some given ciphertext).

Mathematically, the cryptographical problem of constructing a secure system is now the following: Construct a family of time-efficient functions (cryptosystems) $f_k : G \to G$ for which there exists no discriminator of low complexity (for appropriate definitions of efficient and low complexity). An example of a well-known cryptosystem for which this does not hold is DES where it is possible using simple linear relations to distinguish about 2^{40} p/c-pairs from random pairs [14].

Again, when talking computational security as opposed to unconditional or information-theoretical security, it is important to stress that nobody, until now, have published a proof of security not relying on some unproven assumptions. For most block ciphers or secret-key systems even reductions to other hard problems (e.g., NP completeness, factoring, or discrete logarithms), as used in public-key cryptography, are not known. Presently, the best procedure that one can use when constructing secure cryptosystems is collecting evidence in the form of attacks that fail at cryptanalyzing the cipher in question — and to experiment with being clever in general.

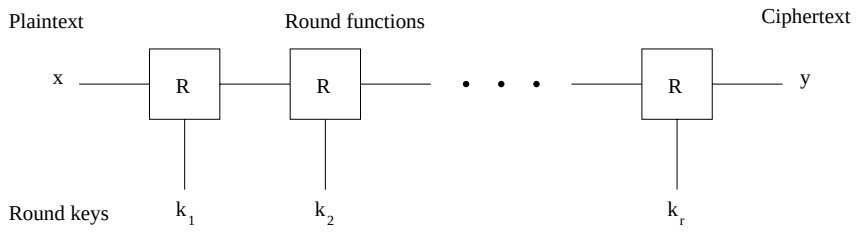

Figure 2.2: Block cipher model.

2.3 Iterated Block Ciphers

A block cipher is a memoryless cryptosystem in which a keyed and (usually) cryptographically weak function, called the round function, is iterated a number of times in the hope that the resulting function is cryptographically strong. The following serves as a more formal description.

Definition 2.1 *Let $R_i : G \to G$ describe a family of bijections called the* round function *from some domain G onto itself indexed by a value $i \in K$, where K is a domain called the* round key space. *Let r be a positive integer. Define the* key space *by K^r, and let the* key *be the vector defined by $k = (k_1, k_2, \ldots, k_r) \in K^r$. Define $f_k : G \to G$ as the composition of r instances of R_i using the components of k as indices. More precisely,*

$$f_k = R_{k_r} \circ R_{k_{r-1}} \circ \cdots \circ R_{k_1}.$$

Then f_k is called an (iterated) block cipher *with r* rounds *and* round function *R_i. The vector k is called the* key *and its components are the* round keys. *If $|G| = 2^n$ we say that the* block length *is n. (The block length is simply the number of bits required to describe the plaintext.) Analogously we say that the* round key length *is m if $|K| = 2^m$ (and hence that the* key length *of the cipher is rm).*

Figure 2.2 illustrates the model. Note that the definition does not mention anything about security and efficiency of the resulting system. Of course, in this sense any function can be viewed as a degenerate, 1-round block cipher with no key. If an efficient system is desired, of course both R and R^{-1} should be "simple" and easy to compute. The round keys are sometimes derived from a master key containing less bits than all the round keys together, this will not be considered here.

Encryption is carried out by the formula $y = f_k(x)$ where y is then the ciphertext corresponding to the plaintext x, and analogously we have decryption $x = f_k^{-1}(y)$, where

$$f_k^{-1} = R_{k_1}^{-1} \circ R_{k_2}^{-1} \circ \cdots \circ R_{k_r}^{-1}.$$

In practise, the round function is often chosen to be an involution because this makes it easier to compute the decryption function. In this case, the inverse

function $f_k^{-1} = f_k$ is simply calculated by applying the round keys to the encryption function in the reverse order.

2.4 Cryptanalysis of Block Ciphers

There is a known-plaintext attack on block ciphers which always succeeds given enough time, namely that of *exhaustive key search*. The idea is simple: Given some plaintext and corresponding ciphertext simply try out all possible keys and stop searching when the key guess is consistent with the obtained plaintext and ciphertext (i.e., when the actual plaintext matches the ciphertext under encryption with the key guess). A similar ciphertext-only variant of this attack can also be implemented if there is enough information available about the statistics of the messages.

For block ciphers with even moderate key lengths this brute-force attack soon becomes infeasible, however, as the number of computational steps in the attack increases exponentially with the key length. More precisely, the expected workload is 2^{w-1} where w is the number of key bits. The complexity of the exhaustive key search, however, is still useful as a measure to compare with the workload of other attacks since it upper-bounds the expected number of computational steps required to break the cipher in question. The number of encryptions performed in an attack is also subject to an upper bound, namely the number of possible plaintexts (since encryption of every possible plaintext yields a complete dictionary between plaintext and ciphertext).

In this sense, no block cipher is secure against a computationally unbounded adversary. Therefore this is the main idea of computational security: To construct a system so computationally expensive to break that it is in reality impossible to carry out an attack.

It was mentioned in Section 2.2 that a discriminator can often be turned into an algorithm that finds the actual key. Actually, the well-known differential cryptanalysis and linear cryptanalysis may be thought of as attacks relying on the existence of a discriminator. Leaving out the inner workings of the two attacks, they both function by finding certain relations between plaintext and ciphertext that holds with a non-negligible probability.

Differential cryptanalysis (DC) considers pairs of plaintext with a fixed difference and then looks at the difference between the resulting ciphertexts. Such a pair of good input and output differences (i.e., a pair with high probability of transition) easily forms the basis of a discriminator. Simply consider enough plaintext pairs with a certain fixed difference – if the output difference is then the same for enough instances, then the pairs considered come from a cipher and are not random at all.

In linear cryptanalysis (LC), probabilistic linear relations between input and output bits of the cipher are found. If the relation holds with a probability non-negligibly bounded away from 1/2 for p/c-pairs (as would not be the case for random pairs) then it also forms the basis of a discriminator and can be used to mount an attack. The reader is referred to [1] and [14] for in-depth information

about differential and linear cryptanalysis, respectively.

In DC and LC, the discriminator is not used as a black box - it is combined with a more subtle approach to recover the last-round key. But it is possible to use a discriminator more or less as a black box (by accepting a higher complexity). In the following, we describe how one can often extend a distinguishing algorithm — the discriminator — for a block cipher into a fully-fledged attack that can recover the entire key. We borrow notation and terminology from our previous work [10] where the notion of a *statistical attack* was defined.

Assume that an r-round block cipher C is given. First, we define by the *reduced cipher* $\tilde{C}$, the cryptosystem consisting of the first $r-1$ rounds of C. The *augmented cipher* $C'_{\tilde{k}}$ is the original cipher followed by an application of the inverse round function using $\tilde{k}$ as the round key. Additionally, we assume that we have available a discriminator for the reduced cipher and that the augmented cipher in some way acts "more like a random function" than the reduced cipher (at least according to the discriminator). The latter assumption is known as the hypothesis of wrong-key randomization, for further discussions about it consult [6] or [7].

The attack is then a known- or chosen-plaintext attack working by a divide-and-conquer technique that finds one round key after another in the following fashion: A guess of the last-round key is made and the obtained ciphertexts are then decrypted by one round. If, according to the discriminator, the result is seen to originate from a reduced cipher rather than an extended cipher, then the key guess was most probably correct. Otherwise the procedure is repeated until the round key is found. By finding the last-round key, the problem of cryptanalysis is then reduced to breaking the reduced-round variant of the cipher.

The following is a more formal description: Let k_r denote the actual last-round key of C and assume that the cryptanalyst has obtained an amount of p/c-pairs $\{(x_i, y_i)\}_{i=1,\ldots,n}$ related to C. A guess $\tilde{k}$ is then made regarding the last-round key. If the key guess matches the actual last-round key k_r, then $C'_{\tilde{k}} = C'_{k_r} = \tilde{C}$. To see whether the first equality is true (remember, k_r is unknown a priori), we make use of the discriminator. The original ciphertexts are decrypted by one round obtaining the pairs $\{x_i, R^{-1}_{\tilde{k}}(x_i)\}_{i=1,\ldots,n} = \{(x_i, \tilde{y}_i)\}_{i=1,\ldots,n}$. By the hypothesis of wrong-key randomization it is possible to distinguish between the two cases by using the discriminator algorithm. Having found the last-round key, the attack can then be repeated recursively on the reduced-round variant of the cipher thus gradually obtaining the entire key. A model of the statistical attack is illustrated in Figure 2.3 (this model was used by Harpes in [6]) and a pseudocode algorithm is given in Figure 2.4.

If the round-key length is w and the discriminator requires n p/c-pairs in order to work, then attacking one round has average complexity $2^{w-1}c(n)$, where $c(n)$ is the complexity of the discriminator (which is linear in the cases of DC and LC).

The interpolation attack that is explained in the following also fits the definition of a statistical attack since it makes use of a discriminator.

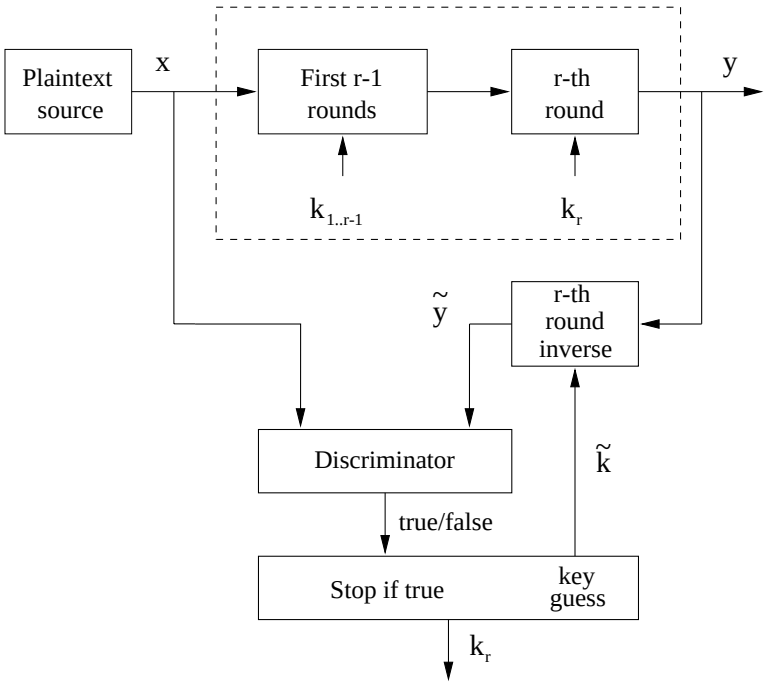

Figure 2.3: A model of the statistical attack.

procedure StatisticalAttack(D, P)

// Input: $D(\cdot)$, a discriminator that returns **true** if the argument set
// describes p/c-pairs from the reduced cipher and **false** otherwise.
// $P = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, a set of p/c-pairs.
// Output: The last-round key k_r or "failure".

for all choices $\tilde{k}$ of last-round key **do**
  for i **from** 1 **to** n **do**
    $\tilde{y}_i := R_{\tilde{k}}^{-1}(y_i)$
  next i
  if $D(\{(x_1, \tilde{y}_1), \ldots, (x_n, \tilde{y}_n)\}) =$ **true then**
    $k_r := \tilde{k}$
    return k_r
  endif
next $\tilde{k}$
return "failure"

Figure 2.4: Pseudocode for the statistical attack.

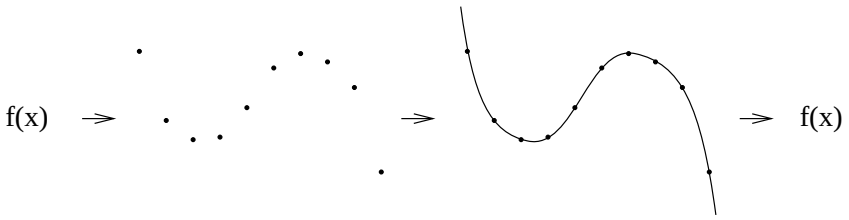

Figure 2.5: Polynomial interpolation as a means of communication.

2.4.1 The Interpolation Attack

A very interesting thing happens if one interprets the output of a cryptosystem as transmissions of certain, long, error-correcting codes over very noisy channels where the choice of codeword depends upon the cipher key: It becomes possible to use decoding algorithms for the purpose of discriminators. If the decoder is good enough to recognize badly damaged codewords, then sometimes it can also be used as a discriminator to find the secret key.

Such an interpretation has been exploited for a while in the area of stream ciphers. Here, the fast correlation attack [15] makes use of efficient decoders for certain error-correcting codes in order to recognize the correct subkey.

In the following we demonstrate the existence of a similar link to the domain of block ciphers. First, let us consider the case where the channel has probability 0 of introducing errors in the transmitted vector, i.e., the transmitted vector and the received vector are identical (for an introduction to coding theory consult [2] or [6]).

Assume that we want to transmit information over the channel by using a Reed-Solomon code [2]. This means, loosely speaking, that we send the information encoded as a degree-d polynomial (the coefficients representing the information). We do it in an indirect fashion, however, by actually sending a number of evaluations of the polynomial (instead of the coefficients). Clearly, if there are more than $d+1$ such evaluation values available and they are not in error, then the polynomial can be reconstructed easily by simple interpolation algorithms like Lagrange's. This situation is shown in Figure 2.5.

If the received information is sent via a new, similar channel also using a Reed-Solomon encoding rule, then we can consider the two serial channels as sub-channels in one large super-channel and the resulting "super"-code will then also be a Reed-Solomon code (due to the fact that one polynomial inserted in another yields a new polynomial). That is, two applications of coding through the channels results in a situation which is equivalent to only one channel and one encoding step. If there is noise, then the resulting error rate is a function of the individual error rates (assuming that everything is independent). This simplification of the channel serialization process is illustrated in Figure 2.6. The process generalizes by induction to serialization of several channels.

The situation resembles closely what happens in a block cipher where the round function is a low-degree polynomial. In this case, if the number of rounds

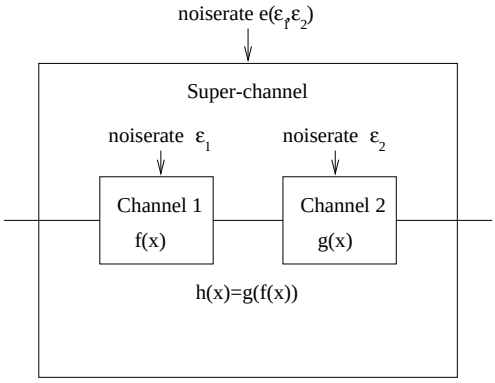

Figure 2.6: Serialization of channels.

Figure 2.7: Polynomial interpolation with noise.

is appropriately low, then the ciphertext of the block cipher can also be expressed as evaluations of a low-degree polynomial of the plaintext. This implies that we can actually interpolate the polynomial and thus obtain the encryption function as a polynomial in its canonical form. Using this polynomial we can then encrypt other values or even decrypt if we reverse the roles of plaintext and ciphertext; in other words the cryptosystem is broken. The polynomial will also be useful as a discriminator since once found, all new p/c-pairs that are intercepted will have to satisfy the polynomial.

The interpolation attack described later takes place in exactly such a setting. Some short-cuts in the form of meet-in-the-middle attacks and optimizations of the interpolation process will also be given.

Now consider the same situation in a setting with a channel that has a nonzero error rate. Figure 2.7 illustrates what happens. Some of the evaluations of the polynomial have been disturbed by errors and are therefore replaced by random values. But it is still possible to recognize the polynomial that was transmitted.

Block cipher	Coding system
cryptosystem	code and channel
round function	subchannel
attack	decoder
key	codeword
plaintext	evaluation points
ciphertext	received word
approximability	error rate
polynomial degree of approximation	code dimension
number of p/c-pairs needed in attack	code length

Table 2.1: Connections between block ciphers and coding systems.

In the block-cipher setting this amounts to saying that there exist fairly accurate approximations of the round function. Here an approximation is a low-degree polynomial that agrees with the round function on a non-negligible fraction of the possible inputs. If there exists an algorithm for finding such an approximation over the whole block cipher, then it can be turned into a discriminator.

Until recently, however, such an algorithm capable of decoding Reed-Solomon codes was not known for high error rates and particularly not when the number of errors exceeded 50% of the number of transmitted values. However, Sudan recently proposed an algorithm that is capable of decoding way beyond half the minimum distance. The algorithm is perfect for settings like the cryptographic one where the codes are very long, the information rate extremely low, and the error rate correspondingly high.

Chapter 6 describes how to use Sudan's algorithm for decoding Reed-Solomon in connection with the cryptanalysis of block ciphers. The algorithm is then generalized by Bezout's theorem to allow the discovery more general relations between plaintext and ciphertext - this falls outside a coding theoretical description, however. In Chapter 7 we develop a decoding algorithm for Reed-Muller codes which can be used in a similar fashion.

Table 2.1 gives an informal dictionary between the two areas, block ciphers and coding systems.

Chapter 3

Overview of Articles

In this chapter we give a descriptive overview of the papers presented in the following chapters. All articles have been retypeset for continuity and typographical homogeneity as have theorems and equations been renumbered to provide unique labeling. Each paper is selfcontained in the sense that its bibliography is included in the corresponding chapter.

The articles have appeared in the following publications (or they are going to):

[J96] Thomas Jakobsen and Carlo Harpes, *Non-Uniformity Measures for Generalized Linear Cryptanalysis and Partitioning Cryptanalysis*, ed. J. Pribyl, Proceedings of Pragocrypt '96, Prague, 1996.

[J97] Thomas Jakobsen and Lars Knudsen, *The Interpolation Attack on Block Ciphers*. Proc. Fast Software Encryption '97, Lecture Notes in Computer Science, vol. 1267, Haifa, Springer, 1997.

[J98a] Thomas Jakobsen, *Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree*. Crypto '98, in Hugo Krawczyk, editor, Lecture Notes in Computer Science 1462, Springer, 1998.

[J98b] Thomas Jakobsen, *Non-Linear Approximations in Block Ciphers*, abstract, Winter School on Coding and Information Theory, 1998, Ebeltoft.

[J99a] Thomas Jakobsen and Lars Knudsen. *Attacks on Block Ciphers of Low Algebraic Degree*. Submitted to Journal of Cryptology, 1999.

[J99b] Agnes Heydtmann and Thomas Jakobsen, *Decoding Reed-Muller Codes Beyond Half the Minimum Distance*. Submitted to Finite Fields 5, Augsburg, 1999.

There are connections to coding theory in several places. The construction of round functions that are resistent to linear cryptanalysis is linked to the study of

exponential sums. For instance, the well-known Carlitz-Uchiyama bound for exponential sums in coding theory tells us how to construct round functions using simple low-degree polynomials that are very resistant to linear cryptanalysis.

Ironically, coding theory also helps us to break ciphers that are constructed in this way. Reed-Solomon and Reed-Muller decoding beyond half the minimum distance makes it possible to cryptanalyze ciphers where the ciphertext is expressible as evaluations of low-degree polynomials with a small but non-negligible probability.

3.1 Bounds on Non-Uniformity Measures for Generalized Linear Cryptanalysis and Partitioning Cryptanalysis

Chapter 4 is a reprint of the paper [J96].

Generalized linear cryptanalysis and partitioning cryptanalysis are both statistical attacks generalizing the linear attack of Matsui [14]. Both of the attacks have been treated by Harpes, Kramer, and Massey in [7] and together with yet another generalization by the author in [9] and in [10].

For the attacks to work, a metric must be used to measure the deviation of a certain estimated probability distribution. In the paper [J96] different measures are described and compared, and various relations between them are shown to exist. This effectively demonstrates some similarities between the various attacks.

3.2 Attacks on Block Ciphers of Low Algebraic Degree

Chapter 5 is a reprint of the paper [J99a]. Portions of this paper were also published in [J97].

To protect a cipher against linear or differential cryptanalysis or their generalizations one can choose a round function with certain properties. An easy way to do this is by using specific simple algebraic functions over finite fields. Then it is not very difficult to prove that the resulting cipher is secure against the attacks.

By using simple algebraic functions, however, the cipher often becomes susceptible to other attacks. The paper [J99a] develops the family of so-called interpolation attacks which work by exploiting polynomial relations between plaintext and ciphertext.

A parallel to the interpolation attack, the higher-order differential attack (HOD) is also presented. Early instances of the HOD attack were described in [11] and [13] but the version here is stronger and more general.

In the paper, several ciphers are shown to be breakable; among these a cipher constructed by Nyberg and Knudsen [17] which was proven to be secure against

differential and linear attacks. The cipher falls completely to an attack using higher-order differentials and slight variants are breakable by the interpolation attack with even quite a high number of rounds.

The paper is a revised journal version of [J97] adding some optimizations and refinements to the interpolation attack. Among the optimizations is a linear-time, constant-memory algorithm to evaluate the obtained polynomial in a specific point. Some preprocessing/memory tradeoffs are also discussed.

3.3 Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree

Chapter 6 is a reprint of the paper [J98a] with some minor typographical corrections.

The interpolation attack works only in a setting where the ciphertext is expressible as a low-degree polynomial in the plaintext with probability 1. One might ask the question, what happens if the output of a cipher is expressible as a low-degree polynomial in the input for only a certain fraction of the inputs, i.e. with a certain probability for a randomly distributed plaintext?

The article [J98a] answers this question for the case where the output is expressible as a low-degree polynomial over a finite field with a certain low but non-negligible probability. The attack which is based on Sudan's algorithm for decoding Reed-Solomon codes may well be thought of as a kind of probabilistic interpolation attack.

Sudan's algorithm can also be used for another cryptographical purpose, namely that of secret sharing with cheaters. Asymptotically, using Sudan's algorithm to reconstruct the secret allows for up towards 100% cheaters in Shamir's well-known threshold secret-sharing scheme.

Theorem 3.1 *Assume that q is a prime power, the function $f : \mathbb{F}_q \to \mathbb{F}_q$ is defined by $f(x) = s + c_1 x + c_2 x^2 + \cdots + c_t x^t$ where t is the so-called threshold value, $c_1, \ldots, c_t \in \mathbb{F}_q$ are random, secret values, $s \in \mathbb{F}_q$ is the secret to be shared, and the n shares are given by $s_i = f(x_i)$ for $1 \leq i \leq n$ where $x_i \neq x_j$, $i \neq j$.*

Assume furthermore, that some fraction of the participants are cheating, i.e., that when asked to present their share they lie and give a wrong value. Then the polynomial f and hence the shared secret value s can be reconstructed in polynomial time by every coalition of m participants with at most εm cheaters, if

$$\varepsilon < 1 - \frac{1}{1+\rho_\kappa} - \frac{\rho_\kappa}{2}\kappa, \text{ where } \rho_\kappa = \left\lfloor \sqrt{\frac{2}{\kappa} - \frac{1}{4}} - \frac{1}{2} \right\rfloor.$$

and $\kappa = \frac{t+1}{m}$.

Note that in the case of coalitions of size t (the original threshold value), the right hand side of the bound becomes zero meaning no cheaters are allowed

(this makes sense since no group of less than t people – truth-speaking or not – should be able to reconstruct the secret).

Proof Follows directly from Theorem 6.7 of the article by translation to the secret sharing setting. □

In other words, the use of Sudan's algorithm in Shamir's secret sharing scheme allows a trade-off between the threshold value and the number of allowed cheaters. The fraction of cheaters is even allowed to exceed $\frac{1}{2}$.

3.4 Decoding Reed-Muller Codes Beyond Half the Minimum Distance

Chapter 7 is a preprint of the paper [J99b].

The probabilistic interpolation attack described in [J98a] only works for ciphers where it is "natural" to regard plaintext and ciphertext as members of a finite field.

For ciphers like the Data Encryption Standard (DES, [8]), there is no natual correspondance to large finite fields. Instead it is useful to regard the bits of the ciphertext as evaluations of multivariate polynomials in the input bits. In this setting one could ask whether there exist simple or low-degree polynomials in the input bits that agree with the actual output bits with a non-neglible probability.

In the article, Sudan's algorithm is generalized to Reed-Muller codes. The resulting decoder can correct errors beyond half the minimum distance when the rate is low. Consequently, the algorithm is ideal for finding low-degree approximations of binary functions in many variables.

One of the steps in the algorithm requires a factorization method for multivariate, boolean polynomials. Since the quotient ring of multivariate, boolean polynomials is not a unique factorization domain, standard algorithms for factoring are not applicable. Through a novel, yet simple approach a factorization algorithm is developed which easily finds all possible factorizations of a given boolean, multivariate polynomial.

3.5 Analysis of S-Boxes and Decoding of Linear Block Codes

Some of the results in Chapter 8 have been presented in [J98b] but have not been published elsewhere. The chapter contains miscellanous results that complement the other chapters.

We demonstrate how to find approximations over the S-boxes of DES using a coding-theoretical interpretation. This is done by decoding and finding low-weight codewords in certain binary, linear codes. Among other techniques, we use the approach of factoring multivariate polynomials described in the above-mentioned article [J99b]. This yields several good approximations.

Chapter 4

Bounds on Non-Uniformity Measures for Generalized Linear Cryptanalysis and Partitioning Cryptanalysis

The following paper was written with Carlo Harpes and originally published in the proceedings of Pragocrypt '96.

Bounds on Non-Uniformity Measures for Generalized Linear Cryptanalysis and Partitioning Cryptanalysis

Thomas Jakobsen[1] Carlo Harpes [2]

Abstract The paper presents a general setting which is used to describe generalized linear and partitioning cryptanalysis. A measure of non-uniformity called imbalance similar to Matsui's bias is defined. Some upper bounds for this measure are presented and used to estimate a cipher's resistance to each of the two attacks. The bounds reveal that there exists a unified measure which reflects the resistance against both attacks and show that the use of almost bent functions can make a cipher immune to the attacks.

Key words: Linear cryptanalysis, partitioning cryptanalysis, statistical attack, block ciphers.

4.1 Introduction

Linear cryptanalysis (LC) was introduced by Matsui [7] in an attempt to break DES [8]. Later, Harpes, Kramer, and Massey generalized linear cryptanalysis by introducing the notion of binary I/O sums [2], and the new attack (GLC) was shown to be more successful in some cases than ordinary linear cryptanalysis. GLC has also been extended beyond the binary case. Harpes made further generalizations in [3] when he introduced the notion of partitioning cryptanalysis (PC) and showed that this is a even more powerful attack. The three mentioned attacks are all known-plaintext or chosen-plaintext attacks on block ciphers and common to all three of them is the statistical analysis of plaintext/ciphertext pairs (p/c-pairs), which in the successful case leads to knowledge of the key of the last (or first) round.

In this work, we develop bounds which can be used to estimate the success probability for each of the attacks. The paper is organized as follows. In the next section we will introduce the notions of imbalance and I/O differences and some other preliminaries. In Section 3, we introduce the statistical attack, which can be used to model GLC and PC. The following sections are devoted to GLC and PC, respectively. Here we present imbalance bounds which are useful for

[1]Department of Mathematics, Building 303, Technical University of Denmark, DK-2800 Lyngby, Denmark. Email: T.Jakobsen@mat.dtu.dk.

[2]Signal and Information Processing Laboratory, Swiss Federal Institute of Technology Zurich, CH-8092 Zürich, Switzerland. Email: harpes@isi.ee.ethz.ch

estimating a cipher's security. It turns out that the Fourier transform is a useful tool in this context. Finally, we draw some conclusions and make suggestions for further work.

4.2 Preliminaries

In this section, we introduce the notation and the definitions used throughout the paper. We consider an iterative block cipher with round function consisting of a keyed abelian group operation "+" at the entry followed by an unkeyed bijection φ (see Figure 4.1). Let $(G, +)$ denote the employed abelian group of order N with neutral element 0. Denote by $-a$ the additive inverse of a, and by $a - b$ the expression $a + (-b)$.

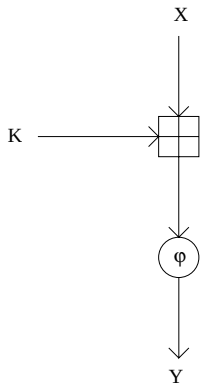

Figure 4.1: **The considered round function.**

More formally, we are looking at an iterative block cipher of r rounds with N possible input values and a round function given by $R_k(x) = \varphi(x + k)$ where $\varphi : G \to G$ is some bijection, and $x, k \in G$.

As in [2], the capital letters X, K, and Y denote random variables describing the input, the key, and the output of the cipher, respectively. The corresponding lower-case letters denote instances of these random variables. With superscripts on letters, e.g. $Y^{(i)}$ and $k^{(i)}$, we indicate that expressions belong to a certain round (i in this case); thus $Y^{(i)}$ is the output from the i-th round and the input to round $i + 1$. We will assume that all subkeys are independent and uniformly distributed. Although subkeys are often generated by some key schedule, the assumption can usually be made without loss of generality.

Let the decomposition of G into cyclic groups be given by $G = \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \times \cdots \times \mathbb{Z}_{N_g}$ (i.e., the decomposition consists of g cyclic groups and $N = N_1 \cdot N_2 \cdot \cdots \cdot N_g$). By subscript on elements from G we denote the corresponding elements of the cyclic subgroups, e.g., $x = (x_1, x_2, \ldots, x_g)$, where $x \in G$ and $x_j \in \mathbb{Z}_{N_j}$.

By c^* we denote the complex conjugate of the complex number c and by M a positive integer that divides N. By $\mu = e^{\frac{2\pi i}{M}}$ and $\nu = e^{\frac{2\pi i}{N}}$ we denote an M-th and an N-th primitive root of unity in $\mathbb{C}$, respectively (here i is the imaginary unit). Furthermore, by $\chi^w : G \to \mathbb{C}$ we denote the function given by $\chi^w(x) = \nu^{<w,x>}$, where $< w, x >= \sum_{j=1}^{g} w_j x_j T_j \bmod N$, for $w, x \in G$ and $T_j = N/N_j$. In other words, $\chi^w(x)$ represents a homomorphism from $(G, +)$ into the multiplicative group of complex numbers of magnitude 1.

By $\hat{G}$ we denote a character group of G. For the purpose of this paper, let the elements of this character group be the functions in the set $\hat{G} = \{\chi^w : w \in G\}$. Note, that $\chi(x)\chi(y) = \chi(x+y)$ and $\chi(-x) = \chi^*(x)$ for all $\chi \in \hat{G}$ and $x, y \in G$. To denote addition and subtraction modulo M, we use $\oplus_M$ and $\ominus_M$, respectively, and $\bullet$ represents the bitwise scalar product.

4.2.1 The Fourier Transform

Definition 4.1 Fourier transform. *Let G be a finite abelian group of order N with character group $\hat{G} = \{\chi^w : w \in G\}$. The Fourier transform of the complex function $\psi : G \to \mathbb{C}$ is the function $\mathcal{F}\{\psi\} : G \to \mathbb{C}$ defined by*

$$\mathcal{F}\{\psi\}(w) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{x \in G} \psi(x) \cdot \chi^{-w}(x)$$

for all $w \in G$.

For a cyclic group (addition over $\mathbb{Z}_N$), the above definition coincides with the usual definition of the discrete Fourier transform, i.e., $\chi^{-w}(x) = \mu^{-w}$, and for the "xor"-group (addition over $\mathbb{Z}_2^n$) it is simply the Walsh-Hadamard transform.

We will often use the well-known Parseval's identity.

Theorem 4.2 Parseval's identity. *For any function $\psi : G \to \mathbb{C}$, we have*

$$\frac{1}{N} \sum_{x \in G} |\psi(x)|^2 = \sum_{w \in G} |\mathcal{F}\{\psi\}(w)|^2 ,$$

where G is an abelian group of order N and $\mathcal{F}\{\phi\}$ is the Fourier transform of ϕ over G.

Another useful property of the Fourier transform is the following.

Theorem 4.3 Inverse Fourier transform. *Any function $\psi : G \to \mathbb{C}$ can be expressed in a unique way as a weighted sum of elements from the character group $\hat{G}$, namely*

$$\psi = \sum_{w \in G} \mathcal{F}\{\psi\}(w) \chi^w,$$

where χ^w is the character corresponding to w.

4.2.2 Imbalance

Definition 4.4 *A function $f : G \to V$ is said to be* balanced *when it takes on each value in V for the same number of arguments.*

Thus, if X is uniformly distributed over G, then f is balanced if and only if $f(X)$ is uniformly distributed over V. To be able to talk about degrees of balance, we introduce the notion of imbalance.

Definition 4.5 *Let S be a random variable with values in $\mathbb{Z}_M$ taking on each value $j \in \mathbb{Z}_M$ with probability p_j. By the* imbalance *of S, we denote the value between 0 and 1, given by*

$$I(S) \stackrel{\text{def}}{=} M \cdot V[p_0, p_1, \ldots, p_{M-1}] = \frac{M}{M-1} \sum_{j \in \mathbb{Z}_M} \left(p_j - \tfrac{1}{M}\right)^2 . \qquad (4.1)$$

Here the expression $V[\cdot]$ denotes the unbiased estimator [6] for the variance of its arguments. We will often consider the imbalance of the conditional probability distribution of a random variable S depending on a key K conditioned on $K = k$. This quantity will be denoted by $I(S|k)$.

Note that for a uniformly distributed X, the imbalance $I(f(X)) = 0$ if and only if the function f is balanced (i.e., $p_0 = p_1 = \cdots = p_{M-1} = 1/M$).

We will now show a lemma which plays a central role. By using this lemma, we have a convenient way to express the imbalance of a function of a random variable.

Lemma 4.6 *Given a function $s : G \to \mathbb{Z}_M$, let X be a uniformly distributed random variable with values in G. The imbalance $I(S)$ of $S = s(X)$ is then given by*

$$I(S) \quad = \quad \frac{1}{(M-1)N^2} \sum_{l=1}^{M-1} \left| \sum_{x \in G} \mu^{ls(x)} \right|^2 , \qquad (4.2)$$

where μ is an M-th root of unity in $\mathbb{C}$.

Proof Let the function p be given by $p(x) = P[s(X) = x]$, and let $\mathcal{F}_M\{p\}$ denote the ordinary cyclic Fourier transform of p with respect to $\mathbb{Z}_M$, i.e., $\mathcal{F}_M\{p\}(w) = \frac{1}{M} \sum_{x=0}^{M-1} p(x)\mu^{-wx}$, where μ is an M-th root of unity in $\mathbb{C}$. Then

(4.2) follows from applying Parseval's identity.

$$\begin{aligned} I(S) &\stackrel{\text{def}}{=} \frac{M}{M-1}\sum_{j=0}^{M-1}\left(p(j)-\frac{1}{M}\right)^2 \\ &= \frac{M}{M-1}\left[\sum_{j=0}^{M-1}(p(j))^2-\frac{1}{M}\right] \\ &= \frac{M^2}{M-1}\left[\sum_{l=0}^{M-1}|\mathcal{F}_M\{p\}(l)|^2\right]-\frac{1}{M-1} \quad \text{by Parseval's identity} \\ &= \frac{1}{(M-1)N^2}\left[\sum_{l=0}^{M-1}\left|\sum_{j=0}^{M-1}Np(j)\mu^{jl}\right|^2\right]-\frac{1}{M-1} \\ &= \frac{1}{(M-1)N^2}\sum_{l=1}^{M-1}\left|\sum_{x\in G}\mu^{ls(x)}\right|^2. \end{aligned}$$

□

From Lemma 4.6, we deduce the following property for balanced functions from G onto $\mathbb{Z}_M$.

Corollary 4.7 *A function $s: G \to \mathbb{Z}_M$ is balanced if and only if $\mathcal{F}\{\mu^{ls}\}(0) = 0$ for all $l \in \mathbb{Z}_M$ where μ is an M-th root of unity in $\mathbb{C}$.*

Proof Follows directly from Lemma 4.6 and the definition of the Fourier transform. □

To describe the behaviour of a key-dependent random variable over all the possible keys, we introduce the following notion.

Definition 4.8 *The average-key imbalance $\bar{I}(S)$ of a random variable S depending on a key K is defined as the average of $I(S|k)$ when $k \in \mathcal{K}$, i.e.,*

$$\bar{I}(S) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}} I(S|k).$$

4.2.3 I/O Differences

I/O sums are modulo-two sums of certain boolean functions [2]. We broaden this definition somewhat and define an (M-ary) I/O *difference* as follows.

Definition 4.9 *An (M-ary) I/O difference S is the modulo-M difference of a balanced function f on some input X and a balanced function g on some output Y:*

$$S = f(X) \ominus_M g(Y).$$

An I/O difference with imbalance 1 will be called a guaranteed *I/O difference.*

Note that by adding the two I/O differences $S_1 = f_1(Y^{(0)}) \ominus_M f_2(Y^{(1)})$ and $S_2 = f_2(Y^{(1)}) \ominus_M f_3(Y^{(2)})$, the term $f_2(Y^{(1)})$ cancels out, and we obtain a new I/O difference, namely an I/O difference for the cascade of both rounds.

4.3 The Statistical Attack

A common part of GLC and PC is the statistical analysis of p/c-pairs, which may lead to determination of the last round key. More specifically, both attacks depend on the existence of a random variable S depending on p/c-pairs (X, Y) in such a way that it leaks information about the final round key. Therefore, the random variable S must be uniformly distributed when X and Y are randomly chosen and independent of each other, and as non-uniformly distributed as possible (with respect to some measure) when X and Y describe an actual p/c-pair.

In the following, let $Z = (X, Y)$, and let S be a random variable which depends on Z only, such that S has values in $\mathbb{Z}_M$, and, moreover, is uniformly distributed over $\mathbb{Z}_M$ when Z is uniformly distributed over G^2. Let a decision metric $\mathcal{L}$ be a function which maps a random variable S into the real numbers, such that $\mathcal{L}(S) = 0$ if and only if S is uniformly distributed. The imbalance measure introduced earlier is an example of such a decision metric.

An attack descriptor of a statistical attack is a pair $(S, \mathcal{L})$, where S is a random variable of the kind described above and $\mathcal{L}$ is a decision metric used to measure the non-uniformity of S. The attack descriptor will be used to describe LC, GLC, and PC.

Example 4.10 Linear Cryptanalysis. *Matsui's LC applies to ciphers acting on the "xor"-group (i.e., $G = \mathbb{Z}_2^n$). For LC we want to estimate the bias of a random variable $S = (a \bullet X) \oplus_2 (b \bullet Y)$ where $a \in \mathbb{Z}_2^n$, $b \in \mathbb{Z}_2^n$. Hence, the attack descriptor $(S, \mathcal{L})$ is given by*

$$\begin{aligned} S &= (a \bullet X) \oplus_2 (b \bullet Y), \\ \mathcal{L}(S) &= |P[S = 0] - 0.5|. \end{aligned}$$

4.4 Generalized Linear Cryptanalysis

It is possible to generalize LC by replacing the expressions $a \bullet X$ and $b \bullet Y$ from Example 4.10 by $f(X)$ and $g(Y)$ respectively, where f and g are carefully chosen balanced, binary-valued functions. Here, we will not restrict these functions to be binary, and we use the imbalance measure from Definition 4.5 as decision metric. Thus, the attack descriptor for GLC is given by

$$\begin{aligned} S &= f(X) \ominus_M g(Y), \\ \mathcal{L} &= I. \end{aligned}$$

The following theorem provides an upper bound for the imbalance of two-round I/O differences. Note that for one-round I/O differences there exists no upper bound on the imbalance strictly smaller than 1 since it is always possible to find a guaranteed I/O difference for one round.

Theorem 4.11 *The average-key imbalance of the two-round I/O difference* $S = f(X) \ominus_M g(Y)$*, where* $Y = \varphi(\varphi(X + K^{(1)}) + K^{(2)})$*, is upper-bounded by the maximum of a certain Fourier power spectrum involving the round bijection. More precisely,*

$$\bar{I}(S) \leq \max_{a,b \in G \setminus \{0\}} \left| \mathcal{F}\{\chi^b \circ \varphi\}(a) \right|^2. \tag{4.3}$$

For a proof see Appendix. By using Theorem 4.11 it is possible to bound the maximum average-key imbalance over a given round function and in this way obtain an indication of the security of a cipher.

4.5 Partitioning Cryptanalysis

In the basic form of PC, one analyzes the distribution of a pair of random variables $(f(X), g(Y))$, where $f : G \to \mathbb{Z}_{M_1}$ and $g : G \to \mathbb{Z}_{M_2}$ are balanced functions. To put this within the framework of looking at random variables with values in $\mathbb{Z}_M$, we use an invertible function $h : \mathbb{Z}_{M_1} \times \mathbb{Z}_{M_2} \to \mathbb{Z}_M$ where $M = M_1 \cdot M_2$. Thus, the attack descriptor is given by

$$\begin{aligned} S &= h(f(X), g(Y)), \\ \mathcal{L} &= I. \end{aligned}$$

The imbalance of S is upper-bounded by the same expression that bounds imbalances for GLC.

Theorem 4.12 *Let* $f : G \to \mathbb{Z}_{M_1}$ *and* $g : G \to \mathbb{Z}_{M_2}$ *be balanced functions, and let* $M = M_1 \cdot M_2$*. Furthermore, let* $h : \mathbb{Z}_{M_1} \times \mathbb{Z}_{M_2} \to \mathbb{Z}_M$ *be an invertible function. The average-key imbalance of the two-round random variable* $S = h(f(X), g(Y))$*, where* $Y = \varphi(\varphi(X + K^{(1)}) + K^{(2)})$ *satisfies*

$$\bar{I}(S) \leq \max_{a,b \in G \setminus \{0\}} \left| \mathcal{F}\{\chi^b \circ \varphi\}(a) \right|^2. \tag{4.4}$$

For a proof see Appendix.

4.6 Conclusion

The presented bounds demonstrate that the expression

$$\max_{a,b \in G \setminus \{0\}} \left| \mathcal{F}\left\{\chi^b \circ \varphi\right\}(a) \right|^2 \tag{4.5}$$

serves as an overall measure of the security of a cipher against LC, GLC, and PC. The bound might be useful if one wants to prove a cipher to be secure against these attacks. A value close to 0 indicates a strong cipher, and a value close to 1 indicates an insecure cipher. As an example, "perfectly non-linear" or almost bent functions [1, 5, 9, 10, 11, 12] have low valued Fourier power spectra, and thus they provide excellent immunity against LC, GLC, and PC.

To compute the value of (4.5) for an actual cipher, one has to find the maximum either analytically or by doing a brute-force search through all $a, b \in G \setminus \{0\}$. By using the Fast Fourier Transform (FFT), the time complexity for this method can be reduced from $O(N^3)$ to $O(N^2 \log N)$. However, as $N = 2^{64}$ for typical ciphers, a direct computation is infeasible. It is an open question whether there exist more efficient methods to find the maximum, or if at least it is possible to find another (perhaps weaker) easily computable upper bound.

To deduce the actual success-rate of an attack given an imbalance bound, one has to consider the probability distributions of the various imbalance measures (typically, one will have to consider a χ^2-distribution). The use of "multiple approximations" [13] has not been considered in this paper but our results can be adapted for such use.

4.7 Acknowledgements

It is a pleasure to thank Tom Høholdt, James Massey, and Kim Lüders-Jensen for helpful comments.

Bibliography

[1] DILLON, J. F., "Elementary Hadamard Difference Sets", *Proceedings of the Sixth Southeastern Conference on Combinatorics, Graph Theory and Computing*, Boca Raton, Florida, 1975.

[2] HARPES, C., G. G. Kramer, and J. L. Massey, "A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma", *Proceedings of Eurocrypt'95*, Springer.

[3] HARPES, C., "Cryptanalysis of Iterated Block Ciphers", Ph.D. thesis No. 11625, Swiss Federal Inst. of Tech., 1996.

[4] JAKOBSEN, T., *Security Against Generalized Linear Cryptanalysis and Partitioning Cryptanalysis*, Semester project at Signal and Information Processing Laboratory, Swiss Federal Institute of Technology Zurich, Zürich 1995.

[5] KUMAR, P. V., R. A. Scholtz and L. R. Welch, "Generalized Bent Functions and Their Properties", *J. Combinatorial Theory*, Ser. A 40, 1985, 90-107.

[6] LARSON, H. J., *Introduction to Probability Theory and Statistical Inference*, John Wiley & Sons, 1969.

[7] MATSUI, M., "Linear Cryptanalysis Method for DES Ciphers", *Proceedings of Eurocrypt '93*, Springer, 1993.

[8] NATIONAL BUREAU OF STANDARDS, "Data Encryption Standard", Federal Information Processing Standards Publications No. 46, 1977.

[9] NYBERG, K., "Constructions of Bent Functions and Difference Sets", *Proceedings of Eurocrypt'90*, Springer, 1990.

[10] NYBERG, K., "New Bent Mappings Suitable for Fast Implementation", *Fast Software Encryption*, Lecture Notes in Computer Science 809, Springer, 1993.

[11] OLSEN, J. D., R. A. Scholtz and L. R. Welch, "Bent Function Sequences", *IEEE Trans. Inform. Theory*, IT-28, 1982, 858-864.

[12] ROTHAUS, O. S., "On 'Bent' Functions", *J. Combinatorial Theory*, Ser. A 20 (1976), 300-305.

[13] KALISKI, B. S. and M. J. Robshaw, "Linear Cryptanalysis Using Multiple Approximations", *Proceedings of Crypto'94*, Springer, 1994.

4.8 Appendix

This appendix contains proofs of Theorems 4.11 and 4.12.

Proof Theorem 4.11. We consider only one and a half round, i.e., we leave out the final bijection since it has no influence on the upper bound and set $Y = \varphi(X + K^{(1)}) + K^{(2)}$. According to the definition of average-key imbalance, we have

$$\begin{aligned} \bar{I}(S) &= \frac{1}{N^2} \sum_{k^{(1)},k^{(2)} \in G} I(S|k^{(1)}, k^{(2)}) \\ &= \frac{1}{N^2} \sum_{k^{(1)},k^{(2)} \in G} I\left(f(X) \ominus_M g(\varphi(X + k^{(1)}) + k^{(2)})\right) \end{aligned}$$

According to Lemma 4.6, this equals

$$\frac{1}{(M-1)N^2} \sum_{k^{(1)},k^{(2)} \in G} \sum_{l=1}^{M-1} \left| \frac{1}{N} \sum_{x \in G} \mu^{lf(x)} \mu^{-lg(\varphi(x+k^{(1)})+k^{(2)})} \right|^2,$$

where μ is an M-th root of unity in $\mathbb{C}$. Letting

$$\bar{I}_l = \frac{1}{N^2} \sum_{k^{(1)},k^{(2)} \in G} \left| \frac{1}{N} \sum_{x \in G} \mu^{lf(x)} \mu^{-lg(\varphi(x+k^{(1)})+k^{(2)})} \right|^2 \tag{4.6}$$

we proceed to upper-bound $\bar{I}_l$ (note that $\bar{I}(S) = 1/(M-1) \sum_{l=1}^{M-1} \bar{I}_l$). To simplify notation, we let $F_a^l = \mathcal{F}\{\mu^{lf}\}(a)$ and $G_b^l = \mathcal{F}\{\mu^{lg}\}(b)$. We express $\mu^{lf(x)}$ by the inverse Fourier transform

$$\mu^{lf(x)} = \sum_{a \in G} F_a^l \chi^a(x).$$

Similarly, for $\mu^{lg(y)}$ we have

$$\mu^{lg(y)} = \sum_{b \in G} G_b^l \chi^b(y).$$

Thus, letting $y = \varphi(x + k^{(1)}) + k^{(2)}$, we obtain

$$\begin{aligned} \mu^{-lg(\varphi(x+k^{(1)})+k^{(2)})} &= \left[\mu^{lg(\varphi(x+k^{(1)})+k^{(2)})}\right]^* \\ &= \left[\sum_{b \in G} G_b^l \chi^b(\varphi(x + k^{(1)}) + k^{(2)})\right]^*. \end{aligned}$$

Substitution in (4.6) yields

$$\begin{aligned}\bar{I}_l &= \frac{1}{N^2}\sum_{k^{(1)},k^{(2)}\in G}\left|\frac{1}{N}\sum_{x\in G}\left[\sum_{a\in G}F_a^l\chi^a(x)\right]\cdot\left[\sum_{b\in G}G_b^l\chi^b(\varphi(x+k^{(1)})+k^{(2)})\right]^*\right|^2 \\ &= \frac{1}{N^2}\sum_{k^{(1)},k^{(2)}\in G}\left|\frac{1}{N}\sum_{x\in G}\left[\sum_{a\in G}F_a^{l*}\chi^{a*}(x)\right]\cdot\left[\sum_{b\in G}G_b^l\chi^b(\varphi(x+k^{(1)})+k^{(2)})\right]\right|^2.\end{aligned}$$

Recall that $\chi^*(x)=\chi(-x)$ and that $\chi(x+y)=\chi(x)\chi(y)$. Consequently,

$$\begin{aligned}\bar{I}_l &= \frac{1}{N^2}\sum_{k^{(1)},k^{(2)}\in G}\left|\sum_{a,b\in G}\left(F_a^{l*}G_b^l\cdot\frac{1}{N}\sum_{x\in G}\chi^{-a}(x)\chi^b(\varphi(x+k^{(1)})+k^{(2)})\right)\right|^2 \\ &= \frac{1}{N^2}\sum_{k^{(1)},k^{(2)}\in G}\left|\sum_{a,b\in G}\left(F_a^{l*}G_b^l\cdot\frac{1}{N}\sum_{x\in G}\chi^{-a}(x)\chi^{-a}(-k^{(1)})\chi^b(\varphi(x))\chi^b(k^{(2)})\right)\right|^2.\end{aligned}$$

Since $\chi^{-a}(-k^{(1)})=\chi^a(k^{(1)})$, we get

$$\bar{I}_l = \frac{1}{N^2}\sum_{k^{(1)},k^{(2)}\in G}\left|\sum_{a,b\in G}\left(F_a^{l*}G_b^l\chi^a(k^{(1)})\chi^b(k^{(2)})\cdot\frac{1}{N}\sum_{x\in G}\chi^b(\varphi(x))\chi^{-a}(x)\right)\right|^2.$$

By applying Parseval's identity twice, this simplifies to

$$\bar{I}_l = \sum_{k^{(1)},k^{(2)}\in G}\left|\sum_{a,b\in G}\left(F_a^{l*}G_b^l\delta_a(k^{(1)})\delta_b(k^{(2)})\cdot\frac{1}{N}\sum_{x\in G}\chi^b(\varphi(x))\chi^{-a}(x)\right)\right|^2, \tag{4.7}$$

where $\delta_a(x)$ equals 1 if $a=x$ and 0 otherwise (Kronecker's delta). Since f and g are balanced, we have $F_0^l=G_0^l=0$ for all l (cf. Corollary 4.7), and we finally obtain an upper bound for $\bar{I}_l$

$$\begin{aligned}\bar{I}_l &= \sum_{a,b\in G\setminus\{0\}}\left|F_a^{l*}G_b^l\cdot\mathcal{F}\{\chi^b\circ\varphi\}(a)\right|^2 \\ &\le \max_{a,b\in G\setminus\{0\}}\left|\mathcal{F}\{\chi^b\circ\varphi\}(a)\right|^2\cdot\sum_{a\in G}\left|F_a^l\right|^2\cdot\sum_{b\in G}\left|G_b^l\right|^2 \\ &= \max_{a,b\in G\setminus\{0\}}\left|\mathcal{F}\{\chi^b\circ\varphi\}(a)\right|^2.\end{aligned}$$

We are now able to deduce the upper bound for $\bar{I}(S)$.

$$\begin{aligned}\bar{I}(S) &= \frac{1}{M-1}\sum_{l=1}^{M-1}\bar{I}_l \\ &\le \max_{a,b\in G\setminus\{0\}}\left|\mathcal{F}\{\chi^b\circ\varphi\}(a)\right|^2.\end{aligned}$$

□

Proof Theorem 4.12. Let the invertible function h be given by $h(i,j) = i + jM_1$, and note that S is in essense an I/O difference since

$$\begin{aligned} S &= h(f(X), g(Y)) \\ &= f(X) \ominus_M \tilde{g}(Y), \end{aligned}$$

where $\tilde{g}(Y) = -M_1 g(Y)$.

Now we can use the same arguments as in the proof of Theorem 4.11. Note, however, that since S is not a true I/O difference due to f and $\tilde{g}$ not being balanced over $\mathbb{Z}_M$, there is the following difference in the step after (4.7). Since f and g are not balanced, the property $F_0^l = G_0^l = 0$ does not hold for all l. However, one can show that for all values of l either F_0^l or G_0^l is zero, and thus their product is always zero. □

Chapter 5

Attacks on Block Ciphers of Low Algebraic Degree

The following paper was written with Lars R. Knudsen and has been submitted to the Journal of Cryptology in 1999.

Parts of the paper were originally published in Fast Software Encryption, Lecture Notes in Computer Science, vol. 1267, Haifa, Springer, 1997.

Attacks on Block Ciphers of Low Algebraic Degree

Thomas Jakobsen[1] Lars R. Knudsen [2]

Abstract In this paper an attack on block ciphers is introduced, the interpolation attack. This new method is useful for attacking ciphers that use simple algebraic functions (in particular quadratic functions) as S-boxes. Also, attacks based on higher-order differentials are introduced. The latter is a special and important case of the interpolation attacks. The attacks are applied to several block ciphers, the 6-round prototype cipher by Knudsen and Nyberg, which is provably secure against ordinary differential cryptanalysis, a modified version of the block cipher SHARK, and a block cipher suggested by Kiefer.

5.1 Introduction

In an r-round iterated cipher the ciphertext is computed by iteratively applying in r rounds a *round function* g to the plaintext, such that

$$x_i = g(k_i, x_{i-1}),$$

where x_0 is the plaintext, k_i is the ith round key, and $x_r = y$ is the ciphertext. A special kind of iterated ciphers are the **Feistel** ciphers. A Feistel cipher with block size $2m$ and r rounds is defined as follows. Let x_0^L and x_0^R be the left and right-hand halves of the plaintext, respectively, each of m bits. The round function g operates as follows

$$\begin{aligned} x_i^L &= x_{i-1}^R \\ x_i^R &= f(k_i, x_{i-1}^R) + x_{i-1}^L, \end{aligned}$$

and the ciphertext is the concatenation of x_r^R and x_r^L. Note that f can be any function taking as arguments an m-bit text and a round key k_i and producing m bits. '+' is a commutative group operation on the set of m-bit blocks. For the remainder of this paper we will assume that '+' is the exclusive-or operation ($\oplus$).

The attacks presented in this paper are classified according to the taxonomy of [5]. In a *key-recovery attack* an attacker finds the secret key. In a *global deduction* an attacker finds an algorithm, which encrypts any plaintext into a valid ciphertext without knowing the secret key. In an *instance deduction* an

[1]Department of Mathematics, Building 303, Technical University of Denmark, DK-2800 Lyngby, Denmark. Email: T.Jakobsen@mat.dtu.dk.

[2]University of Bergen, Dept. of Informatics, Bergen, Norway. Email: lars.knudsen@ii.uib.no

attacker finds an algorithm, which encrypts a subset of all plaintexts into valid ciphertexts without knowing the secret key.

The *reduced cipher* is the cipher that one gets by removing the final round of the original cipher. The output from this cipher is denoted $\tilde{y} = (\tilde{y}_L, \tilde{y}_R)$.

In the key-recovery attacks one tries to find the value of the last-round key. A guess of this value is used to decrypt the ciphertext by one round and in this way one hopes to obtain the output from the reduced cipher. If there exists a method to distinguish whether this is the actual output from the reduced cipher or not, then one can find the last-round key. Once this key has been found, attacks similar to the ones presented here can be mounted on a cipher one round shorter than the original. As the measurement of the time needed by an attack, the total number of encryptions of the attacked block cipher is used. Note that this general description of an attack can be extended to the case where the attacker looks for the first-round key instead of the last-round key or both at the same time.

This paper considers two types of attacks. First, attacks based on higher order differentials are given. Generalisations of this attack is then introduced under the name of interpolation attacks.

Let the ciphertext bits y_j be expressed as multivariate polynomials $q_j(x) \in \mathrm{GF}(2)[x_1, \ldots, x_m]$, where the x_is are the plaintext bits. The nonlinear order of the encryption function is then defined to be the maximum total degree of these polynomials, $\max_j \deg q_j$. The higher-order differential attack is applicable if the nonlinear order d of the ciphertext from the reduced cipher as a function of the plaintext is low. Since a dth-order differential over such a cipher is a constant, it is possible to predict certain values of the output of the reduced cipher, which can be used to recover the last-round key.

In the univariate version of the interpolation attack, the ciphertexts are expressed as polynomials $p(x) \in \mathrm{GF}(2^m)$ of the plaintexts x. If such a polynomial has a sufficiently low degree or a low number of coefficients, it is possible to reconstruct it from a collection of plaintexts and their corresponding ciphertexts. This can be used to construct an algorithm which can encrypt and decrypt without knowledge of the secret key, and it can be used to recover the secret key in iterated ciphers. By viewing the plaintexts and ciphertexts as the concatenation of s blocks of m/s bits, where s divides m, the attack is generalised to multivariate polynomials. It follows that the higher-order differential attack is the special case where $s = m$.

This paper is organised as follows. § 5.2 gives new attacks based on higher-order differentials and in § 5.3 a new attack on block ciphers is presented, the interpolation attack. In § 5.4 the attacks are applied to the cipher by Knudsen and Nyberg [12] and to a modification hereof, to a modified version of SHARK [13], and to a cipher by Kiefer [4]. Conclusions are in § 5.5.

5.2 Attacks Using Higher-Order Differentials

In [7] Lai gave a definition of higher-order derivatives of discrete functions. Later Knudsen used higher-order differentials to cryptanalyse ciphers presumably secure against conventional differential attacks, that is, attacks based on first order differentials [6]. An extension of Knudsen's attacks is given next. The reader is referred to [6, 7] for the definitions of higher-order differentials.

Consider a Feistel cipher with block size $2m$. Suppose that x_R is kept constant and consider the right-hand side $\tilde{y}_R$ of the output from the reduced cipher. Since x_R is a constant, each bit of $\tilde{y}_R$ can be expressed as a multivariate polynomial $\mathrm{GF}(2)[x_1, x_2, \ldots, x_m]$ in the bits of $x_L = (x_1, x_2, \ldots, x_m)$. Assume that none of these polynomials have degree higher than d. Then according to [7, Proposition 2] (see also [6]), we have

$$\sum_{x_L \in \mathcal{L}_d} p(x_L) = c, \tag{5.1}$$

where $\mathcal{L}_d$ denotes a d-dimensional subspace of $\mathrm{GF}(2)^m$, c is a constant for any space parallel to $\mathcal{L}_d$, and p is a function which computes the output from the reduced cipher. It follows that

$$\sigma(w) = \sum_{x_L \in \mathcal{L}_{d+1}} p(x_L + w) = 0 \text{ for all } w \in GF(2)^m \tag{5.2}$$

if and only if $p(x)$ is a polynomial of degree d or lower. In the following algorithm, the variables $x = (x_L, x_R)$ and $y = (y_L, y_R)$ hold the plaintext and the ciphertext, respectively. L is a full rank $(d+1) \times m$ matrix over $\mathrm{GF}(2)$ and f the round function.

1. Let x_R and w be m-bit constants.
2. For all $a \in \mathrm{GF}(2)^{d+1}$:
   (a) Let $x_L = aL + w$.
   (b) Obtain the ciphertext $y(a)$ of plaintext (x_L, x_R).
3. For all values, k, of the last-round key:
   (a) Let $\sigma = 0$.
   (b) For all $a \in \mathrm{GF}(2)^{d+1}$:
      i. Let $y = y(a)$.
      ii. Let $\tilde{y}_R = y_L \oplus f(k, y_R)$.
      iii. Let $\sigma = \sigma \oplus \tilde{y}_R$.

The key for which σ ends up being zero is the correct last-round key with a high probability. Consequently, for every possible value k of the last-round key, we check whether the corresponding value of σ is zero, and if it is, then we have found the correct key with high probability. If one wants a higher level of certainty, the algorithm is simply repeated with another choice of w. This method is easily generalised to any iterated cipher, and we get the following result, extending that of [6, Th. 11].

Theorem 5.1 *Given an iterated block cipher, let d denote the maximum polynomial degree of $m' > 1$ ciphertext bits of the round next to the last expressed as a function of the plaintext bits. Furthermore, let b denote the number of last-round key bits. Assume that the polynomial degree of the ciphertext bits increases with the number of rounds. Then there exists a d-th order differential attack of expected time complexity 2^{b+d} requiring $2^{d+1} \cdot g$ chosen plaintexts with $g = \lceil b/m' \rceil$ which will successfully recover the last-round key.*

Proof We give the proof in the case of a Feistel cipher, from which the general case follows. Consider the iteration (3b). Let k denote the correct value of the last-round key, and let k' denote any wrong value. Then

$$\begin{aligned} \tilde{y}_R &= y_L \oplus f(k, y_R) \\ \tilde{y'}_R &= y_L \oplus f(k', y_R) \\ &= \tilde{y}_R \oplus f(k, y_R) \oplus f(k', y_R). \end{aligned}$$

The difference between $\tilde{y}_R$, obtained using the correct key, and $\tilde{y'}_R$, obtained with a wrong key, is two applications of the function f. By assumption the polynomial degree increases with the number of rounds, and consequently one can expect that σ will be zero only for the correct value of the last-round key with a high probability. An incorrect value of the last-round key will be suggested with probability $2^{-m'}$. After one attack about $2^{b-m'}$ values will be candidates for the correct key. Repeating the attack $g = \lceil b/m' \rceil$ times discards most wrong values of the key. The time complexity of the attack is $2^{b+d} + 2^{b+d-m'} + 2^{b+d-2m'} + ... + 1$ which is roughly 2^{b+d} when $m' > 1$. □ The

attack can be improved by a factor of two, if the constant of Equation (5.1) can be predicted. In that case the iterations (2) and (3b) of the above algorithm are performed only for all $a \in \mathrm{GF}(2^d)$. The key for which $\sigma = c$ will be the correct key with a high probability. For most ciphers, depending on the f-function, there are possible extensions to the above attack. It may be possible to perform the attack for only a subset of the bits in the last-round key, and also it may be possible to search for (a part of) the first-round key.

5.3 The Interpolation Attack

In this section, a new attack is introduced on block ciphers. The attack is based on the following well-known formula.

Let F be a field and let $2n$ elements $x_1, \ldots, x_n,\ y_1, \ldots, y_n \in F$ be given, where the x_is are distinct. Define

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \le j \le n, j \ne i} \frac{x - x_j}{x_i - x_j}. \tag{5.3}$$

Then $f(x)$ is the only polynomial over F of degree at most $n-1$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$. Equation (5.3) is known as the *Lagrange interpolation formula* (see e.g. [2, page 185]).

In the *interpolation attacks* presented in this paper one constructs polynomials using pairs of plaintexts and ciphertexts. For these attacks it is not always necessary to find all the coefficients of the polynomial, merely to compute the value of $f(x)$ for one or a few values of x. In the following it is shown how to evaluate f in linear time in some point x (in particular $x = 0$) using n (other) evaluations of f.

Assume that f has degree at most $n-1$. Also assume that there are given n points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i = \alpha^i$ for some primitive element $\alpha \in F$ and $y_i = f(x_i)$. Then to evaluate f in a point $x \neq x_1, x_2, \ldots, x_n$, we use the Lagrange interpolation formula:

$$f(x) = \sum_{i=1}^{n} y_i \prod_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}$$

Evaluating this expression for $x = 0$ yields

$$f(0) = \sum_{i=1}^{n} y_i \prod_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{x_j}{x_j - x_i} = \sum_{i=1}^{n} y_i \cdot \frac{g_i}{h_i},$$

where (with $x_i = \alpha^i$)

$$g_i = \frac{\alpha^{(1+2+\cdots+n)}}{\alpha^i} = \alpha^{n(n+1)/2-i}$$

and

$$h_i = \prod_{\substack{1 \leq j \leq n \\ j \neq i}} \left(\alpha^j - \alpha^i\right).$$

Rewriting one gets

$$h_1 = \prod_{j=2}^{n} \left(\alpha^j - \alpha\right)$$

and the recurrence relation

$$h_{i+1} = h_i \cdot \frac{\alpha^{n-1}\left(\alpha^i - 1\right)}{\alpha^n - \alpha^i}.$$

In a similar way, more general formulas for computing $f(x)$ for arbitrary values of x can be derived. Using these expressions, it is possible to compute $f(0)$ (or more generally $f(x)$) in linear time $O(n)$ and constant memory.

Note that the values of g_i and h_i can be precomputed since they are independent of the y_i values.

Next it is explained how to utilize the above in attacks on block ciphers. Consider an m-bit secret-key block cipher for a fixed (unknown) key. The ciphertext y can be described as a polynomial $p(x) \in \mathrm{GF}(2^m)[x]$ of the plaintext. If the number of coefficients of these polynomials is sufficiently low, one

can reconstruct it with sufficiently many plaintext-ciphertext pairs by solving a simple system of linear equations. Subsequently, one has an algorithm which can encrypt and decrypt plaintexts and ciphertexts but without knowledge of the secret key.

In a chosen plaintext variant of this attack it is possible for an attacker to establish polynomials with a reduced number of coefficients by fixing some of the bits in the chosen plaintexts. In that case, the result is an instance deduction, since the obtained algorithm can only encrypt plaintexts for which a number of bits are fixed to a certain value.

Consider a key-recovery variant of the attack. Instead of specifying the ciphertext as a function of the plaintext, the output from the reduced cipher $\tilde{y}$ is expressed as a polynomial $p(x) \in \mathrm{GF}(2^m)[x]$ of the plaintext. Assume that this polynomial has degree d (and hence has at most $d+1$ unknown coefficients) and that $d+2$ known p/c-pairs are available. Then for all values of the last-round key one decrypts the ciphertexts one round, constructs the polynomial with $d+1$ pairs and checks whether the polynomial is correct for the remaining p/c-pair. If this is the case, then the correct value of the last-round key has been found with a high probability, by reasoning similarly as in the proof of Theorem 5.1. In this variant of the attack, it suffices to be able to construct a function value of the polynomial, and the method in the beginning of this section can be applied. But even if the key guess is wrong, the value of the polynomial may be accepted with probability 2^{-m} since this is the chance of correctly guessing at random one out of 2^m values. Assuming that the key consists of b bits, this means that on average one falsely gets 2^{b-m} values accepted as candidates for the correct key. However, by simply checking additional pairs with the polynomial, the probability of failure falls accordingly.

One may also consider a multivariate generalization of the interpolation attack. An m-bit plaintext can be viewed as the concatenation of s subblocks each of m/s bits corresponding to elements in $\mathrm{GF}(2^{m/s})$. Accordingly, the ciphertext may then also be viewed as s subblocks of m/s bits. Each of these ciphertext subblocks is then expressible as a multivariate polynomial evaluated in the plaintext subblocks.

The following result sums up the attack.

Theorem 5.2 *Consider an m-bit iterated block cipher with s subblocks each of m/s bits. Express an output subblock from the round next to the last as a (uni- or multivariate) polynomial in $GF(2^{m/s})$ of (some of) the plaintext blocks and let n denote the number of coefficients in the polynomial. Assume that the number of coefficients in such a polynomial increases with the number of rounds in the cipher. Furthermore, let b denote the number of last-round key bits involved in the attack. Then there exists an interpolation attack of expected time complexity $2^b(n+1)$ requiring $n+\lambda$ known (or chosen) plaintexts with $\lambda = \lceil bs/m \rceil$ which will successfully recover the last-round key.*

Proof Let k denote the correct value of the last-round key, and let k' denote any wrong value. Let $\tilde{y}$ and $\tilde{y'}$ be the text obtained from the ciphertexts by

decrypting one round with the correct key k, respectively a wrong key k'. Then

$$\begin{aligned} \tilde{y} &= g^{-1}(k, y) \\ \tilde{y'} &= g^{-1}(k', y) \\ &= g^{-1}(k', g(\tilde{y}, k)) \end{aligned}$$

The difference between $\tilde{y}$, obtained using the correct key, and $\tilde{y'}$, obtained with a wrong key, is two applications of the round function g. By assumption the polynomial degree increases with the number of rounds, thus for a wrong guessed value of the key one will not succeed to generate a correct polynomial. The probability of guessing correctly a random m/s-bit value is $2^{-m/s}$. Repeating the feat λ times has probability $2^{-\lambda m/s}$. Therefore the expected number of false positives is $t = 2^{b-\lambda m/s}$ for which the expected number of texts needed follows. The time complexity is $(2^b(n+1) + 2^{b-m/s}(n+2) + 2^{b-2m/s}(n+3) + \dots)$ which is roughly $2^b(n+1)$ for $m > s$. □

Similar to the attack of Theorem 5.1 it may be possible to perform the attack for only a subset of the bits of the last-round key, and also it may be possible to search for (a subset of) the first-round key, depending on the structure of the round function. Similarly for some round functions it may be advantageous to solve algebraically for the round key in the last round instead of trying all possible 2^b values, as illustrated in [14].

In the above interpretation, the higher-order differential attack is a special case with $s = m$, 1-bit subblocks, and polynomials in m variables. However, note that in the higher-order differential attacks typically one attacks many 1-bit subblocks simultaneously.

For some ciphers it is possible to mount attacks for several, different values of s. It depends on the specific block cipher which of these attacks is the most efficient. In Section 5.4 examples are given to illustrate this.

Meet-in-the-middle Approach

The attacks described in this section are extensions of the attacks in the previous sections using a meet-in-the-middle technique. Only the extension of the key-recovery attack is described; the extension of the global and instance deductions follows easily.

Once more, one tries to guess the correct last-round key and use this to (hopefully) obtain $\tilde{y}$, the output from the reduced cipher. In the following, only the verification of $\tilde{y}$ is described. Given an iterated cipher of r rounds, let z denote the output of round r', where $r' \leq (r-1)$. The value of z is expressible via the plaintext x as a polynomial $h_1(x) \in \mathrm{GF}(2^m)[x]$ where m is the block size. Similarly, z can be expressed as a polynomial $h_2(\tilde{y}) \in \mathrm{GF}(2^m)[\tilde{y}]$ of the output $\tilde{y}$ of the reduced cipher. Let the degree of $h_1(x)$ be d_1, the degree of $h_2(\tilde{y})$ be d_2 and let $d = d_1 + d_2$. Thus, the following equation

$$h_1(x) = h_2(\tilde{y}) \tag{5.4}$$

has at most $d+2$ unknowns. One can show that the equation is uniquely solvable up to a multiplication and an addition of both g and h with a constant. To ensure that a non-trivial and unique solution is obtained, the coefficient corresponding to the highest exponent is set equal to 1 and the constant term equal to 0. After this, the equation is solved by using d known or chosen plaintexts. What is left is to check whether yet another p/c-pair $(x, \tilde{y})$ satisfies $h_1(x) = h_2(\tilde{y})$. If it does, then it is assumed that the correct value of the last-round key has been found.

The following result sums up the attack.

Theorem 5.3 *Consider an m-bit iterated block cipher with s subblocks each of m/s bits and with r rounds. Express the output from round r', $r' \leq r-1$, as a multivariate polynomial of (some of) the plaintext blocks and let n_1 denote the number of coefficients in the polynomial. Also, express the output from round r' as a polynomial of the output subblocks from round $(r-1)$, and let n_2 denote the number of coefficients in the polynomial. Furthermore, set $n = n_1 + n_2$ and let b denote the number of last-round key bits. Assume that the number of coefficients in such polynomials increases with the number of rounds. Then there exists an interpolation attack of expected time complexity $2^b(n+1)$ requiring $n+g$ known (or chosen) plaintexts with $g = \lceil bs/m \rceil$ which will successfully recover the last-round key.*

The best known methods for solving a system of n linear equations in n unknowns that the authors are aware of, require $O(n^2)$ words of memory and run in time (approximately) $O(n^3)$. However, it is not necessary to solve the systems of equations, merely to have an algorithm which can detect whether a solution exists. Furthermore, the equation (5.4) has a very special form and it is expected that special methods will exist which require less memory and time.

5.4 Examples

In this section the attacks of the previous sections are applied to a range of proposed block ciphers.

5.4.1 Nyberg and Knudsen's cipher

Based on the use of a quadratic function over a Galois field, Knudsen and Nyberg demonstrated in [12] how to construct a cipher which is provably secure against differential cryptanalysis [1]. The cipher is a Feistel cipher with the nonlinear function f given by $F : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{32})$ with

$$f(k, x) = d(h(e(x) \oplus k)),$$

where $h : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{33})$, $h(x) = x^3$, $k \in \mathrm{GF}(2^{33})$, $e : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{33})$ is a function which extends its argument by concatenation with an affine combination of the input bits, and $d : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{32})$ discards one bit from its argument. When used with 6 rounds, one can show that the probability of any differential and/or linear hull can be bounded sufficiently low and subsequently

# Rounds	# Chosen plaintexts	Running time
6	2^9	2^{41}
6	2^5	2^{70}
7	2^{17}	2^{49}
7	2^9	2^{74}
8	2^{17}	2^{82}

Table 5.1: Higher-order differential attacks on the Knudsen-Nyberg cipher.

there is a proof that this yields a secure cipher (with respect to conventional differential cryptanalysis). Also, the cipher is secure against the linear attack [8], which follows from [10].

In the following the higher-order differential attack is applied. Choose plaintexts where the right-hand halves are fixed. Since the output bits from the round function are only quadratic in the input bits, the polynomial degree of the bits in the reduced cipher as a function of the plaintext bits is not higher than 8.

Therefore from Theorem 5.1 it follows that there exists an attack, which requires only $2^{8+1} = 512$ chosen plaintexts and an expected running time of order 2^{41}. A variant of the attack searching for the keys in the last two rounds requires about 32 chosen plaintexts and an expected running time of order 2^{70}. Similarly, there are attacks on versions with 7 and 8 rounds, the complexities are given in Table 5.1. The attack has been implemented on scaled-down versions, and it recovers the last-round key as predicted. Note that these attacks are applicable to ciphers with any block size $2m$, as long as the number of chosen plaintexts is less than 2^m. The bigger the block size the more rounds can be attacked.

Also, in [14] it was shown that for these attacks the round key in the last round can be solved for algebraically as opposed to trying all possible 2^b values.

5.4.2 A Dedicated Cipher

Consider the Feistel cipher with round function given by $f(k, x) = h(x \oplus k)$ where $h : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{32})$, $h(x) = x^3$, that is, the input to the cubing function is not extended and the output not truncated as in the previous case. This cipher is similar to the cipher in the previous section, and is also secure against the (conventional) differential attacks [12] and against the linear attack [8]. The cipher is as vulnerable to the higher order differential attack as the previous one, but much more vulnerable to the interpolation attack for $s > 1$, as shown in the following.

Express the ciphertext halves as polynomials of the plaintext halves in $GF(2^{32})[x]$. It follows by easy calculations that these polynomials have at most $3^{2r-1} + 3^r + 3^{r-1} + 1$ coefficients. Note, that degrees of x_R and x_L are at most 3^r and 3^{r-1}, respectively. Thus, this polynomial can be reconstructed by considering at most $3^{2r-1} + 3^r + 3^{r-1} + 1$ plaintext/ciphertext pairs (p/c-pairs) using, e.g., Lagrange interpolation. With $r = 6$ the attack needs at most 2^{18}

known p/c-pairs, which yields an algorithm for a global deduction. Note that the number of coefficients will be lower than specified, since not all elements $x_L^i x_R^j$ for $0 \leq i \leq 3^r$ and $0 \leq j \leq 3^{r-1}$ will appear in the polynomial.

For the key-recovery attack assume that the right-hand half x_R of the plaintext is fixed (that is, consider a chosen plaintext attack), and consider the right-hand side of the output $\tilde{y}_R = p(x_L)$ from the reduced cipher expressed as a polynomial $p(x_L) \in \mathrm{GF}(2^{32})[x_L]$. This polynomial has degree at most $3^3 = 27$ since the degree does not increase in the first round and since $\tilde{y}_R$ equals the left-hand half of the output of the fourth round. Consequently, 28 pairs of corresponding values of x_L and $\tilde{y}$ are enough to determine it uniquely (using Lagrange interpolation). It is then tested whether $\tilde{y}$ is actually output from the reduced cipher or not. This is done by verifying whether a 29-th p/c-pair agrees with the obtained polynomial. If it does, then it is assumed that the correct key has been found. The expected time complexity is $29 \times 2^{32-1} \approx 2^{36}$.

The meet-in-the-middle variant can be applied as follows. Assume again that the right-hand half x_R of the plaintext is fixed. Let z_L denote the left-hand half of the output from round four. The value of z_L is expressible via the plaintext as a polynomial $g(x_L) \in \mathrm{GF}(2^{32})[x_L]$. This polynomial has degree at most 3^2, that is, there are at most 10 non-zero coefficients in $g(x_L)$. Similarly, z_L can be expressed as a polynomial $h(\tilde{y}_L, \tilde{y}_R) \in \mathrm{GF}(2^{32})[\tilde{y}_L, \tilde{y}_R]$ of the output from the reduced cipher. It follows that $h(\tilde{y}_L, \tilde{y}_R) = \tilde{y}_L^3 \oplus a\tilde{y}_L^2 \oplus b\tilde{y}_L \oplus c \oplus \tilde{y}_R$, where a, b, and c are some key-dependent constants. Thus, there are at most $10 + 3 = 13$ unknown coefficients of the equation

$$g(x_L) = h(\tilde{y}_L, \tilde{y}_R) \tag{5.5}$$

Setting the constant term of g to equal 0 (the coefficient corresponding to the highest exponent in h has already been found to equal 1), one proceeds to solve the resulting system of equations by using 12 p/c-pairs from the reduced cipher. Thus, one obtains the polynomials g and h. It is then checked whether yet another p/c-pair $(x, \tilde{y})$ satisfies $g(x_L) = h(\tilde{y}_L, \tilde{y}_R)$. If it does, then it is assumed that the correct key has been found.

Similar attacks can be applied to versions of the cipher with up to 32 rounds at least in theory. Consider the version with 32 rounds. Let $g(x_L) \in \mathrm{GF}(2^{32})[x_L]$ be an expression of the left-hand half z_L of the output from round 22. The degree of this polynomial is at most 3^{20}. Let $h(\tilde{y}_L, \tilde{y}_R) \in \mathrm{GF}(2^{32})[\tilde{y}_L, \tilde{y}_R]$ be an expression of z_L from the output of the reduced cipher. In the algebraic normal form of $h(\tilde{y}_L, \tilde{y}_R)$, the number of exponents in $\tilde{y}_L$ and $\tilde{y}_R$ is at most $(3^9 + 1)$ and $(3^{10} + 1)$, respectively. Thus, the number of coefficients in $h(\tilde{y}_L, \tilde{y}_R)$ is at most $(3^9 + 1)(3^{10} + 1) \approx 3^{19}$. This means that the number of coefficients in Equation (5.5) is at most $3^{20} + 3^{19} \approx 2^{32}$.

5.4.3 Attacks on Modified SHARK

The iterated cipher SHARK was described by Rijmen, Daemen, *et al.* in [13]. The cipher has a block size of nm bits and each round has a non-linear layer

and a diffusion layer. The non-linear layer consists of n parallel m-bit S-boxes. The diffusion layer consists of an nm-bit linear mapping constructed from a Reed-Solomon code. There are two suggested ways to introduce the keys into the cipher. The first is by a simple exclusive-or with the inputs to the S-boxes, the other uses a key-dependent affine mapping. Also, an output transformation is applied after the last round of SHARK. The transformation consists of a key addition and an inverse diffusion layer.

Denote by SHARK(n, m, r) the version with a block size of nm bits using n parallel m-bit S-boxes in r rounds. In [13] an implementation SHARK$(8, 8, r)$ (64-bit blocks) is given. The 8 S-boxes are identical and constructed from the permutation $h : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ given by $h(x) = x^{-1}$. The cipher is analysed with respect to linear and differential attacks, and it is argued that 8 rounds of SHARK$(8, 8, r)$ give a security level comparable to that of triple-DES, and from [13, Table 1] it follows that 4 rounds of this version give a security level comparable to that of DES.

In the following it is shown that there are many instances of SHARK that can be broken significantly faster than expected.

First of all, the number of rounds of SHARK must be determined with respect to the non-linear order of the S-boxes. Assume that the outputs of the S-box have non-linear order d in the input bits. Since the S-boxes represent the only non-linear component in SHARK, the non-linear order of the ciphertexts after r rounds of encryption will be at most d^r. To avoid attacks based on higher-order differentials it must be ensured that d^r is high, preferably that $d^r \geq nm$. Thus, for a 64-bit block cipher, if $d = 2$, e.g. using the cubing function in a Galois field, the number of rounds must be at least 6.

Consider versions of SHARK where the keys are mixed with the texts by the exclusive-or operation. As will be shown there are instances of SHARK(n, m, r), for which the interpolation attacks are applicable. Consider 64-bit versions using as S-box $h(x) = x^{-1}$ in $\mathrm{GF}(2^m)$, which is the S-box suggested in [13]. The inverse permutation in a Galois field has a high algebraic degree, note that $h(x) = x^{-1} = x^{2^m-2}$ in $\mathrm{GF}(2^m)$. However, as will be shown, the interpolation attack is applicable with a complexity which depends only on the number of S-boxes and on the number of rounds in the cipher.

Consider first a version with $n = 1$. It follows by easy calculations that the ciphertext y *after any number of rounds* can be expressed as a fraction of polynomials of the plaintext x (or similarly, x can be expressed as a polynomial of y) as follows

$$y = \frac{x \oplus a}{bx \oplus c} \tag{5.6}$$

where a, b, c are key-dependent constants. These three constants can be found using the interpolation attack with only 4 known p/c-pairs[3] by considering and solving $y \cdot (bx \oplus c) = (x \oplus a)$. The result is a global deduction, that is, an algorithm that encrypts (decrypts) any plaintext (ciphertext).

[3] In [9] a similar cipher was investigated. It was explained that this cipher could be solved with a number of known plaintexts linear in the number of rounds.

For $n > 1$ the number of coefficients in the polynomials used in the attacks increases with the number of diffusion layers in the cipher. Note that because of the inverse diffusion layer in the output transformation there are only $r-1$ diffusion layers in an r-round version of SHARK. In the following consider a version of SHARK(n, m, r). Let the plaintext words each of m bits be denoted $x_1, \ldots, x_n$, and let the ciphertext words be denoted $y_1, \ldots, y_n$. Express the ciphertext words as polynomials in $GF(2^m)$ in terms of the plaintext words. For $r = 1$, one gets expressions of the form $y_i = \frac{a x_i}{b x_i \oplus c}$. For $r = 2$ each ciphertext word can be written as a fraction of polynomials where in the denominator one gets an expression of degree at most n. Also, the degree of the polynomial in the enumerator is at most the degree of the polynomial in the denominator. Thus, the number of coefficients in the fraction of polymials is at most 2×2^n. By doing similar calculations for $r = 3$ and so on, it follows that the number of coefficients in the polynomials for r rounds is at most

$$2 \cdot (n^{r-2} + 1)^n.$$

This is also the number of known plaintexts for the interpolation attack on an r-round version yielding a global deduction. It follows that the attack is independent of the sizes of the S-boxes, and depends only on the number of S-boxes and the number of rounds.

The interpolation attack with the meet-in-the-middle technique can be applied also for these ciphers. Consider the interpolation attack with known plaintexts. One first establishes

$$\frac{q_{j,1}(y_1, \ldots, y_n)}{q_{j,2}(y_1, \ldots, y_n)} = \frac{p_{i,1}(x_1, \ldots, x_n)}{p_{i,2}(x_1, \ldots, x_n)}, \tag{5.7}$$

that is, expressions of the ciphertexts in one middle round, where $i + j = r - 1$, using polynomials of both the plaintext and the ciphertext. Subsequently, one can solve the following systems of equations

$$q_{j,1}(y_1, \ldots, y_n) \cdot p_{i,2}(x_1, \ldots, x_n) = p_{i,1}(x_1, \ldots x_n) \cdot q_{j,2}(y_1, \ldots, y_n). \tag{5.8}$$

The number of known plaintexts required to solve (5.8) is

$$2 \cdot (n^{r_1 - 1} + 1)^n \cdot (n^{r_2 - 1} + 1)^n,$$

where $r_1 + r_2 = r - 1$ and $r_1, r_2 \geq 1$, which follows by calculations similar to the above.

The round keys for SHARK are typically quite big, so the general key-recovery attack described earlier in this paper may be impractical. However, it is possible to perform the attack for only a subset of the first-round and/or last-round keys. As an example, one can repeat the attack for all values of the first s words of the first-round key and express the ciphertext (of a middle round) as a polynomial $p_{i,1}(S(x_1 \oplus k_1), \ldots, S(x_s \oplus k_s), x_{s+1}, \ldots x_n)$, where $S(\cdot)$ are the S-boxes and x_i are the plaintext words. The values of the key words for which

# Rounds	# S-boxes	Known plaintexts	Memory	Time
any	1	3		
6	2	2^9	2^{18}	2^{27}
6	4	2^{27}	2^{54}	2^{81}
3	8	2^{17}	2^{34}	2^{51}
4	8	2^{35}	2^{70}	2^{105}
5	8	2^{52}	2^{104}	2^{156}
6	8	2^{75}	2^{150}	2^{225}

Table 5.2: Complexities of the interpolation attack on variants of SHARK using as S-box $h(x) = x^{-1}$.

the interpolation succeeds are candidates for the secret key, and the attack is repeated sufficiently many times until one value of the secret key is found.

In Table 5.2 the complexities are given of the interpolation attack on variants of SHARK using as S-box $h(x) = x^{-1}$ in $\mathrm{GF}(2^m)$. It follows that using 8 S-boxes, the 64-bit variant with up to 5 rounds and the 128-bit variant with up to 8 rounds are, at least theoretically, vulnerable to our attacks. The required amount of memory and time for the versions with 5 and 6 rounds are of course unrealistic today. However, as discussed earlier, the linear equations obtained in the attacks are of a very special form. Therefore there might exist methods solving such systems faster than for systems of arbitrary linear equations. Furthermore, the complexities were computed assuming that the number of coefficients in the polynomials are maximum. In practice, this number might be much smaller. Also, depending on the chosen key length the attacks can be faster than an exhaustive search for the keys. In a chosen plaintext attack the number of coefficients in the polynomials used in the attack can be reduced by fixing some plaintext bits. As examples, there exist interpolation attacks on the variant with 8 S-boxes and 4 rounds using about 2^{21} chosen plaintexts and on the variant with 8 S-boxes and 7 rounds using about 2^{61} chosen plaintexts.

It has been demonstrated that certain instantiations of SHARK are insecure. The results also demonstrate a case where the use of bigger and fewer S-boxes does not result in more secure ciphers. Finally, it is noted that the designers of SHARK expressed their concern with the use of the inverse in a Galois field as S-boxes[13].

5.4.4 Kiefer's Scheme

In this section the scheme by Kiefer [4] is attacked in a higher-order differential attack. The cipher is probabilistic and uses the following encryption rule:

$$x_i \mapsto (F(k) \oplus r_i, f_k(r_i) \oplus x_i), \tag{5.9}$$

where $F : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a one-way function, $f_k : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a function depending on the key $k \in \mathrm{GF}(2^m)$ in some complex way, $r_i \in \mathrm{GF}(2^m)$

is a random value, and $x_i \in \mathrm{GF}(2^m)$ is a message block. The function f_k has the form $f_k = \pi_k \circ g$ where $\pi_k : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a bitwise linear transform depending on k and $g : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ is a public, almost perfectly non-linear function of the form $g(x) = x^{2^s+1}$ for some s.

Assume that enough plaintext is available to have four pairs on the form

$$(a_i, b_i) = (F(k) \oplus r_i, f_k(r_i)), \ i = 1, \ldots, 4 \tag{5.10}$$

such that $a_1 \oplus a_2 = a_3 \oplus a_4$. Define $\beta = \bigoplus_{i=1}^{4} b_i$ and $\gamma = \bigoplus_{i=1}^{4} g(r_i)$. Then

$$\beta = \bigoplus_{i=1}^{4} b_i = \pi_k \left(\bigoplus_{i=1}^{4} g(r_i) \right) = \pi_k(\gamma). \tag{5.11}$$

Since $\{a_1, \ldots, a_4\}$ is a two-dimensional subspace of $\mathrm{GF}(2^n)$, the elements in $\{r_1, \ldots, r_4\}$ also constitute a two-dimensional subspace. Note also that the Hamming weight of the exponent in the definition of g expressed as a binary number is only two, implying that the output bits are only quadratic in the input bits. By Equation (5.1), this implies that one can compute the value of γ.

If repeated m times, one obtains m corresponding pairs of β and γ. This makes it possible to solve Equation (5.11) with respect to the unknown function π_k (it is a linear transform). After having found π_k, invert f_k and thus obtain a value of r_i. Subsequently, compute $F(k)$ and the system is broken.

It remains to compute the minimum number t of known plaintexts needed to obtain m times four pairs (a_i, b_i) with the required property; recall that the cipher is probabilistic and thus the attacker has no control over the values of r_i. By using a birthday paradox type argument it can be shown that $t \approx (m \cdot 2^{m+2})^{\frac{1}{4}}$. For a typical block size of $m = 64$ this gives $t \approx 2^{18}$.

5.5 Concluding Remarks

A new attack on block ciphers, the interpolation attack, was introduced. The interpolation attack is a natural extension of the higher-order differential attack, but in many cases much more efficient than the latter. It was demonstrated that the attack works on several proposed block ciphers. In particular, it was shown that a cipher provably secure against differential and linear cryptanalysis is very vulnerable to the interpolation attack. Also, variants of the attack were used to cryptanalyse the (unmodified) cipher by Knudsen and Nyberg, variants of the cipher SHARK, and a cipher by Kiefer.

Recently, a probabilistic version of the interpolation attack was introduced [3]. This version of the attack finds a polynomial relation between plaintexts and ciphertexts which hold only for a fraction of all cases. It was demonstrated that this attack can be used to break the cipher by Knudsen and Nyberg [12].

5.6 Acknowledgments

The authors wish to thank Vincent Rijmen for valuable comments.

Bibliography

[1] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer Verlag, 1993.

[2] P.M. Cohn. *Algebra, Volume 1*. John Wiley & Sons, 1982.

[3] T. Jakobsen. Cryptanalysis of block ciphers with probabilistic non-linear relations of low degree. In H. Krawczyk, editor, *Advances in Cryptology: CRYPTO'98, LNCS 1462*, pages 212–222. Springer Verlag, 1998.

[4] K. Kiefer. A new design concept for building secure block ciphers. In J. Pribyl, editor, *Proceedings of the 1st International Conference on the Theory and Applications of Cryptology, PRAGOCRYPT'96, Prague, Czech Republic*, pages 30–41. CTU Publishing House, 1996.

[5] L.R. Knudsen. *Block Ciphers – Analysis, Design and Applications*. PhD thesis, Aarhus University, Denmark, 1994.

[6] L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 196–211. Springer Verlag, 1995.

[7] X. Lai. Higher order derivatives and differential cryptanalysis. In R. Blahut, editor, *Communication and Cryptography, Two Sides of One Tapestry*. Kluwer Academic Publishers, 1994. ISBN 0-7923-9469-0.

[8] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93, LNCS 765*, pages 386–397. Springer Verlag, 1993.

[9] K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93, LNCS 765*, pages 55–64. Springer Verlag, 1993.

[10] K. Nyberg. Linear approximations of block ciphers. In A. De Santis, editor, *Advances in Cryptology - EUROCRYPT'94, LNCS 950*, pages 439–444. Springer Verlag, 1995.

[11] K. Nyberg. Block ciphers. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT'96, LNCS 1163*. Springer Verlag, 1996.

[12] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *The Journal of Cryptology*, 8(1):27–38, 1995.

[13] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher SHARK. In Gollmann D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 99–112. Springer Verlag, 1996.

[14] S. Moriai, T. Shimoyama, and T. Kaneko. Improving the higher order differential attack and cryptanalysis of the KN cipher. Presented at Information Security Workshop'97, ISW 97, Kanazawa, Sept., 1997.

Chapter 6

Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relation of Low Degree

The following paper was originally published in the Lecture Notes in Computer Science, vol. 1462, Proceedings of Crypto '98, Santa Barbara, Springer, 1998.

Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree

Thomas Jakobsen
Department of Mathematics
Building 303
Technical University of Denmark
DK-2800 Lyngby, Denmark
Email: T.Jakobsen@mat.dtu.dk.

Abstract Using recent results from coding theory, it is shown how to break block ciphers operating on GF(q) where the ciphertext is expressible as evaluations of an unknown univariate polynomial of low degree m over the plaintext with a typically low but non-negligible probability μ. The method employed is essentially Sudan's algorithm for decoding Reed-Solomon codes beyond the error-correction diameter. The known-plaintext attack needs $n = 2m/\mu^2$ plaintext/ciphertext pairs and the running time is polynomial in n. Furthermore, it is shown how to discover more general non-linear relations $p(x, y) = 0$ between plaintext x and ciphertext y that hold with small probability μ. The second attack needs access to $n = (2m/\mu)^2$ plaintext/ciphertext pairs where $m = \deg p$ and its running time is also polynomial in n. As a demonstration, we break up to 10 rounds of a cipher constructed by Nyberg and Knudsen provably secure against differential and linear cryptanalysis.

Key words: Cryptanalysis, block cipher, interpolation attack, non-linear relations, Reed-Solomon codes, Sudan's algorithm.

6.1 Introduction

For some block ciphers, the round function can be described by a low degree polynomial for a non-negligible fraction of its input values. This may happen if there are bad S-boxes or if simple algebraic functions are used unwisely. (Some simple functions provide very good immunity against differential and linear cryptanalysis.) This paper shows how one may break such ciphers.

Previous work has focused on either the linear case or the case where the output is always expressible as a low degree polynomial (not just a fraction of the time). For instance, Matsui's linear cryptanalysis [14] is applicable when some of the output bits can be described as a linear combination of the input bits for a sufficient fraction of the possible plaintexts. Jakobsen and Knudsen's interpolation attack [9] demonstrates how to break ciphers for which the ciphertext is always (with probability 1) expressible as a low degree polynomial of the plaintext. Their attack fails when "noise" is introduced. Similarly, Lai's higher order differentials [10] [9] work only in the case where the output is always expressible as a low-degree polynomial of the input.

Assume that the output of some block cipher can be expressed as evaluations of a degree m polynomial for a fraction of μ of its possible inputs. We will say that such a cipher is (m, μ)-expressible. Intuitively, such ciphers appear to be weak. However, the problem of successfully cryptanalyzing (m, μ)-expressible ciphers can be shown to be essentially equivalent to the problem of decoding very low-rate Reed-Solomon codes subject to severe noise (with error rate above $\frac{1}{2}$). Efficient decoding of such codes was not possible until recently where Sudan [17] [18] published a very novel algorithm which is able to correct several more errors in polynomial time.

The paper is organized as follows. First we give some preliminaries in Section 6.2 and show how to obtain the round keys of a block cipher one at a time given that there exists a method to distinguish random pairs from actual plaintext/ciphertext pairs.

In Section 6.3, the Reed-Solomon codes and Sudan's result will be explained and in Section 6.4 an attack using Sudan's algorithm is presented. If there exist low degree polynomials describing the ciphertext for a sufficient number of the inputs, then the algorithm will find them. This information leak gives probabilistic knowledge of the ciphertext. As mentioned above this information in turn can be used to obtain the round keys. As a demonstration we cryptanalyze several rounds of a cipher constructed by Nyberg and Knudsen [15] which is immune to both differential and linear cryptanalysis. We break several rounds faster than exhaustive key search and using less than 2^{32} plaintext/ciphertext pairs (*p/c-pairs*).

Section 6.5 describes a more general attack. Here the probabilistic relation between plaintext x and ciphertext y has the more general form $p(x, y) = 0$ for some bivariate polynomial p with low degree. We conclude in Section 6.6 with some comments and by stating possible applications and extensions of the attack.

6.2 Preliminaries

We consider r-round iterated block ciphers with round function

$$C_i = F_{K_i}(C_{i-1})$$

where C_0 is the plaintext, K is the ith round key, and C_r is the ciphertext. We will assume that F is a bijection taking values in $\mathrm{GF}(q)$, where q is an integer such that a finite field of size q exists. In addition, we assume that the round keys are independent, uniformly distributed, and, moreover, that they are introduced by some group operation in such a way that the cipher is a Markov cipher [12]. Considering plaintext and keys to be random variables this implies that the inputs to each round may be considered independent.

Definition 6.1 *Given a function* $f : \mathrm{GF}(q) \to \mathrm{GF}(q)$ *and a polynomial* $p : \mathrm{GF}(q) \to \mathrm{GF}(q)$ *we say that* f *is* (m, μ)-expressible *if*

$$f(x) = p(x) \text{ holds with probability at least } \mu, \tag{6.1}$$

where $\deg(p) \leq m$.

Example 6.2 *Let* $a, b \in \mathrm{GF}(2^w)$ *and let the function* $\mathrm{XOR}(a,b) = a + b$ *be defined by the bitwise addition of its arguments. Similarly, let* $\mathrm{ADD}(a,b)$ *be defined by the modulo-n addition of the arguments considered as elements of* $\mathbb{Z}_n$ *where* $n = 2^w$. *These functions are used in several block ciphers to represent "incompatible" groups, e.g. in [11] or [16].*

Given two values a and b, if there is no bit position other than the most significant bit where both have a 1, *then* $\mathrm{XOR}(a,b) = \mathrm{ADD}(a,b)$. *In other words,* ADD *is* $\left(1, \left(\frac{3}{4}\right)^{w-1}\right)$*-expressible over* $\mathrm{GF}(2^w)$.

We now show what happens if one iterates (m, μ)-expressible round functions.

Proposition 6.3 *Consider an r-round Markov cipher with round function F. Assume that F is* (m, μ)*-expressible. Then the cipher is* (m^r, μ^r)*-expressible.*

Note that there may be a better approximation for the whole r-round cipher. However, it is at least (m^r, μ^r)-expressible.

Proof Consider two applications of F_{k_i}: $y = F_{k_1}(x)$ and $z = F_{k_2}(y)$, i.e., $z = F_{k_2}(F_{k_1}(x))$. Then y is expressible as a polynomial $q_1(x)$ with $\deg(p_1) \leq m$ for a fraction $\mu_1 = \mu$ of the possible values of x. Similarly z is expressible as a polynomial $p_2(y)$ with $\deg(p_2) \leq m$ for a fraction $\mu_2 = \mu$ of the possible values of y.

Since the cipher is a Markov cipher, we may assume that the inputs to each round are statistically independent, and hence z is expressible as a polynomial $p(x) = p_2(p_1(x))$ with $\deg(p) \leq m^2$ for a fraction $\mu_1\mu_2 = \mu^2$ of the possible input values, i.e., it is (m^2, μ^2)-expressible.

The proof is finished by induction on the number of rounds. □

If we have a probabilistic relation between plaintext and ciphertext expressed as a polynomial, then we already have an information leakage and the cipher may be considered broken. Indeed, the following proposition shows us how we may divide and conquer using this information to obtain the round keys one at a time, in effect peeling off one round after another. But first we need some definitions.

Definition 6.4 *Let there be given a set* $S = \{(x_i, y_i)\}_{i=1,\ldots,n}$ *of pairs and a block cipher. An algorithm which can successfully distinguish a set of p/c-pairs from a set of random pairs is called a* discriminator *(with respect to that cipher).*

Matsui's linear relations, the differential characteristics of Biham and Shamir, and the polynomial relations described above are all examples of useful expressions for discriminators.

The following is a variant of what Harpes, Kramer, and Massey [6] refer to as the hypothesis of wrong-key randomization.

Definition 6.5 *Let there be given an r-round block cipher C. Define by the* reduced *cipher $\tilde{C}$ the first $r-1$ rounds of C. Additionally, let there be given a set $S = \{(x_i, y_i)\}_{i=1,\dots,n}$ of p/c-pairs and a discriminator for the reduced cipher $\tilde{C}$. Let S_k be the set constructed from S by decrypting the ciphertexts y by one round using last-round key k. Furthermore, let k_c denote the actual (correct) last-round key and let $k_w \neq k_c$ be a wrong guess. The discriminator is said to be* compliant *if it successfully distinguishes S_{k_c} from S_{k_w}.*

Informally speaking, the term "wrong-key randomization" comes from the fact that (hopefully) decryption using the wrong last-round key will randomize the p/c-pairs.

Proposition 6.6 *Given some block cipher C, assume that there exists a compliant discriminator for the corresponding reduced cipher $\tilde{C}$ requiring access to n pairs and running in t steps. Then it is possible to obtain the last round key of C using n p/c-pairs and expected time $\frac{1}{2}t|\mathcal{K}|$ where $\mathcal{K}$ is the key space of the last round.*

Proof To find the last round key simply make a guess and decrypt the ciphertexts by one round. Then use the discriminator to check if the decryptions belong to the reduced cipher. If this is the case we found the correct key. Otherwise proceed with another guess. There are $|\mathcal{K}|$ possible round keys and the discriminator runs in t steps for an expected running time of $\frac{1}{2}t|\mathcal{K}|$. □

Note that an attack like the above might be entirely impractical due to large $|\mathcal{K}|$. The motivation to include Prop. 6.6, however, was to demonstrate how an information leak can sometimes be exploited to break a cipher entirely. The existence of a polynomial approximation in itself is usually enough to consider a cipher broken.

6.3 Reed-Solomon Codes

The Reed-Solomon codes [13] are a class of linear codes over the alphabet $\mathrm{GF}(q)$. The $[n,k]_q$ Reed-Solomon code, where n is the length (usually $n = q-1$) and k is the dimension of the code is obtained by letting each message $r = r_0 \dots r_{k-1}$ denote the coefficients of a degree $k-1$ polynomial $p(x) = \sum_{i=0}^{k-1} r_i x^i$. The corresponding codeword $y = y_0 \dots y_{n-1}$ is the concatenation of evaluations of p over distinct elements of $\mathrm{GF}(q) \setminus \{0\}$, e.g. $y_i = p(\alpha^i)$, $i = 0, \dots, n-1$, where α is a primitive element of $\mathrm{GF}(q)$.

There exist efficient algorithms for decoding Reed-Solomon codes. For instance, the classical Berlekamp-Massey algorithm [13], which is capable of correcting $t = \lfloor (d-1)/2 \rfloor$ errors, where d is the minimum distance of the code. However, for previously known algorithms t/n never exceeds 0.5 by much, not even for very low rates. To be useful for our purpose this bound on t is too low. Sudan's algorithm, on the other hand, corrects 100% of the errors asymptotically (for rates going towards 0).

The decoding problem as treated by Sudan may be stated as the following: Given integers n, k, and e. Furthermore n pairs $\{(x_i, y_i)\}_{i=1}^{n}$, $x_i, y_i \in \mathrm{GF}(q)$ with pairwise distinct x_i. Compute all polynomials $p_1, \ldots, p_m$ of degree $k-1$ such that for every $j = 1, 2, \ldots, m$, the following holds: $p_j(x_i) = y_i$ for at least $(n-e)$ values of $i = 1, \ldots, n$. It is not hard to see the similarity between this decoding problem and the problem of discovering a probabilistic relation $y = p(x)$ between plaintext x and ciphertext y.

The algorithm given by Sudan [18] solves this problem in polynomial time for values of e very close to n. The main result of Sudan [17] is the following:

Theorem 6.7 *For every ε and κ, the bounded distance decoding problem with parameters n, $k = \kappa n$, and $e = \epsilon(\kappa)n$ can be solved in polynomial time provided*

$$\varepsilon(\kappa) < 1 - \frac{1}{1+\rho_k} - \frac{\rho_k}{2}\kappa, \text{ where } \rho_\kappa = \left\lfloor \sqrt{\frac{2}{\kappa} - \frac{1}{4}} - \frac{1}{2} \right\rfloor.$$

Here κ is the fraction of information bits per codeword and $\varepsilon(\kappa)$ is the corresponding error rate. Note that for small κ we have $\rho_\kappa \approx \sqrt{2/\kappa}$, and in this case the right hand side of the inequality is approximately $1 - \sqrt{2\kappa}$.

Decoding beyond the packing radius is achieved by a very novel approach. Sudan's algorithm obtains a bivariate polynomial $Q(x, y)$ which is then factored into irreducibles. The error positions are then derived from the factorization and the received vector.

The following section shows how the error-correcting algorithm may be used to mount an attack.

6.4 Attack 1

Definition 6.8 *Let $a, b \in \mathbb{N}$. The (a,b)-weighted degree of a bivariate polynomial $Q(x,y) = \sum_{ij} q_{ij} x^i y^j$ is defined by*

$$\deg^{(ab)}(Q) = \max\{ia + jb | q_{ij} \neq 0\}.$$

The following algorithm is based on the modified Sudan's algorithm found in [4].

Attack 1:

- Input: n p/c-pairs $\{(x_i, y_i)\}_{i=1}^{n}$, $0 \leq \mu \leq 1$, $m \in \mathbb{N}$, such that $n > (2m)/(\mu^2)$.
- Output: All expressions $y - p(x)$ with $\deg(p) \leq m$ such that $y = p(x)$ holds with probability at least μ.
- Step A: Denote by $s_i(x, y)$ the i-th bivariate monomial in the $(1, m-1)$-weighted graded order. Let $Q(x, y) = \sum_{i=1}^{n+1} s_i(x, y)$ and let q_{ij} denote the coefficient of the monomial $x^i y^j$. Find a nonzero solution q_{ij} to the set of linear equations $Q(x_s, y_s) = 0$, $s = 1, \ldots, n$.

- Step B: Factor the polynomial $Q(x, y)$ into irreducibles over $\mathrm{GF}(q)[x, y]$.
- Step C: Output all factors $y - p(x)$ with $\deg(p) \leq m$ such that $p(x_i) = y_i$ for at least a fraction μ of $i = 1, \ldots, n$.

For a proof of Sudan's algorithm consult [17], [18], or [4]. The algorithm runs in polynomial time since there are efficient algorithms for solving linear equations and factoring polynomials [5].

As an optimization [8], note that it is possible to obtain from a bivariate polynomial factors on the form $y - p(x)$ by using a homomorphism from $\mathrm{GF}(q)[y]$ to $\mathrm{GF}(q_2)$ (for an appropriate power of q, q_2). Simply consider $Q(x, y) \in \mathrm{GF}(q)[x, y]$ as a polynomial in y from $\mathrm{GF}(q_2)[y]$ and then use, e.g., Berlekamp's algorithm [2] for factorization of univariate polynomials.

Theorem 6.9 *An (m, μ)-expressible cipher can be broken by Attack 1 using*

$$n = \frac{2m}{\mu^2} \tag{6.2}$$

plaintext/ciphertext pairs in time polynomial in n.

Proof The theorem follows directly by rewriting Sudan's formula (setting $m = k$, $\mu = 1 - \varepsilon$ and approximating $\lfloor \sqrt{2/\kappa + 1/4} - 1/2 \rfloor$ by $\sqrt{2/\kappa}$; κ is assumed to be near 0 since $k \ll n$). □

Example 6.10 *The cipher constructed in [15] by Knudsen and Nyberg is immune to differential and linear cryptanalysis. It falls for an attack using Sudan's algorithm.*

The cipher is a Feistel network with round function $F_k(x) = d(f(e(x) + k))$ where $f : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{33})$, $f(x) = x^3$, $k \in \mathrm{GF}(2^{33})$, $e : \mathrm{GF}(2^{32}) \to \mathrm{GF}(2^{33})$ is a function which extends its argument by concatenation with an affine combination of the input bits, and $d : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{32})$ discards one bit from the argument. As in [9] we call this cipher $\mathcal{KN}$. The following equations describe the cipher

$$\begin{aligned} C_i^L &= C_{i-1}^R \\ C_i^R &= F_{K_i}(C_{i-1}^R) + C_{i-1}^L. \end{aligned}$$

The plaintext is (C_0^L, C_0^R) and the ciphertext is the concatenation of C_r^R and C_r^L. Note that because of the extend and discard functions, one round cannot be written as a low-degree polynomial over $\mathrm{GF}(q)$.

Define the following variant $\mathcal{KN}'$ taking inputs $(C_0^L, C_0^R) \in \mathrm{GF}(2^{33})^2$ and

having outputs $(C_r^R, C_r^L) \in \mathrm{GF}(2^{33})^2$*:*

$$\begin{aligned} D_0^L &= d(C_0^L) \\ D_0^R &= d(C_0^R) \\ D_i^L &= D_{i-1}^R \\ D_i^R &= F_{K_i}(D_{i-1}^R) + D_{i-1}^L \\ C_r^L &= e(D_r^L) \\ C_r^R &= e(D_r^R), \end{aligned}$$

In other words, $\mathcal{KN}'$ *is simply* $\mathcal{KN}$ *preceded by discard operations and followed by extend operations. Clearly, if we can break* $\mathcal{KN}'$ *then we can also break* $\mathcal{KN}$*. Consequently, we proceed by attacking* $\mathcal{KN}'$*.*

Consider yet another cipher $\mathcal{PURE}$ *defined by the purely algebraically given round function* $\tilde{F} : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{33})$*,* $\tilde{F}_k(x) = f(x+k)$*,* $f(x) = x^3$*. I.e.,*

$$\begin{aligned} C_i^L &= C_{i-1}^R \\ C_i^R &= \tilde{F}_{K_i}(C_{i-1}^R) + C_{i-1}^L. \end{aligned}$$

Again, the ciphertext is the concatenation of C_r^R *and* C_r^L*. Here* $C_i^L, C_i^R \in \mathrm{GF}(2^{33})$*. Essentially,* $\mathcal{PURE}$ *is the same cipher as* $\mathcal{KN}'$ *but without the extend/discard functions that ruin the algebraic simplicity. Now keep the right half of the plaintext* C_0^R *constant and express the right half of the ciphertext* C_r^L *as a polynomial of the left half of the plaintext* C_0^L*. In the original proposal,* $\mathcal{KN}$ *has* $r = 6$ *rounds. Assume that this holds for* $\mathcal{KN}'$ *and* $\mathcal{PURE}$ *as well. Then the output polynomial of the right half has degree* $3^{(6-2)} = 81$ *due to the cipher's simple algebraic structure (*C_0^L *passes through* $r-2$ *instances of* $\tilde{F}$ *before "becoming"* C_r^L*). This implies that* $\mathcal{PURE}$ *can be broken by the interpolation attack which was exactly what was done in [9].*

Assume that the position of the discarded bit is the same as the position of the extended bit. In this case, given the same inputs, the outputs from the round functions F *and* $\tilde{F}$ *of* $\mathcal{KN}'$ *and* $\mathcal{PURE}$*, respectively, will agree with probability* $\frac{1}{2}$ *(when the extension function correctly "guesses" the missing bit). In other words,* $e(d(f(x)))$ *is* $(3, \frac{1}{2})$*-expressible over* $\mathrm{GF}(2^{33})$*. Consequently, given identical inputs with right halves fixed, the right halves of the outputs of the two ciphers* $\mathcal{KN}'$ *and* $\mathcal{PURE}$ *will agree on a fraction of* $2^{-(6-2)} = 1/16$ *of the possible plaintexts (we assume that the inputs to each round are uncorrelated).*

Now we can use Thm. 6.9. We have $m = 81$ *and* $\mu = 1/16$*. Consequently, using Sudan's algorithm we need at least*

$$n = \frac{2 \times 81}{\left(\frac{1}{16}\right)^2} \approx 40000 < 2^{16}$$

pairs (x_i, y_i) *to successfully discriminate random samples from p/c-pairs.*

Combining Prop. 6.3 and Thm. 6.9 we can calculate the maximum number of

rounds possible to break. Solving

$$\frac{2 \cdot 3^{r-2}}{1/\left(2^{r-2}\right)^2} \leq 2^{32}$$

for integer solutions gives a maximum of $r = 10$ rounds for which the cipher is breakable using at most 2^{32} p/c-pairs. Using higher order differentials (h.o.d.) as in [9], one can break only 7 rounds of $\mathcal{KN}$. Additionally, the h.o.d. approach depends on the extension bit being an affine combination of the input bits; this implies that the output bits of the round function may be considered as evaluations of quadratic polynomials of the input bits. Our attack does not rely on this assumption.

6.5 Attack 2

In [1], Ar et al. shows how to obtain low-degree relations $p(x, y) = 0$ that hold on a non-negligible number of elements of some set $\{(x_i, y_i)\}_{i=1,\ldots,n}$ (given the relations exist). Here we present a slightly weaker theorem which has the advantage of a shorter and less involved proof. In order to prove that the attack works, we need the following lemmas.

Lemma 6.11 Bézout's Theorem. *Let $P(x, y), Q(x, y) \in \mathrm{GF}(q)[x, y]$ be polynomials in two variables over $\mathrm{GF}(q)$. If the polynomials have no common factors, then the number of common zeros is at most $\deg P \cdot \deg Q$, where deg denotes total degree.*

For a proof consult [7].

Lemma 6.12 *Let $f(a, b)$ denote the number of bivariate polynomials in $\mathrm{GF}(q)[x, y]$ with degree a in x and degree b in y. Similarly, let $\Phi(a, b)$ count the number of irreducibles among these polynomials. Then*

$$\Phi(a, b) = (1 - q^{-a}) f(a, b) + O(aq^{ab}),$$

where the constant in the O-term depends on q and a.

A proof is found in [3]. Restated we get the following.

Lemma 6.13 *Let $p(x, y)$ be a random bivariate polynomial over $\mathrm{GF}(q)$ of degree a in x and degree b in y. Then the probability of $p(x, y)$ being irreducible satisfies*

$$\mathrm{Prob}[p(x, y) \text{ is irreducible}] \geq 1 - q^{-\max\{a,b\}}.$$

In other words, nearly all bivariate polynomials are irreducible.

Attack 2:

- Input: μ, m, n p/c-pairs $\{(x_i, y_i)\}$, where $n > (2m/\mu)^2$.

- Output (with high probability): All probabilistic links $p(x, y)$ with $\deg(p) \leq m$ satisfying $\text{Prob}[p(x, y) = 0] \geq \mu$.
- Step A: Let $t_i(x, y)$ denote the i-th monomial in the graded order. Let $Q(x, y) = \sum_{i=1}^{n+1} t_i(x, y)$ and let q_{ij} be the coefficient of the monomial $x^i y^j$. Find a nonzero solution q_{ij} to the set of linear equations $Q(x_s, y_s) = 0$, $s = 1, \ldots, n$.
- Step B: Factor $Q(x, y)$. Output all factors of degree less than m.

Theorem 6.14 *Given a block cipher, assume that there exists a probabilistic relation $p(x, y) = 0$ with $\deg(p) \leq m$ between plaintext x and ciphertext y which holds for a fraction μ of the possible plaintexts.*

Then the cipher can be broken by Attack 2 using at most

$$n = \left(\frac{2m}{\mu}\right)^2$$

plaintext/ciphertext pairs and time polynomial in n.

Proof First, we show that $Q(x, y)$ has non-constant factors if there is a probabilistic low degree relation between input and output. Assume that

$$p(x, y) = 0 \text{ with probability } \mu \tag{6.3}$$

for some $p(x, y) \in \text{GF}(q)[x, y]$ with $\deg p \leq m$. In addition, assume that

$$n > \left(\frac{2m}{\mu}\right)^2. \tag{6.4}$$

We have $Q(x_i, y_i) = 0$ for n pairs (x_i, y_i). Of these $m = \mu n$ pairs have the additional property that $p(x, y) = 0$. Consequently, the number of common zeros of p and Q is at least m. We also have $\deg p \leq m$ and because of the way we constructed Q, we have $\deg Q \leq 2\sqrt{n}$. Then due to (6.4) we have $\deg Q \cdot \deg p < m$. By Bézout's theorem this means that p and Q have a common factor. Since $p \neq Q$, the polynomial $Q(x, y)$ must be reducible. In addition, p has a high probability of being irreducible implying that p is most probably one of the obtained factors.

Secondly, to prove that the algorithm outputs nothing when the pairs are truly random (implying that no probabilistic relations exist) it suffices to show that Q is "random". Recall that a random bivariate polynomial has very slim chances of being reducible. Also notice that the construction of Q is a matter of solving n linear equations of n unknown variables (assuming Q is to be normalized). In fact, we may choose the pairs (x_i, y_i) such that we obtain any assignment of coefficients $q_{ij} \in \text{GF}(q)$. As a consequence, given random input we may assume that Q is random and therefore irreducible with high probability.

The algorithm runs in polynomial time since there exists efficient algorithms for solving sets of linear equations and factoring polynomials. □

6.6 Comments

It is possible to break block ciphers which are probabilistically expressible as low degree polynomials faster than exhaustion of the key space. This fact should lead to new design criteria. Clearly, to thwart these attacks it is not enough that round functions have high boolean complexity. Likewise, good properties against differential and linear attacks are no guarantee either. In fact, many almost perfect non-linear functions should be avoided exactly because they are too simple algebraically. At least, they should not be the only ingredients of a strong block cipher. It remains to carry out analysis of existing block ciphers and discover whether they are susceptible to these new attacks.

Although both attacks run in polynomial time, in practice the running time may be substantial. Step A dominates the complexity; more precisely, solving linear equations using, e.g., simple Gaussian elimination gives time complexity $O(n^3)$. Factorization of bivariate polynomials over a finite field has complexity $O(n \log q)^{O(1)}$, see [5]. The memory requirement (to hold the system of linear equations) is proportional to the square of the number of unknown coefficients q_{ij} in $Q(x, y)$, i.e. $O(n^2)$. It might be possible to improve these complexities with elimination algorithms suited for a particular purpose.

For both algorithms to work, there must be included in the n pairs at least μn pairs that satisfy the polynomial relation. Statistically, for n randomly chosen p/c-pairs this holds approximately 50% of the time since there is a fraction μ of good pairs among all possible p/c-pairs. To obtain a higher success rate one can simply use sufficiently many more p/c-pairs.

In this paper we have considered only bivariate relations. However, Sudan et al. describe extensions to several variables. This might be useful for ciphers where there is no natural correspondence between input or output and GF(q), e.g., for DES a more natural input domain would be GF(2)^w instead of GF(2^w) leading to polynomial relations of the form $y_i = p(x_1, \ldots, x_{64})$ or $q(x_1, \ldots, x_{64}, y_1, \ldots, y_{64}) = 0$.

Notice that for Prop. 6.6 to work, the discriminator does not need to explicitly output the probabilistic relation; it just has to state whether one exists. This fact might make it possible to construct even better attacks. In the error-correction setting, this resolves to computing whether a received word is closer to the set of codewords than some given distance.

More recently, Sudan [19] has improved his algorithm by requiring each pair (x, y) to appear as a root in Q with multiplicity greater than 1. This makes it possible to correct even more errors when decoding. In our case, the new results imply that the factor of 2 in Eq. (6.2) becomes close to 1.

Finally, note that both attacks are very well suited for (nearly-)black box analysis since no structure on the block cipher is assumed except the correspondence between plaintext/ciphertext and the elements of GF(q).

6.7 Acknowledgements

Thanks to Tom Høholdt for fruitful discussions and for mentioning Sudan's results in the first place.

Bibliography

[1] Sigal Ar, Richard J. Lipton, Ronitt Rubinfeld, and Madhu Sudan. Reconstructing Algebraic Functions from Mixed Data, *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, 1992, pp. 503–512. To appear SIAM Journal on Computing.

[2] Elwyn R. Berlekamp. Factoring Polynomials over Large Finite Fields. *Mathematics of Computation*, pp. 713, vol. 24, no. 111, 1970.

[3] Leonard Carlitz. The Distribution of Irreducible Polynomials in Several Indeterminates II. *Canadian Journal of Mathematics* 17:261-266, 1965.

[4] Weishi Feng and Richard E. Blahut. On Decoding Reed-Solomon Codes Beyond the Packing Radii. Preprint. Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Nov., 1997.

[5] Joachim von zur Gathen and Erich Kaltofen. Factoring multivariate polynomials over finite fields. *Math. Comput.*, 45:251-261, 1985.

[6] Carlo Harpes, Gerhard G. Kramer, and James L. Massey. A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma. *Eurocrypt '95*, Lectures Notes in Computer Science, Springer, 1995.

[7] Robin Hartshorne. *Algebraic Geometry*. Springer-Verlag, New York, 1977.

[8] Tom Høholdt. Private communication.

[9] Thomas Jakobsen and Lars R. Knudsen. The Interpolation Attack on Block Ciphers. *Fast Software Encryption IV*, Lecture Notes in Computer Science, Springer, Haifa, 1997.

[10] Xueijia Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of James L. Massey on the occasion of his 60'th birthday*, Feb. 10–13, 1994, Monte-Verita, Ascona, Switzerland, 1994.

[11] Xuejia Lai and James L. Massey. A Proposal for a New Block Encryption Standard, *Advances in Cryptology - Eurocrypt '90 Proceedings*, Springer-Verlag, Berlin, 1991, pp. 389–404.

[12] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. *Advances in Cryptology, Proceedings Eurocrypt '91*, LNCS 547, D. W. Davies, Ed., Springer-Verlag, 1991, pp. 17–38.

[13] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[14] Mitsuru Matsui. Linear cryptanalysis for DES cipher. *Lecture Notes in Computer Science*, *765* (1994), 386–397. (Advances in Cryptology - EUROCRYPT '93.)

[15] Kaisa Nyberg and Lars R. Knudsen. Provable Security Against a Differential Attack. *Journal of Cryptology*, vol. 8, no. 1, 1995.

[16] Ronald L. Rivest. The RC5 encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop*, Lecture Notes in Computer Science, vol. 1008, pp. 86-96, Leuven, Belgium, Springer-Verlag, Published 1995.

[17] Madhu Sudan. Decoding Reed Solomon Codes beyond the Error-Correction Diameter. *Proc. 35th Annual Allerton Conference on Communication, Control and Computing*, University of Illinois at Urbana-Champaign, 1997.

[18] Madhu Sudan. Decoding of Reed Solomon Codes beyond the Error-Correction Bound. *Journal of Complexity*, 13(1):180-193, March 1997.

[19] Madhu Sudan. Preprint. May 1998.

Chapter 7

Decoding Reed-Muller Codes Beyond Half the Minimum Distance

The following paper was written with Agnes Heydtmann. An early version was presented at the Winter School in Coding and Information Theory 1998, Ebeltoft, Denmark.

Decoding Reed-Muller Codes Beyond Half the Minimum Distance

Agnes E. Heydtmann and Thomas Jakobsen [1]

Abstract Inspired by Sudan's recent algorithm for Reed-Solomon codes we propose an efficient method for decoding r^{th}-order Reed-Muller codes of length 2^m which can correct errors beyond half the minimum distance.

This procedure involves interpolating a polynomial $Q \in \mathbb{F}_2[x_1, \ldots, x_m, y]$ vanishing when evaluated at points in $\mathbb{F}_2^m$ joint with the corresponding received bits. To obtain a list of codewords closest to the received word we need to factor Q considered as an element of the quotient ring of boolean polynomials which is not a unique factorization domain. Therefore we introduce a novel, yet simple polynomial-time factorization algorithm for multivariate boolean polynomials that produces generators for the coset of factors.

Let $p = 2^{-\lambda}$ be the probability of algorithm failure and assume that the weights of a Reed-Muller code are approximately binomially distributed. This assumption is supported by known weight distributions for some short-length Reed-Muller codes. Then with probability at least $1 - p$, the algorithm corrects

$$\tau \leq \max_{\rho} \min \left\{ 2^m - \sum_{i=0}^{r+\rho} \binom{m}{i} - \lambda, \sum_{i=0}^{\rho} \binom{m}{i} - 1 \right\}$$

independently and uniformly distributed errors.

For the $\mathcal{RM}(2, 9)$ code for example, the algorithm corrects up to 120 errors with probability at least 0.99 whereas half the minimum distance is 64. Under the above assumption, we can correct up to half the block length asymptotically for fixed r.

7.1 Introduction

The Reed-Muller codes [13] are a class of codes with a simple construction given by Muller [9] that allows easy majority logic decoding as described by Reed [14].

At first glance, the structure seems to be very simple and the minimum distance is in fact also easily derived, but beyond this there are still several unanswered questions. The weight distribution and the covering radius, for example, are known only for specific instances of the codes.

Most known algorithms for decoding Reed-Muller codes do not decode beyond half the minimum distance and those that do correct more errors do not

[1] Department of Mathematics, Building 303, Technical University of Denmark, DK-2800 Lyngby, Denmark. Email: agnes@math.uni-sb.de, T.Jakobsen@mat.dtu.dk.

During part of this work Agnes E. Heydtmann was supported by a graduate student scholarship (HSPIII) of the German Academic Exchange Service.

have an efficient running time in the general case. Sudan's recent algorithm [18] allows efficient decoding of *Reed-Solomon* codes beyond half their minimum distance. A generalization to some classes of algebraic-geometry codes has been made by Shokrollahi and Wasserman [16]. As there are several similarities in the construction of Reed-Solomon and Reed-Muller codes, it is natural to speculate whether Sudan-like techniques can be applied to the Reed-Muller case.

This article provides such a generalization. Although the algorithm presented has several similarities with Sudan's algorithm, its requirements and proofs are quite different due to the small size of the underlying field. The algorithm also bears some resemblance to the Welch-Berlekamp algorithm for decoding Reed-Solomon codes [2, 21].

The algorithm works best in settings with a low information rate, however the error rate can be correspondingly high. Although when going beyond half the minimum distance, the decoding is not guaranteed to be unique, it is still possible with high probability to correctly determine the codeword sent.

In a sense, decoding of Reed-Muller codes corresponds to the approximation of binary functions by low-degree multivariate polynomials. Consequently, a decoding algorithm which goes beyond half the minimum distance is useful in other applications which are not directly related to error correction. For instance, cryptanalysis of secret-key ciphers [8] often deals with finding approximations, see e.g. [4]; here our algorithm could prove to be very useful. In fact, the research resulting in this article was partly initated because of an observation by Shimoyama and Kaneko [15] who realized that a specific quadratic relation over an S-box of the Data Encryption Standard (DES) [8, 10] has the best linear approximation as a factor.

The article is organized as follows. We proceed in Section 7.2 by defining the Reed-Muller codes and related algebraic concepts. In Section 7.3, the algorithm is presented together with main results and we give some conjectures about its running time and correctness for *random* error patterns. The conjectures are related to the (currently unknown) weight distribution of Reed-Muller codes and they are supported by an implementation of the algorithm and known weight-distributions for short-length Reed-Muller codes as well as asymptotic results. In Section 7.4 we go on to prove the correctness of the algorithm. Section 7.5 concerns the error-correcting capabilities of the algorithm and Section 7.6 covers complexity issues. We conclude in Section 7.7.

7.2 Preliminaries

We begin by defining Reed-Muller codes in terms of elements of a quotient ring.

Definition 7.1

1. *The* ring of boolean polynomials *in m variables $x_1, \ldots, x_m$ is the quotient ring*

$$R_{[x_1,\ldots,x_m]} = \mathbb{F}_2[x_1, \ldots, x_m] \,/\, \langle x_1^2 + x_1, \ldots, x_m^2 + x_m \rangle .$$

Elements of the ring are called boolean polynomials *and in our notation we naturally identify cosets and polynomials in* $\mathbb{F}_2[x_1, \ldots, x_m]$ *that are linear in each of the* m *variables. Furthermore, monomials in* $\mathbb{F}_2[x_1, \ldots, x_m]$ *that are linear in each of the* m *variables and their cosets in* $R_{[x_1,\ldots,x_m]}$ *are called* boolean monomials.

2. *The* degree *of a coset* $f \in R_{[x_1,\ldots,x_m]}$ *denoted by* $\deg f$*, is the degree of the unique element that is linear in each of the* m *variables (i.e., the maximum number of variables occurring in any single term of* f*).*

3. *The* r-th order Reed-Muller code *of length* $n = 2^m$ *with* $0 \leq r \leq m$*, is the set*

$$\mathcal{RM}(r, m) = \left\{ (f(P_1), \ldots, f(P_{2^m})) \mid f \in R_{[x_1,\ldots,x_m]}, \deg f \leq r \right\}$$

where $P_1, \ldots, P_{2^m}$ *are the distinct elements of* $\mathbb{F}_2^m$*. We associate naturally* $f \in R_{[x_1,\ldots,x_m]}$*,* $\deg f \leq r$ *with* $\mathcal{RM}(r, m)$ *words.*

Note that each coset contains exactly one polynomial which is linear in each of the m variables and that all elements of one coset in $R_{[x_1,\ldots,x_m]}$ are equal when considered as functions $\mathbb{F}_2^m \to \mathbb{F}_2$. Therefore $\mathcal{RM}(r, m)$ codes are well defined.

The boolean monomials in $R_{[x_1,\ldots,x_m]}$ evaluate to linearly independent vectors in $\mathbb{F}_2^{2^m}$ and those of degree r result in codewords of minimal weight. Thus the $\mathcal{RM}(r, m)$ code is a linear $[2^m, \binom{m}{0} + \cdots + \binom{m}{r}, 2^{m-r}]$ code.

The ring of boolean polynomials $R_{[x_1,\ldots,x_m]}$ is not even an integral domain as $h \cdot (h + 1) = 0$ for any $h \in R_{[x_1,\ldots,x_m]}$. Therefore the ring is certainly not a unique factorization domain. Nevertheless we define the following:

Definition 7.2 *Let* $R_{[x_1,\ldots,x_m]}$ *be a ring of boolean polynomials and let* $h, f \in R_{[x_1,\ldots,x_m]}$*. The boolean polynomial* f *is said to be a* factor *of* h *or equivalently to* divide h*, if there exists* $g \in R_{[x_1,\ldots,x_m]}$ *such that*

$$h = f \cdot g.$$

The following theorem easily characterizes the factors of a boolean polynomial.

Theorem 7.3 *A boolean polynomial* $f \in R_{[x_1,\ldots,x_m]}$ *divides a boolean polynomial* h *if and only if*

$$(f + 1) \cdot h = 0 \quad \Longleftrightarrow \quad (f + 1) \in \mathrm{Ann}(h) = \left\{ a \in R_{[x_1,\ldots,x_m]} \mid a \cdot h = 0 \right\}.$$

Here $\mathrm{Ann}(\cdot)$ denotes the annihilator of a boolean polynomial. This implies that the factors of a boolean polynomial form a coset of an ideal in $R_{[x_1,\ldots,x_m]}$. A factorization algorithm for multivariate, boolean polynomials is immediately obtained: Simply solve the corresponding set of linear equations in the coefficients of the polynomial factor. In this way it is even straight forward to restrict the solutions to be bounded in their degree.

Proof

$\Rightarrow$: Assume that there exists a boolean polynomial g such that $h = f \cdot g$. Then

$$(f+1) \cdot h = (f+1) \cdot f \cdot g = (f^2 + f) \cdot g = 0$$

where the last equality holds since $(f^2 + f) \in \langle x_1^2 + x_1, \ldots, x_m^2 + x_m \rangle$.
$\Leftarrow$: Assume that $(f+1) \cdot h = 0$. Then $h = fh$, i.e. there exists a boolean polynomial $g = h$ such that $h = fg$ which means that f is a factor of h by the definition. $\square$

We will also use the following standard notions:

Definition 7.4 *Let* $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ *be an* n*-bit word. The* weight *of* $\mathbf{a}$ *denoted by* $\mathrm{w}(\mathbf{a})$ *is the number of non-zero positions in* $\mathbf{a}$*. The* support *of* $\mathbf{a}$ *is the set of its non-zero positions*

$$\mathrm{supp}(\mathbf{a}) = \{i |\, a_i \neq 0\}.$$

Clearly, $\mathrm{w}(\mathbf{a}) = |\mathrm{supp}(\mathbf{a})|$. *We say that a word* $\mathbf{a} \in \mathbb{F}_2^n$ covers *another word* $\mathbf{b} \in \mathbb{F}_2^n$ *or that* $\mathbf{b}$ *is* covered *by* $\mathbf{a}$ *when* $\mathrm{supp}(\mathbf{b}) \subseteq \mathrm{supp}(\mathbf{a})$.

In the following, let $k_\rho = \sum_{i=0}^{\rho} \binom{m}{i}$ be the dimension of $\mathcal{RM}(\rho, m)$. Additionally, let τ denote the maximum number of errors that should be corrected by the algorithm and let ρ be the smallest integer for which

$$\tau \leq k_\rho - 1.$$

Definition 7.5 *Let* $p_0, p_1, \ldots, p_{k_\rho - 1} \in R_{[x_1, \ldots, x_m]}$ *denote* k_ρ *linearly independent, boolean polynomials with* $\deg p_i \leq \rho$*. Let further* $f \in R_{[x_1, \ldots, x_m]}$ *of degree at most* ρ*. Then* f *can be expressed uniquely as*

$$f = \sum_{i=1}^{k_\rho} c_i p_i$$

where $c_i \in \mathbb{F}_2$*. Define by the* order *of* f

$$\mathrm{ord}\, f = \max_i \{i |\, c_i = 1\}.$$

Note that the above does not define an ordering on $R_{[x_1, \ldots, x_m]}$ since the relation is not antisymmetric. Clearly, $\mathrm{ord}\, p_i = i$ and if p_i is a monomial for all i, then the order of a polynomial is simply the maximum order of the monomials in it.

7.3 The Decoding Algorithm

We now describe the algorithm and the main results.

Suppose we receive $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ corresponding to a $\mathcal{RM}(r, m)$ word with at most τ errors. Under certain conditions, the algorithm then correctly decodes the received word.

Algorithm 1 Reed-Muller Decoder

Input: $\mathbf{y}$, r, ρ, and m

Output: All boolean polynomials f corresponding to $\mathcal{RM}(r,m)$ codewords closest to $\mathbf{y}$.

1: Choose $p_0, p_1, \ldots, p_{k_\rho - 1} \in R_{[x_1,\ldots,x_m]}$ linearly independent with $\deg p_i \leq \rho$.
2: Find $Q = yQ_1 + Q_2 \in R_{[y,x_1,\ldots,x_m]}$, $Q_1, Q_2 \in R_{[x_1,\ldots,x_m]}$ with $\operatorname{ord} Q_1$ lowest possible and $\deg Q_2 \leq r + \rho$ such that $Q(P_i, y_i) = 0$ for all i.
3: Find f with $\deg f \leq r$ such that $f(P_i) = y_i, \forall\, i : Q_1(P_i) = 1$.
4: Return those f that correspond to codewords closest to $\mathbf{y}$.

Theorem 7.6 (Correctness.) *Suppose that $f_{\mathbf{c}} \in R_{[x_1,\ldots,x_m]}$ corresponds to a $\mathcal{RM}(r,m)$ codeword $\mathbf{c}$ closest to $\mathbf{y}$. If the error pattern $\mathbf{e} = \mathbf{c} + \mathbf{y}$ does not cover a non-zero $\mathcal{RM}(r+\rho, m)$ codeword and $\mathrm{w}(\mathbf{e}) \leq \tau < k_\rho$, then the algorithm returns $f_{\mathbf{c}}$.*

For the proof refer to Section 7.4.

Theorem 7.7 (Efficiency.) *In Step 1, let $p_0 = 1$ and let p_i, $1 \leq i \leq k_\rho - 1$ be chosen randomly, let also the error pattern be random satisfying the conditions in Theorem 7.6. Assume that the conditions in Theorem 7.6 are satisfied. In addition, let π_1 denote the probability that the $\mathcal{RM}(\rho, m)$ codeword corresponding to the polynomial Q_1 obtained in Step 2 of the algorithm is covered by some $\mathcal{RM}(r,m)$ codeword $\neq (1, 1, \ldots, 1)$.*

Then with probability at least $1 - \pi_1$, the algorithm has a running time of $O(n^3)$.

The theorem is proved in Section 7.6. Coupled with the two following conjectures that are supported by experimental results and considerations about weight distributions, we have an efficient algorithm for decoding Reed-Muller codes beyond half the minimum distance.

Conjecture 7.8 *For $r + \rho \approx \frac{m}{2}$, large m and large $\tau < n - k_{r+\rho}$, a random error pattern of at most τ errors covers a non-zero $\mathcal{RM}(r+\rho, m)$ codeword with probability at most*

$$\pi_2 \leq 2^{k_{r+\rho} - n + \tau}.$$

As we shall see in Section 7.5, the conjecture is true if we can prove that the weights of Reed-Muller codes are close to being binomially distributed. Of course for $\tau < d_{\min}\left[\mathcal{RM}(r+\rho, m)\right] = 2^{m-r-\rho}$, the error pattern can never cover a nonzero $\mathcal{RM}(r+\rho, m)$ codeword, but then we are not beyond half the minimum distance.

Conjecture 7.9 *Let I be a set of at most $k_\rho - 1$ positions and let $\mathbf{q}_1$ denote a randomly chosen $\mathcal{RM}(\rho, m)$ codeword which is zero at the positions in I. Let π_1 denote the probability that $\mathbf{q}_1$ is covered by a $\mathcal{RM}(r,m)$ codeword $\neq (1, 1, \ldots, 1)$.*

If $\rho > r$, then $\pi_1 \to 0$ for $m \to \infty$.

In Section 7.6, we will see that this probability equals the probability mentioned in Theorem 7.7 and we also prove that it equals the probability of having small factors in a random polynomial. It appears that this probability is negligible.

The following is an example of a Reed-Muller code capable of being decoded beyond half the minimum distance.

Example 7.10 *Consider $\mathcal{RM}(1,6)$ which has half minimum distance* 16 *and let $\rho = 2$ which enables us to correct up to* 21 *errors if there is no non-zero $\mathcal{RM}(3,6)$ codeword covered by the error pattern. Suppose we send the codeword* $\mathbf{c}$ *corresponding to $f = x_1 + x_2 + x_4$ and $\mathbf{y} = \mathbf{c} + \mathbf{e}$ is received corresponding to the occurrence of* 20 *errors as indicated by* $\mathbf{e}$*:*

$$\begin{array}{rl} & \mathbf{c}: 01100110100110010110011010011001011001101001100101100110100110001 \\ + & \mathbf{e}: 10000000010100011001000011001010000010100010110000110001000010001 \\ \hline = & \mathbf{y}: 11100110110010001111011001010011011011001011010101010111100100000 \end{array}$$

Choosing the p_i's as monomials of increasing order, Step 2 of the algorithm interpolates the polynomial

$$\begin{aligned} Q = {} & y(x_2x_3 + x_1x_4 + x_3x_4 + x_1x_5 + x_2x_5 + x_3x_5 + x_4x_5 + x_1x_6 + x_2x_6 + x_4) \\ & + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_1x_3x_5 + x_2x_3x_5 + x_3x_4x_5 + x_1x_4x_6 \\ & + x_2x_4x_6 + x_2x_3 + x_1x_4 + x_2x_4 + x_3x_4 + x_1x_5 + x_2x_5 + x_4x_5 + x_1x_6 \\ & + x_2x_6 + x_4 \end{aligned}$$

which has the following unique factorization of the form $(y+f)Q_1$ with $\deg f \leq r$

$$\begin{aligned} Q = {} & (y + x_1 + x_2 + x_4) \\ & (x_2x_3 + x_1x_4 + x_3x_4 + x_1x_5 + x_2x_5 + x_3x_5 + x_4x_5 + x_1x_6 + x_2x_6 + x_4). \end{aligned}$$

The first factor corresponds to the sent codeword $\mathbf{y}$.

The code was chosen to be the smallest example going beyond half the minimum distance. Generally, attempting to decode such a small code much beyond half the minimum distance is not a very good idea because only a few error patterns satisfy the conditions of Theorem 7.6. Later, we discuss an implementation of the $\mathcal{RM}(2,9)$ code which is more robust due to its larger size.

7.4 Correctness of the Algorithm

Assume that some ordering $p_0, \ldots, p_{k_\rho - 1}$ of polynomials of degree at most ρ is fixed as described in Definition 7.5. We will investigate two types of boolean polynomials $\tilde{Q}, Q \in R_{[x_1,\ldots,x_m,y]}$ interpolating the points (P_i, y_i). Therefore we introduce the following notation: Let $f \in R_{[x_1,\ldots,x_m]}$ correspond to a $\mathcal{RM}(r,m)$ codeword at distance $t \leq \tau < k_\rho$ to the received word $\mathbf{y} = (y_1, \ldots, y_{2^m})$ such that there exists no $\mathcal{RM}(r,m)$ codeword closer to $\mathbf{y}$. By $\tilde{Q} = (y + f)\tilde{Q}_1$, $\tilde{Q}_1 \in R_{[x_1,\ldots,x_m]}$ we denote a non-zero boolean polynomial that vanishes at (P_i, y_i) with $\operatorname{ord} \tilde{Q}_1$ smallest possible. Note that $\deg \tilde{Q}_1 f \leq \rho + r$ and $\tilde{Q}_1(P_i) = 0$ whenever $f(P_i) \neq y_i$. The polynomial $\tilde{Q}_1$ may be thought of as an error locator.

By $Q = yQ_1 + Q_2$, $Q_1, Q_2 \in R_{[x_1,\ldots,x_m]}$ we denote another non-zero boolean polynomial that vanishes at (P_i, y_i) with $\operatorname{ord} Q_1$ smallest possible and $\deg Q_2 \leq \rho + r$. Both Q and $\tilde{Q}$ are uniquely determined and we will proceed to show that, under some conditions, the two polynomials are in fact identical.

Theorem 7.11 *If the error pattern* $\mathbf{e} = (y_1 + f(P_1), \ldots, y_{2^m} + f(P_{2^m}))$ *does not cover a* $\mathcal{RM}(m, \rho + r)$ *word, then* $Q = \tilde{Q}$ *and is thereby unique. Also,* $\operatorname{ord} Q_1 \leq \tau$, *where* τ *is the number of errors.*

In the following let $\operatorname{supp}(\mathbf{e}) = \{l_1, \ldots, l_t\}$ denote the error positions and $I_j = \{l_{j+1}, \ldots, l_t\}$ for $0 \leq j < t$. Denote by $\tilde{Q}^{(j)} = (y+f)\tilde{Q}_1^{(j)}$, $\tilde{Q}_1^{(j)} \in R_{[x_1,\ldots,x_m]}$ the boolean polynomial that vanishes at (P_i, y_i) for $i \notin I_j$ (i.e., at the first j error positions and at every non-erroneous position) with $\operatorname{ord} \tilde{Q}_1^{(j)}$ smallest possible. Analogously, denote by $Q^{(j)} = yQ_1^{(j)} + Q_2^{(j)}$, $Q_1^{(j)}, Q_2^{(j)} \in R_{[x_1,\ldots,x_m]}$ the boolean polynomial that vanishes at (P_i, y_i) for $i \notin I_j$ with $\operatorname{ord} Q_1^{(j)}$ smallest possible and $\deg Q_2^{(j)} \leq \rho + r$. Both $\tilde{Q}^{(j)}$ and $Q^{(j)}$ are uniquely determined because of their minimality.

Lemma 7.12 *Suppose* $Q^{(j)} = \tilde{Q}^{(j)}$ *with* $\operatorname{ord} Q_1^{(j)} = \operatorname{ord} \tilde{Q}_1^{(j)} = j$.

1. *If* $Q^{(j)}(P_i, y_i) = 0$ *for all* $i \in I_j$, *then* $Q^{(j)} = Q = \tilde{Q}$.
2. *If* $Q^{(j)}(P_i, y_i) \neq 0$ *for some* $i \in I_j$ *(we may assume without loss of generality that* $i = l_{j+1}$*), then* $Q^{(j+1)} = \tilde{Q}^{(j+1)}$ *with* $\operatorname{ord} Q_1^{(j+1)} = \operatorname{ord} \tilde{Q}_1^{(j+1)} = j + 1$.

Proof We consider the two cases.

1. If $Q^{(j)}(P_i, y_i) = 0$ for all $i \in I_j$, then $Q^{(j)} = Q$. and by assumption also $Q^{(j)} = \tilde{Q}^{(j)}$ implying $\tilde{Q}^{(j)} = \tilde{Q}$. All in all, we get $Q^{(j)} = Q = \tilde{Q}$.

2. If $Q^{(j)}(P_i, y_i) \neq 0$ for an $i \in I_j$, then without loss of generality we may assume that $i = l_{j+1}$.

   Consider the polynomials $Q^{(j+1)}$ and $\tilde{Q}^{(j+1)}$. Note that $\tilde{Q}_1^{(j+1)}(P_i) = 0$ for $i = l_1, \ldots, l_j$ and the j points can be interpolated with the $j + 1$ polynomials $p_0, \ldots, p_j$. Further, $\operatorname{ord} \tilde{Q}_1^{(j+1)} < j + 1$ would contradict the uniqueness of $\tilde{Q}^{(j)}$. Therefore $\operatorname{ord} \tilde{Q}_1^{(j+1)} = j + 1$. On the other hand if $\operatorname{ord} Q_1^{(j+1)} < j+1$, it would be in contradiction to $Q^{(j)}$'s uniqueness. Also $\operatorname{ord} Q_1^{(j+1)} > j + 1$ is impossible as $\tilde{Q}^{(j+1)}$ is of the same type as $Q^{(j+1)}$, but with $\operatorname{ord} \tilde{Q}_1^{(j+1)} < \operatorname{ord} Q^{(j+1)}$. I.e., $\operatorname{ord} Q_1^{(j+1)} = j + 1$ also.

   Suppose now that $Q^{(j+1)}$ and $\tilde{Q}^{(j+1)}$ are distinct polynomials and consider their non-zero sum $Q' = Q^{(j+1)} + \tilde{Q}^{(j+1)} = yQ_1' + Q_2'$ with $Q_1', Q_2' \in R_{[x_1,\ldots,x_m]}$. Since both $Q_1^{(j+1)}$ and $\tilde{Q}_1^{(j+1)}$ have order $j + 1$, the p_{j+1} terms cancel each other and we have $\operatorname{ord} Q_1' < j+1$. In addition, we have

$Q' \neq Q^{(j)}$ as their values at $(P_{l_{j+1}}, y_{l_{j+1}})$ are distinct, but they both vanish for (P_i, y_i), $i \notin I_j$ which contradicts the uniqueness of $Q^{(j)}$. Therefore we must have the equality $Q^{(j+1)} = \tilde{Q}^{(j+1)}$ with $\operatorname{ord} Q_1^{(j+1)} = \operatorname{ord} \tilde{Q}_1^{(j+1)} = j+1$.

□

Proof *(of Theorem 7.11)* With the help of Lemma 7.12, we prove the theorem by induction on the number $j = 0, \ldots, t$ of error positions that are interpolated into $Q^{(j)}$ in addition to the non-erroneous positions.

$j = 0$: Obviously we can choose $\tilde{Q}^{(0)} = (y+f) \cdot 1$ and $Q^{(0)}$ must be of the form $Q^{(0)} = y + h$ with $h \in R_{[x_1,\ldots,x_m]}$, $\deg \leq \rho + r$. Suppose $Q^{(0)} \neq \tilde{Q}^{(0)}$. Then $f(P_i) + h(P_i) = 0$ when i is is a non-erroneous position, but $f + h \neq 0$ which implies that the support of the non-zero $\mathcal{RM}(m, \rho + r)$ word corresponding to $f + h$ is contained in the support of the error pattern $\mathbf{e} = (y_1 + f(P_1), \ldots, y_{2^m} + f(P_{2^m}))$. This contradicts our assumptions and therefore $Q^{(0)} = \tilde{Q}^{(0)}$.

$j \to j+1$: Suppose we have $Q^{(j)} = \tilde{Q}^{(j)}$ with $\operatorname{ord} Q_1^{(j)} = \operatorname{ord} \tilde{Q}_1^{(j)} = j$. By Lemma 7.12 this implies that either $Q^{(j)} = Q = \tilde{Q}$ or (renaming the remaining error positions such that $Q^{(j)}(P_{l_{j+1}}, y_i) = 1$) we have $Q^{(j+1)} = \tilde{Q}^{(j+1)}$ with $\operatorname{ord} Q_1^{(j+1)} = \operatorname{ord} \tilde{Q}_1^{(j+1)} = j+1$.

After having interpolated up to (P_{l_t}, y_{l_t}), we are done and obtain $Q = \tilde{Q}$ with $\operatorname{ord} Q_1 \leq \tau$. The order in which we interpolate does not matter as we have now proven the uniqueness of Q, and the result follows. □

We can now prove that the algorithm works in the sense of Theorem 7.6.

Proof (*of Theorem 7.6*) According to Theorem 7.11, when Step 2 is finished we have

$$Q = yQ_1 + Q_2 = (y + f_{\mathbf{c}})Q_1. \tag{7.1}$$

As mentioned earlier, the polynomial Q_1 may be thought of as an error locator since $Q_1(P_i) = 1$ implies that position i is error-free. Consequently, $f_{\mathbf{c}}$ can be obtained by interpolating a polynomial going through the received bits that are known to be correct. In other words, $f_{\mathbf{c}}$ is a solution to the equation in Step 3. □

7.5 Error Correction Capability

In the following, a *binomial-weight code* is a code which has a binomial distribution of weights, i.e. a binomial-weight $[n, k]$ code has weight enumerators $A_i \approx 2^{k-n}\binom{n}{i}$. The following theorem shows that for binomial-weight codes, it is improbable that a random word of low weight covers a non-zero codeword.

Theorem 7.13 *Let C denote a binomial-weight $[n,k]$ code and let π_2 denote the probability that a random error pattern of length n with at most τ errors covers a non-zero codeword from C. Then the following upper bound on π_2 holds:*

$$\pi_2 \leq 2^{k-n+\tau}.$$

Proof Counting error patterns with the desired property by choosing existing codewords and complementing the support by adding additional non-zero bits, we get the following bound

$$\begin{aligned}
\pi_2 &= \frac{|\{\mathbf{e} \in \mathbb{F}_2^n | \mathrm{w}(\mathbf{e}) \leq \tau, \exists\, \mathbf{c} \in C, \mathbf{c} \neq 0 : \mathrm{supp}(\mathbf{c}) \subseteq \mathrm{supp}(\mathbf{e})\}|}{|\{\mathbf{e} \in \mathbb{F}_2^n | \mathrm{w}(\mathbf{e}) \leq \tau\}|} \\
&\leq \left(\sum_{t=1}^{\tau} \sum_{w=d_{\min}}^{t} A_w \binom{n-w}{t-w}\right) \cdot \left(\sum_{t=0}^{\tau} \binom{n}{t}\right)^{-1} \qquad (7.2) \\
&\leq 2^{k-n} \left(\sum_{t=0}^{\tau} \frac{n!}{t!(n-t)!} \sum_{w=d_{\min}}^{t} \frac{t!}{w!(t-w)!}\right) \cdot \left(\sum_{t=0}^{\tau} \binom{n}{t}\right)^{-1} \\
&\leq 2^{k-n} \left(\sum_{t=0}^{\tau} \binom{n}{t} \sum_{w=0}^{t} \binom{t}{w}\right) \cdot \left(\sum_{t=0}^{\tau} \binom{n}{t}\right)^{-1} \\
&\leq 2^{k-n+\tau}.
\end{aligned}$$

□

Consequently, one way of proving Conjecture 7.8 would be by proving Reed-Muller codes to be almost binomial-weight for $r \approx \frac{m}{2}$ and large m, τ.

The binomial weight distribution hypothesis for Reed-Muller codes is supported by actual weight distributions for several short-length Reed-Muller codes as well as a theorem by Sidel'nikov [7, 17]. Unfortunately, Sidel'nikov's theorem converges too slowly to prove the assumption. Random codes and random linear codes, and consequently almost all codes, are approximately binomial-weight. Sidel'nikov's theorem shows that even for some non-random, large codes, the weight distribution is approximately binomial.

Theorem 7.14 *(Sidel'nikov.) Let C be an $[n,k,d]$ binary code with weight distribution $(A_0, A_1, \ldots, A_n)$, define $a_i = 2^{-k} A_i$, and let $d' \geq 3$ be the minimum distance of the dual code $C^{\perp}$. Furthermore, let $A(z)$ denote the* cumulative distribution function *defined by*

$$A(z) = \sum_{i \geq \mu - \sigma z}^{n} a_i$$

where $\mu = \sum_{i=0}^{n} i a_i$ denotes the mean weight of the code and $\sigma^2 = \sum_{i=0}^{n} (\mu - i)^2 a_i$ denotes the variance. Then

$$|A(z) - \Phi(z)| \leq \frac{20}{\sqrt{d'}}$$

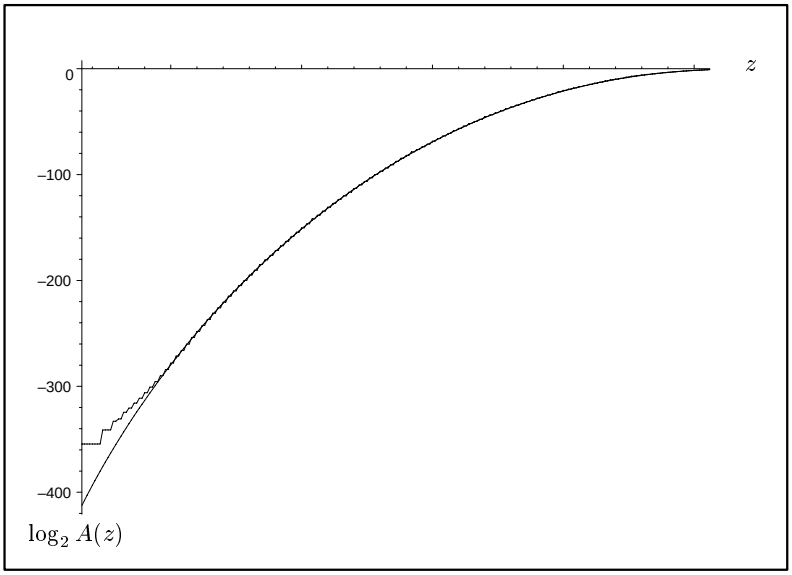

Figure 7.1: The cumulative weight distribution for a binomial-weight code (lower graph) and the $\mathcal{RM}(9,5)$ code (upper graph).

where Φ is the cumulative normal distribution function.

For binary codes $\sigma^2 = \frac{n}{4}$ and for Reed-Muller codes $\mu = \frac{n}{2}$. For specific codes, the convergence is often much faster ([7], Ch. 9, §10), but in general the weight distribution of Reed-Muller codes is not known.

We now demonstrate that the code $\mathcal{RM}(9,5)$ is almost binomially distributed. Figure 7.1 shows the cumulative binomial distribution and the cumulative weight distribution of $\mathcal{RM}(9,5)$ which is known, cf. [20]. The two distributions follow each other closely with a notable gap for small weights only.

Using inequality (7.2), one can derive an accurate upper bound for π_2 from a known weight distribution. In Figure 7.2 this bound is compared for the $\mathcal{RM}(9,5)$ code with the bound of Theorem 7.13 which holds if the code is binomial-weight. As one can see, the two bounds follow each other closely indicating that even in this short-length case, the assumption holds.

From Theorem 7.13 we get the following corollary that gives us an upper bound on the number of errors that can be corrected.

Corollary 7.15 *Let C be a binomial-weight $[n,k]$ code and let a random word* **e** *of weight at most τ be given. If*

$$\tau \leq n - k - \lambda,$$

then the probability that **e** *covers a non-zero word from the code is at most $2^{-\lambda}$.*

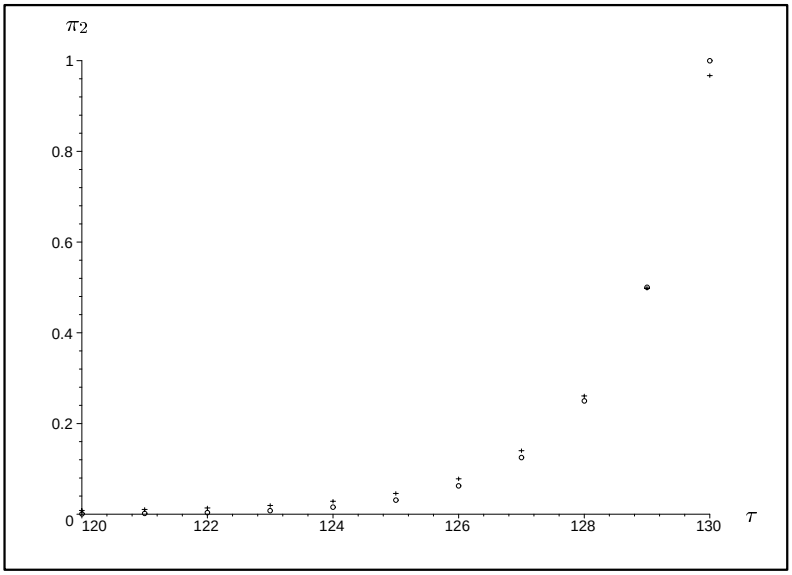

Figure 7.2: Probability bounds for a binomial-weight code '$\circ$' (cf. Theorem 7.13) and the $\mathcal{RM}(9,5)$ code '+'.

We can now state the following sufficient conditions for the algorithm to decode correctly.

Theorem 7.16 *Consider the Reed-Muller code $\mathcal{RM}(r,m)$ and assume that there are at most τ random errors which are independently and uniformly distributed. If*

$$\tau \leq \max_{\rho} \min \left\{ 2^m - \sum_{i=0}^{r+\rho} \binom{m}{i} - \lambda, \sum_{i=0}^{\rho} \binom{m}{i} - 1 \right\}. \tag{7.3}$$

and Conjecture 7.8 holds true, then the algorithm decodes correctly with probability of failure at most $2^{-\lambda}$.

Proof The proof follows by uniting the bound of Corollary 7.15 and $\tau < k_\rho$ which is a prerequisite of Theorem 7.11. $\square$

Note that if $(y+f)Q_1 = Q$ with $\deg f \leq r$ and $\rho \leq r$, then also $(y+f+Q_1+1)Q_1 = Q$. So the condition that $\rho > r$ in Conjecture 7.9 makes sense.

Observe that the right-hand side of (7.3) never exceeds $n/2$ (as expected for binary codes). The following theorem shows that we get close to $n/2$ asymptotically.

Theorem 7.17 *Consider the code $\mathcal{RM}(r,m)$ for fixed r and m going towards infinity. Let τ_m be the maximum number of errors that we can correct by the algorithm for the particular value of m as indicated by (7.3). Define the maximum error rate $\varepsilon_m = \tau_m/n$. If Conjecture 7.8 holds, then we have*

$$\lim_{m\to\infty} \varepsilon_m = \frac{1}{2}.$$

In other words, asymptotically the decoder is capable of correcting up to the maximum number of errors possible in this setting.

Proof The result follows by Theorem 7.16 choosing $\rho = \lfloor \frac{m-2}{2} \rfloor - r$. We show that in this case each of the expressions in the minimization converge to $\frac{1}{2}$. The first expression in (7.3) is

$$\begin{aligned}
\tau' &= 2^m - \sum_{i=0}^{r+\rho} \binom{m}{i} - \lambda \\
&= 2^m - \sum_{i=0}^{\lfloor \frac{m-2}{2} \rfloor} \binom{m}{i} - \lambda \\
&\geq 2^{m-1} - \lambda.
\end{aligned}$$

The second expression is

$$\begin{aligned}
\tau'' &= \sum_{i=0}^{\rho} \binom{m}{i} - 1 \\
&\geq \sum_{i=0}^{\lceil \frac{m-2}{2} \rceil} \binom{m}{i} - \sum_{\lfloor \frac{m-2}{2} \rfloor - r + 1}^{\lceil \frac{m-2}{2} \rceil} \binom{m}{i} \\
&\geq 2^{m-1} - (r+1)\frac{2^m}{\sqrt{2\pi m}}.
\end{aligned}$$

The last inequality follows by the fact that there are at most $r+1$ terms in the last binomial sum and that

$$\binom{n}{\nu n} < \frac{1}{\sqrt{2\pi n \nu \mu}} \nu^{-\nu n} \mu^{-\mu n}$$

for positive integer n, $0 < \nu < 1$ and $\mu = 1 - \nu$ (cf. [12], App. A).

For $m \to \infty$ and constant (or sub-exponential) λ, the lower bounds on τ'/n and τ''/n both converge to $\frac{1}{2}$ and since one of the values is always below $\frac{1}{2}$ the result follows. □

7.6 Algorithm Complexity

In the following, the running time of the algorithm is estimated. We will assume that solving a set of linear equations takes time $O(a^3)$ with a being the maximum

of the number of unknowns and the number of equations (using, e.g., Gaussian elimination).

The complexity of Step 1 is negligible.

A straight-forward implementation of Step 2 repeatedly solves the system of equations for increasingly higher values of the order of Q_1. This gives a total complexity of $O(n^4)$ in Step 2. A better approach would use binary search for the point of uniqueness rather than increasing the order by one in each iteration. This yields the complexity $O(n^3 \log n)$. However, as increasing the order of Q_1 simply involves adding yet another equation to the system, an even more efficient approach would be solving the progressively larger set of equations incrementally. This gives a complexity of $O(n^3)$.

Approaches that exploit the structure of the system of equations may lower this complexity (in Sudan's algorithm for Reed-Solomon codes such an approach is possible, see [11] where an $O(sn^2)$ interpolation algorithm is described for low values of s).

In Step 3, solving should be interpreted as "computing a basis for the set of solutions". Let $\mathbf{q}_1$ denote the $\mathcal{RM}(\rho, m)$ word that corresponds to Q_1. As each non-zero bit in $\mathbf{q}_1$ results in a linear equation, the complexity of this step depends on the weight of $\mathbf{q}_1$. In the worst case this gives a complexity of $O(n^3)$.

In the best case, the first k_r non-zero bits of $\mathbf{q}_1$ treated correspond to information bits of a $\mathcal{RM}(r, m)$ word. Then k_r equations suffice to compute the k_r unknown coefficients of f. If the evaluation of Q_1 is derived by a fast Fourier-like transform, the overall complexity becomes $O(k_r^3 + n \log n)$. Under all circumstances, however, the complexity of Step 3 is dominated by Step 2.

The complexity of Step 4 depends on the number of solutions to the equation solved in Step 3. Let ℓ be the dimension of the subspace of solutions to the corresponding homogenous system. Evaluating every solution in all $n = 2^m$ points using a fast Fourier approach then gives a complexity of $O(2^\ell n \log n)$.

The complexity of Step 3 is dominated by Step 2 and consequently the overall running time is $O(2^\ell n \log n + n^3)$. As this is exponential in ℓ, even moderate values of ℓ can give a high running time. If there is only one solution in Step 3, however, ℓ will be *zero* resulting in just a cubic running time. There are strong indications that this is often the case. The conditions for ℓ to be zero follow.

Theorem 7.18 *Let ℓ be the dimension of the subspace of solutions to the corresponding homogenous system in Step 3. Furthermore, let $\mathbf{q}_1$ be the $\mathcal{RM}(\rho, m)$ codeword corresponding to Q_1. The following three cases are then equivalent*

1. *$\ell = 0$, i.e., there is exactly one solution to the equation in Step 3;*
2. *$\mathbf{q}_1$ is not covered by any $\mathcal{RM}(r, m)$ codeword $\neq (1, 1, \ldots, 1)$;*
3. *Q_1 has no factors $\neq 1$ of degree at most r (possibly including Q_1) itself.*

Proof We show each of the following implications by proof of contradiction.

$1 \Rightarrow 2$: Let f be a solution to Step 3, i.e. $(y + f)Q_1 = Q$. Assume that there exists a $\mathcal{RM}(r, m)$ codeword $\mathbf{c}$ corresponding to the polynomial $f_\mathbf{c}$ that

covers $\mathbf{q}_1$. Then $(f_\mathbf{c}+1)Q_1 = 0$ and hence $(y+f+f_\mathbf{c}+1)Q_1 = Q$ implying that $f+f_\mathbf{c}+1$ is another solution to Step 3 which contradicts our assumption.

$2 \Rightarrow 3$: Assume that there is a factor $f_\mathbf{c}$ in Q_1 with $\deg f_\mathbf{c} \leq r$. This implies that $(f_\mathbf{c}+1)Q_1 = 0$. Let $\mathbf{c}$ be the $\mathcal{RM}(r,m)$ codeword corresponding to $f_\mathbf{c}$. Then $\mathbf{c}$ must cover q_1.

$3 \Rightarrow 1$: Assume that Q_1 has no factors of degree at most r. If the equation in Step 3 has two distinct solutions f_1 and f_2, then by Theorem 7.11 $(y+f_1)Q_1 = Q_2$ and $(y+f_2)Q_1 = Q_2$. Therefore $(f_1+f_2)Q_1 = 0$, i.e., f_1+f_2+1 is a factor of Q by Theorem 7.3 which contradicts our assumption.

□

Now we can easily see that 2^ℓ is the number of factors $\neq 1$ in Q_1 of degree at most r.

By choosing the polynomials $p_1, \ldots, p_{k_\rho - 1}$ at random, the polynomial Q_1 will also be random constrained only by the requirements that $Q_1(P_i)$ be zero if i is an error position. This gives Theorem 7.7 as an immediate corollary.

Proof (*of Theorem 7.7*) Follows directly from Theorem 7.18 and the running time considerations above. □

Of course the question is, when does Q_1 have small factors? There will definitely be multiple solutions if $r \geq \rho$ since Q_1 has itself as a factor of degree $\rho \leq r$. The question in the opposite direction is apparently not an easy one to answer. Empirically, however, when $r < \rho$ it seems as if there is practically always only one solution, or in other words, a random Q_1 polynomial has very slim chances of having small factors. The conjecture holds in experiments with higher-order Reed-Muller codes up to length 512. The correct probability does not seem easy to establish or even to estimate although it appears to be very low.

Asking when a completely *random* pattern covers a codeword or vice versa is easier to answer using arguments like those in Theorem 7.13. For self-dual codes (like $\mathcal{RM}(\frac{m-1}{2}, m)$, m odd) one can even upper-bound the probability that a randomly chosen codeword covers another randomly chosen codeword by using generalizations of Gleason's theorem to biweight enumerators [6]. All these probabilities are asymptotically low. We conjecture that the related probability that we seek also goes to zero as the code length increases (Conjecture 7.9).

Even if $\ell >$ is too large to search through all the solutions for a correct polynomial, one might choose other $\hat{p}_0, \ldots, \hat{p}_{k_\rho - 1}$ resulting in another solution to Step 2 and consequently a new system of equations in Step 3. Considering this new system together with the old one, should hopefully decrease the number of solutions. The procedure can of course be iterated to further lower the probability of having more than one solution. This approach works well in practice.

Notice that if one can bound the probability π_1 of having many solutions just non-negligibly away from 1, then one might also be able to construct a

polynomial-time algorithm which fails with negligible probability by simply repeating Steps 1 and 2 with new orderings in the above manner until the combined system of equations in Step 3 has only the closest codewords as solutions.

Furthermore, the algorithm does not have to process all the points. One can stop as soon as Step 3 has an adequate number of solutions. In a situation with side information, a good approach would be to process the most reliable points first. Erasures are dealt with by simply omitting the points in question from the interpolation.

The polynomial p_0 is chosen to be 1 to ensure that no errors result in $Q = y + f$ where f corresponds to the sent codeword. There is a reason for choosing random p_i and not just using monomials in increasing order. If one uses e.g. $p_0 = 1, p_1 = x_1, p_2 = x_2, \ldots, p_{k_\rho - 1} = x_{m-\rho+1} \cdots x_{m-1} x_m$ and there are only few errors, then Q_1 ends up with a small degree which might be less than r. In this case, as we have seen, Step 3 will have wrong solutions. Using monomials actually results in a counter-intuitive situation in which the decoder does not work properly if there are too *few* errors. By injecting additional errors, one can fix this, but the nicer approach is to simply choose the p_i polynomials at random.

Instead of solving the system given in Step 3 to obtain $f_{\mathbf{c}}$, one might instead choose to apply the techniques demonstrated in Theorem 7.3 and simply factor Q by solving $(y + f + 1)Q = 0$. This is similar to the approach taken in Sudan's algorithm. Yet another approach to Step 3 follows from the last identity of (7.1) which gives us $f_{\mathbf{c}} Q_1 = Q_2$; a system of equations which is simpler than the previously mentioned, but with the same set of solutions. This is similar to the equation solved by the Welch-Berlekamp algorithm.

The memory requirements of the algorithm is $O(n^2)$ for storing the various systems of equations.

7.7 Conclusions

The algorithm has been implemented in the computer algebra system Singular [3]. Among other codes, it successfully decodes a $\mathcal{RM}(2, 9)$ code using $\rho = 3$ with up to 120 errors where half the minimum distance is 64. The failure probability π_2 is at most 0.01 by the real weight distribution of $\mathcal{RM}(2 + 3, 9)$ considered in Section 7.5. In each of 10 tries, Step 2 of the algorithm produced only one candidate so no time was wasted filtering away wrong polynomials. The time spent decoding one word was approximately one minute for an unoptimized implementation running on a Pentium 200 MHz under Linux.

The algorithm improves results given in the paper by Ar *et al.* [1] concerning the reconstruction of "noisy multivariate polynomials" in the binary case. It might be possible to extend our method to work also with generalized Reed-Muller codes (which operate on non-binary alphabets).

There exists a more general version [1] of Sudan's algorithm not directly related to coding theory which finds relations $g(x, y) = 0$ that hold with high probability. The same sort of generalization ought to be possible in the Reed-

Muller setting. It would allow the discovery of probabilistic relations of the type $g(x_1, \ldots, x_{m_1}, y_1, \ldots, y_{m_2}) = 0$ as opposed to $y = f(x_1, \ldots, x_{m_1})$.

The algorithm can be used to generalize the cryptanalytical results of [4] and [5] to a kind of probabilistic interpolation attack for bit-oriented block ciphers. Such work is currently in progress. For these purposes it sometimes suffices to know whether there exists a low-degree approximation without actually obtaining it. In this case, the first two steps of the algorithm are enough. If the resulting polynomial Q has a low degree, then there must exist a low degree approximation. The algorithm might also be useful for areas like digital hardware optimization and construction of binary decision trees.

Finally, to verify the conjectures in practice for longer codes of higher order, more simulation results are needed. To achieve this, an optimized, low-level implementation of the algorithm using direct bit manipulations would be useful.

7.8 Acknowledgements

The authors wish to thank Tom Høholdt and Jørn Justesen for invaluable discussions.

Bibliography

[1] Sigal Ar, Richard J. Lipton, Ronitt Rubinfeld, and Madhu Sudan, Reconstructing Algebraic Functions from Erroneous Data, *SIAM Journal on Computing*, 28(2):487–510, April 1999.

[2] Elwyn R. Berlekamp, Bounded Distance +1 Soft-Decision Reed-Solomon Decoding, *IEEE Trans. on Info. Th.*, vol. 42, no. 3, May 1996, pp. 704-720.

[3] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 User Manual. In *Reports On Computer Algebra*, number 21. Centre for Computer Algebra, University of Kaiserslautern, June 1998. `http://www.mathematik.uni-kl.de/~zca/Singular`

[4] Thomas Jakobsen, Cryptanalysis of Block Ciphers with Probabilistic Nonlinear Relations of Low Degree. In Hugo Krawczyk (Ed.): *Advances in Cryptology - CRYPTO '98*, Proceedings. Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 212–222.

[5] Thomas Jakobsen and Lars Knudsen, Attacks on Block Ciphers of Low Algebra Degree, submitted to *Journal of Cryptlogy*, 1999. Preliminary version: The Interpolation Attack on Block Ciphers, in *Proc. Fast Software Encryption '97*, Lecture Notes in Computer Science, vol. 1267, Springer, 1997.

[6] F. Jessie MacWilliams, Colin L. Mallows, and Neil J. A. Sloane, Generalizations of Gleason's Theorem on Weight Enumerators of Self-Dual Codes, *IEEE Trans. on Info. Th.*, vol. IT-18, No. 6, November 1972.

[7] F. Jessie MacWilliams and Neil J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1981.

[8] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press Series on Discrete Mathematics and Its Applications, 1996.

[9] D. E. Muller, Application of Boolean algebra to switching circuit design and to error detection, *IEEE Trans. Comput.* **3** (1954), 6–12.

[10] National Institute of Standards and Technology (NIST). FIPS Publication 46-1: *Data Encryption Standard*. January 22, 1988. Originally issued by National Bureau of Standards.

[11] Rasmus Refslund Nielsen and Tom Høholdt, Decoding Reed-Solomon codes beyond half the minimum distance, *Proceedings of the International Conference on Coding Theory, Cryptography, and Related Areas*, Mexico, 1998, Lecture Notes in Computer Science, to appear, Springer.

[12] W. Wesley Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd edition, MIT Press, Cambridge, Massachusetts, 1972.

[13] Richard Brualdi, William Cary Huffman and Vera S. Pless, editors, *Handbook of Coding Theory*, Elsevier, 1998.

[14] I. S. Reed, A class of multiple-error-correcting codes and the decoding scheme, *IRE Trans. Inform. Theory* **IT-4** (1954), 38–49.

[15] Takeshi Shimoyama, Toshinobu Kaneko, Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES. In Hugo Krawczyk (Ed.): *Advances in Cryptology - CRYPTO '98*, Proceedings. Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 200–211.

[16] M. A. Shokrollahi and H. Wasserman, Decoding algebraic-geometric codes beyond the error-correction bound, *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pp. 241–248, 1998.

[17] V. M. Sidel'nikov, Weight spectrum of binary Bose-Chaudhuri-Hocquenghem codes, *Problems of Info. Trans.*, **5**(1), 1969, 12–16.

[18] Madhu Sudan, Decoding of Reed-Solomon codes beyond the error-correction bound, *Journal of Complexity*, 13(1):180–193, March 1997.

[19] Tsukasa Sugita, Tadao Kasami, Toru Fujiwara, The weight distribution of the third-order Reed-Muller code of length 512, *IEEE Trans. on Info. Th.*, Sept. 1996, pp. 1622–1625.

[20] Tsukasa Sugita, Tadao Kasami, Toru Fujiwara, Weight Distributions of the Third and Fifth Order Reed-Muller Codes of Length 512, Nara Inst. Sci. Technol., Tech. Rep. NAIST-IS-TR96006, Feb. 1996, Japan. `http://isw3.aist-nara.ac.jp/IS/TechReport2/report/96006.ps`

[21] Lloyd R. Welch and Elwyn R. Berlekamp, Error correction of algebraic block codes, *US Patent* Number 4,633,470, issued December 1986.

Chapter 8

Analysis of S-boxes and Decoding of Linear Block Codes

Parts of the following chapter have been presented at the Winter School in Coding and Information Theory 1998, Ebeltoft, Denmark and appeared as an abstract in the proceedings with the title "Non-Linear Approximations in Block Ciphers".

Analysis of S-boxes and Decoding of Linear Block Codes

Thomas Jakobsen [1]

Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ denote some function with an m-bit argument and an n-bit value. The function may be thought of as an S-box from a block cipher. This chapter treats, from a coding-theoretical point of view, the problem of trying to determine simple probabilistic relations $p(x, y) = 0$ that hold with a high probability for $y = f(x)$ and a uniformly distributed $x \in \mathbb{F}_2^m$. By constructing an appropriate generator matrix and then finding small-weight codewords in the corresponding code, one can obtain such good relations. Via this relationship it is also described what properties are necessary in a function to ensure it has no good approximations. On the constructive side, Carlitz-Uchiyama's bound and results about Kloosterman sums can be used directly to construct certain good functions or S-boxes where there are no linear relations that hold with high probability (giving resistance against linear cryptanalysis).

Later, with the S-boxes of DES as examples, various techniques for obtaining such relations are demonstrated.

8.1 The Linear Case

We first show how the search for good *linear* relations over some function can be viewed as a decoding problem. Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a function with m input bits and n output bits. Assume that we want to find the best linear approximation over that function, i.e., we want to find a linear combination of input and output bits that has a probability of being 0 (or 1) which is bounded as far away from $1/2$ as possible.

Recall the definition of imbalance as used in [7].

Definition 8.1 *The imbalance of a binary function $f(x) : D \to \mathbb{F}_2$ is defined by*

$$\mathrm{Imb}(f) = 2 \cdot \left| \mathrm{Prob}[f(X) = 0] - \frac{1}{2} \right|$$

for a uniformly distributed argument X.

The problem is finding a linear expression with maximum imbalance. This is equivalent to finding bitmasks a and b that solves the following optimization problem

$$\max_{a \in \mathbb{F}_2^m, b \in \mathbb{F}_2^n} \left| \mathrm{Prob}\left[(\langle a, x \rangle \oplus \langle b, f(x) \rangle) = 0 \right] - \frac{1}{2} \right| \tag{8.1}$$

[1] Department of Mathematics, Building 303, Technical University of Denmark, DK-2800 Lyngby, Denmark. Email: T.Jakobsen@mat.dtu.dk.

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two n-bit vectors over $\mathbb{F}_2$.

Now consider the following $(m + n + 1) \times 2^m$ matrix where vectors are considered to be columns and the ξ_is enumerate all possible binary vectors of length m (for instance, ξ_i may be considered to be the binary representation of i):

$$G_f = \begin{bmatrix} \xi_0 & \xi_1 & \xi_2 & \cdots & \xi_{2^m-1} \\ f(\xi_0) & f(\xi_1) & f(\xi_2) & \cdots & f(\xi_{2^m-1}) \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{8.2}$$

Let C_f denote the code which has G_f as its generator matrix.

Notice that the upper m rows of the matrix define a first-order Reed-Muller code. When f is bijective the next $n = m$ rows also make up a (permuted) first-order Reed-Muller code. Finally, the combined code is extended by the all-one vector.

The following theorem shows that G_f is useful when trying to solve the optimization problem in (8.1) and hence compute the best linear approximation.

Theorem 8.2 *Let G_f be the generator matrix of the linear, binary code C_f constructed from the function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ as described above. Finding the minimum distance $d_{\min}$ of this code is then equivalent to solving the optimization problem given in (8.1).*

Proof Let $x_{\min} = (a_1, a_2, \ldots, a_m, b_1, b_2, \ldots, b_n, c) \in \mathbb{F}_2^{m+n+1}$ denote the information bit row vector that gives rise to a minimum weight codeword $c_{\min}$, i.e. $x_{\min}G = c_{\min}$ where $d_{\min} = \mathrm{w}(c_{\min})$.

Note that each of the first m rows of G represent all the possible values of a specific bit in all the possible arguments of f (i.e., values of ξ_i) and the next n rows each hold a bit of the corresponding function value.

Now it is possible to see directly that the vectors $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_n)$ are indeed the masks that solve the optimization problem in Equation (8.1). This is the case since each of the rows in the matrix may be thought of as function tables (in this case the functions are defined by monomials of degree at most 1). A sum of rows is then actually the function table for the function that is constructed by adding the corresponding monomials, i.e., a linear combination of the involved bits. If the function table has a low weight then the corresponding function has a high imbalance. This observation was mentioned by Wagner [26].

The value c is 0 if the involved probability is larger that $\frac{1}{2}$ and 1 if the probability is less than $\frac{1}{2}$ corresponding to the situation where the expression $\langle a, x \rangle \oplus \langle b, f(x) \rangle$ is most often 0 or 1 (another way to state it, is that we want to find a and b such that $\langle a, x \rangle \oplus \langle b, f(x) \rangle$ is as unbalanced as possible with a bias towards being c most often). □

Instead of considering an optimization involving probabilities, the problem can be viewed as a computation involving certain exponential sums. This immediately becomes apparent when looking at the following identities.

$$
\begin{aligned}
&\mathrm{Imb}(< a, x > \oplus < b, f(x) >) \\
&= \quad 2 \cdot \left| \mathrm{Prob}\left[(< a, x > \oplus < b, f(x) >) = 0\right] - \frac{1}{2} \right| \\
&= \quad \frac{1}{2^m} \left| \sum_{x \in \mathbb{F}_2^m} (-1)^{(< a, x > \oplus < b, f(x) >)} \right| \\
&= \quad \frac{1}{2^m} \left| \sum_{x \in \mathbb{F}_2^m} (-1)^{\mathrm{Tr}(\alpha x + \beta f(x))} \right| \\
&= \quad \frac{1}{2^m} \left| \sum_{x \in \mathbb{F}_2^m} \chi(\alpha x + \beta f(x)) \right|,
\end{aligned}
$$

where α and β are certain unique values in $\mathbb{F}_m$ corresponding to the values of a and b, respectively, and χ is an additive character of $\mathbb{F}$. The last expression is seen to be an exponential sum and Carlitz-Uchiyama's bound tells us that the value is upper-bounded by an expression involving the degree of f.

Theorem 8.3 Carlitz-Uchiyama's bound. *Let $f \in \mathbb{F}_{2^m}[x]$ be a univariate polynomial over a finite field of characteristic 2 such that $f \neq g^2 + g + b$ for all polynomials $g \in \mathbb{F}_{2^m}[x]$ and constants $b \in \mathbb{F}_{2^m}$, and let χ denote some non-trivial additive character of $\mathbb{F}_{2^m}$. Then*

$$
\left| \sum_{x \in \mathbb{F}_{2^m}} \chi(f(x)) \right| \leq (\deg f - 1) q^{1/2}. \tag{8.3}
$$

For a proof consult [5].

The left-hand side of (8.3) is an exponential sum on the required form. Consequently, in order to obtain a low upper-bound we choose an f with a low degree. Quadratic polynomials are an excessively bad choice (not to mention linear polynomials) if the characteristic of the field is 2, since this leaves us with a function whose output bits are strictly linear in the input bits.

A good choice is $f(x) = x^3$ which is exactly the choice of round function taken in the construction of Knudsen and Nyberg's cipher [17]. In their case, the round function was chosen such that their cipher becomes immune against differential (and also linear) cryptanalysis, although in [17] a different proof was used.

Using the above cubic function actually turns the corresponding matrix G_f into the generator matrix of an extended, dual BCH code (and hence the parity check matrix of the corresponding BCH code). A remarkable link between coding theory and cryptology becomes clear: By constructing good codes — that is, codes with a high minimum distance — with a certain structure, we can actually construct S-boxes with nice cryptographic properties.

Note that the problem of decoding the code described by (8.2) may actually be viewed as a problem involving certain correlations. By using the fast Fourier transform (FFT) it becomes possible to solve the problem in time $O(n^2 \log n)$ instead of $O(n^3)$ required in a straight-forward approach (see [10] for a treatment of this).

In a similar fashion it becomes apparent using Kloosterman sums why $f(x) = x^{-1}$ is good choice of S-box to ensure that linear attacks fail. (Kloosterman sums have the form $\sum_x \mathrm{Tr}(\alpha x + \beta x^{-1})$.) For example, the cipher Shark by Rijmen et al. [19] uses the reciprocal function as S-box. This function has an advantage when compared to x^3, namely that output bits have a high non-linear order when considered to be multivariate polynomials in the input bits; this makes the cipher more resistant against attacks using higher-order differentials.

8.2 Nonlinear Approximations

It is possible to extend the coding theoretical interpretation even further to cover also the computation of good *nonlinear* approximations over an S-box. Indeed by just adding additional rows to the generator matrix one can obtain knowledge about higher-order approximations in a very analogous fashion, simply by computing the minimum distance of the resulting code.

In the above example, the set of possible approximations A was simply a linear space spanned by all the linear, bivariate monomials in $B_1 = \{1, x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m\}$, in other words, the set $A = \mathrm{span}B_1$ represents all possible linear approximations. We may choose to define a more general basis containing other functions that we want to consider, for instance all quadratic monomials $B_2 = \{1, x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_m, x_1x_2, x_1x_3, \ldots x_{m-1}x_m, x_1y_1, x_1y_2, \ldots, x_my_n, y_1y_2, y_1y_3, \ldots, y_{n-1}y_n\}$. In this case, the linear space $A = \mathrm{span}B_2$ is then the set of all bivariate, quadratic (or linear) boolean polynomials. (Here, simply consider boolean polynomials to be the multivariate polynomials that have exponents not larger than 1.) For each element of B_2 we add a row to G_f containing the evaluations of the corresponding monomial (or function, more generally) for all possible values of $x = (x_1, x_2, \ldots, x_m) \in \mathbb{F}_2^m$ and $y = (y_1, y_2, \ldots, y_n) = f(x)$. This is quite similar to the construction of Reed-Muller codes. In fact G_f *will* describe a Reed-Muller code if only monomials in x are considered.

Again, the codeword of lowest weight in the corresponding code C_f will correspond directly to the best approximation from A possible over f.

One may ask, whether this coding-theoretical interpretation provides us with cryptanalytical tools and knowledge which can effectively find the best approximation in any sense over any given S-box. Unfortunately, this is not the case in general since Vardy has recently proved the long-time hypothesis that determining the minimum distance of a linear code is NP-complete [25]. However, in the following section we show that it is sometimes possible to find good approximations over S-boxes using algorithms for decoding.

2	14	12	11	4	2	1	12	7	4	10	7	11	13	6	1
8	5	5	0	3	15	15	10	13	3	0	9	14	8	9	6
4	11	2	8	1	12	11	7	10	1	13	14	7	2	8	13
15	6	9	15	12	0	5	9	6	10	3	4	0	5	14	3

Figure 8.1: DES S-box no. 5.

$$
\begin{bmatrix}
0000000000000000000000000000000011111111111111111111111111111111 \\
0000000000000000111111111111111100000000000000001111111111111111 \\
0000000011111111000000001111111100000000111111110000000011111111 \\
0000111100001111000011110000111100001111000011110000111100001111 \\
0011001100110011001100110011001100110011001100110011001100110011 \\
0101010101010101010101010101010101010101010101010101010101010101 \\
0111000100101100100001111001111001010110101100111011100101000010 \\
0110100111010110011001101000100110000101001110011101101010010110 \\
1101010010111010000011101001001011000111001110011010000111000011 \\
0001001010011101011011011010010010010110110100110110011001001101 \\
1111111111111111111111111111111111111111111111111111111111111111
\end{bmatrix}
$$

Figure 8.2: G_{S_5}, the generator matrix corresponding to the fifth S-box of DES.

8.3 Examples of How to Find Approximations

This section is devoted to demonstrating with examples how the coding theoretical interpretation works for the purpose of finding S-box approximations. We analyze the S-boxes of DES since they are small, handy instances of such functions. The S-boxes of DES are nonlinear functions (usually given by tables) each with 6 input bits and 4 output bits.

The contents of S-box no. 5 appear in Figure 8.1. Values should be considered in their binary representations. Beginning with $S_5(0) = 2$ in the upper-left corner, proceeding with $S_5(1) = 14$ rows-first, and ending in the lower-right corner with $S_5(63) = 3$ (for DES, this is a bit unconventional but easier representation of the S-box). According to the specification of the standard, we number bits from left to right.

We now construct the corresponding generator matrix G_{S_5} in order to find the best linear approximation. The matrix appears in Figure 8.2. A brute-force approach evaluating all possible vectors (codewords) in the space described by G_{S_5} (Figure 8.2) soon reveals that the codeword of lowest weight corresponds to the linear approximation $y_1 + y_2 + y_3 + y_4 + x_2$. This is actually the best linear approximation possible, not only over S_5 but over all the S-boxes, a fact discovered by Matsui in his paper [14].

We now turn to the problem of computing nonlinear approximations of low degree. Of course a similar brute-force approach of trying all possible combinations of the appropriate monomials can be applied. However, the increased

number of rows in G_f soon makes it impractical to carry out the search.

In other words, we haven't gained any simplification from the coding theoretical reinterpretation, yet. Until now, it has functioned more like a dictionary between two fields in itself yielding no speed-up over traditional methods.

Consequently, we now try to apply more subtle techniques from the field of coding. We will use a method which is similar to the Sudan-like decoding technique that was the topic of the previous chapter and the article in Chapter 6.

The idea is the following. Given the input/output-pairs that describe the S-box $P = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{64}, y_{64})\}$ (here each $x_i \in \mathbb{F}_2^6$ and $y_i \in \mathbb{F}_2^4$ are vectors) carry out the procedure described below.

1. Construct a multivariate, boolean polynomial Q which vanishes when evaluated at the points in P and has as small a degree as possible. This can be done via simple interpolation techniques by solving a system of linear equations.

2. Factorize Q. There might be several factorizations since the quotient ring of boolean polynomials is not a unique factorization domain.

3. Look for factors in Q that are good approximations, i.e., factors $p(x, y)$ such that $\text{Prob}[p(x, y) = 0]$ is nonnegligibly bounded away from $1/2$.

For definitions and properties about boolean polynomials (in particular factoring) see the previous chapter.

Just as Sudan's algorithm for Reed-Solomon codes easily generalizes to allow the discovery of more general bivariate relations $p(x, y) = 0$ instead of just relations on the form $y = p(x)$, it turns out that the above approach produces several good Reed-Muller-like approximations.

A number of such approximations are listed in Table 8.1. For each S-box of DES, the type of the best approximation found by the above approach is listed together with the corresponding imbalance and the degree of the Q-polynomial in which the approximation was a factor. For some S-boxes the Q-polynomials of lowest degree are quadratic and in other cases they are cubic. This was noted in [22] where the authors tried to obtain these quadratic Q-polynomial for another purpose than our. They found the polynomials by Gröbner base calculations although such complicated procedures are not necessary at all since interpolation suffices.

The list does not represent the best approximations in existence since the approach is not guaranteed to find these. Notice, however, that the best existing linear approximation of S_5 found by Matsui also appears on the list.

Shimoyama and Kaneko were the first to notice that a certain Q-polynomial over S_5 had Matsui's best linear relation as a factor but they pursued the fact no further and did not try to explain it (they actually factored the polynomial by hand [21] since no factorization method of multivariate, boolean polynomials was known at that time!). The work in this chapter on finding nonlinear approximations by factoring was initiated mostly because of their remark — the

S-box	Relation	Imbalance	$\deg Q$
1	QE	32/32	2
	LA	18/32	2, 3
	QA	20/32	2, 3
2	QA	16/32	3
3	QA	22/32	3
4	QE	32/32	2
	LA	16/32	2, 3
	QA	24/32	2, 3
5	QE	32/32	2
	LA*	20/32	2, 3
6	LA	14/32	3
	QA	16/32	3
7	QA	16/32	3
8	LA	12/32	3
	QA	19/32	3

Table 8.1: Examples of equations and approximations over the S-boxes of DES. QE: Quadratic equation. LA: Linear approximation. QA: Quadratic approximation. $*$: Best linear approximation in DES as discovered by Matsui.

fact that the best linear approximation appeared as a factor of a low-degree interpolated polynomial struck us as something which could not be a coincidence.

We explain in the following sufficient conditions under which there exist Q-polynomials that factor into good approximations.

Definition 8.4 *Define by the* boolean ideal I_v *corresponding to the variables in the list v, the ideal constructed from a variety involving the variables described in the index as follows:*

$$I_{x_1,\ldots,x_n} = \left\langle x_1^2 + x_1, \ldots, x_n^2 + x_n \right\rangle.$$

Denote by the ring of boolean polynomials corresponding to the list of variables v, the quotient ring

$$R_v = \mathbb{F}_2[v]/I_v.$$

Theorem 8.5 *Let $L = [l^{(1)}, \ldots, l^{(n)}]$ be a list of s-bit vectors, $l^{(i)} \in \mathbb{F}_2^s$. Furthermore, let $T \subset R_{[x_1,\ldots,x_s]}$ be the set of all nonzero, boolean polynomials in s variables such that $Q(l^{(i)}) = 0$ for all i.*

Let $T_d \subset T$ be the subset of polynomials with degree at most d and let $g_1 \in R_{[x_1,\ldots,x_{s_1}]}$ and $g_2 \in R_{[x_{s_1+1},\ldots,x_s]}$ with $s_1 + s_2 = s$ describe an approximation of L in the sense that $g_1(l_1^{(i)}, \ldots, l_{s_1}^{(i)}) = g_2(l_{s_1+1}^{(i)}, \ldots, l_s^{(i)})$ for σ values of i. Let $\tau = n - \sigma$ denote the number of values of i for which the approximation does not hold. Assume without loss of generality that $s_1 \geq s_2$ and let $\gamma = \deg(g_1 + g_2)$.

If

$$\tau < \sum_{i=0}^{d-\gamma} \binom{s_1}{i}$$

then there exists a polynomial $Q \in T_d$ having $g_1 + g_2$ as a factor.

The list L may contain vectors on the form $l = (\mathbf{x}, \mathbf{y})$ where $\mathbf{y} = f(\mathbf{x})$. The theorem can then be used to find approximations over f.

The theorem is a generalization of the results from Chapter 7 in the sense that when $s = m$, $s_1 = m - 1$, $s_2 = 1$, and $L = [(P_1, y_1), \ldots, (P_n, y_n)]$ then the theorem specializes to the Reed-Muller case previously considered.

Note that a similar theorem holds if we do not split the bits into the first s_1 bits and the remaining s_2. Any partitioning in s_1 and s_2 positions can be used.

Proof We show how to construct a Q with the given properties such that $Q = gh$ for an appropriate choice of h. Define $g \in \mathbb{F}_2[x_1, \ldots, x_s]$ by $g = g_1 + g_2$ and let $Q = gh$ where $h \in \mathbb{F}_2[x_1, \ldots, x_{s_1}]$ with $\deg h \leq d - \deg(g_1 + g_2)$. It follows that $\deg Q \leq d$.

We need to show that there exists an h such that $Q(l^{(i)}) = 0$ for all values of i. Independently of h, however, the expression $Q(l^{(i)})$ already vanishes for at least σ values of i due to g, i.e., at most τ values of i remain for which we require $Q(l^{(i)}) = 0$.

The number of coefficients in h is at most $\sum_{i=0}^{\deg h} \binom{s_1}{i}$. If the number of possible coefficients is larger than the number τ of additional zeros required, then we can construct h by interpolation such that Q has the given properties.

We also need to show that $Q \neq 0$. This follows from the fact that g_2 does not have any variables in common with h, i.e. $g_2 h \neq 0$. Since g_1 can contain only the same variables as h (and not g_2), we have $g_1 h + g_2 h = gh \neq 0$. The theorem follows directly from this. □

Notice that the theorem does not say how many of the Q-polynomials that factor. Among other things, this depends on the exact imbalance of the approximation since a higher imbalance implies a lower value of τ and a correspondingly larger set of solutions to the set of linear equations determined Q.

Also notice that the factorization result of Theorem 8.5 sometimes holds for higher values of τ than those allowed by the bound. This happens if there are linear dependencies in the set of linear equations solved when constructing Q; this allows us to have less unknowns than linear equations and still obtain a nontrivial solution.

Corollary 8.6 *Let a DES-like S-box $f : \mathbb{F}_2^6 \to \mathbb{F}_2^4$ with 6 inputs and 4 outputs be given. Furthermore, let $x^{(i)}$ for $i = 1, \ldots, 64$ denote the possible values of its input argument.*

Let T_2 be the set of all quadratic, boolean polynomials Q in 10 variables that satisfy the equation $Q(x^{(i)}, f(x^{(i)})) = 0$, $i = 1, 2, \ldots, 64$. Similarly, let T_3 be the set of all cubic, boolean polynomials with the same property.

Let z denote one of the inputs $x_1, \ldots, x_6$ or one of the outputs $y_1, \ldots, y_4$. With a slight abuse of the notation, let $R' = R_{[x_1,\ldots,y_4]\setminus\{z\}}$ denote the set of boolean polynomials in the remaining variables.

If z is expressible as a linear approximation $h_1 \in R'$ in the remaining variables with imbalance at least $22/32$ then one of the polynomials in T_2 will have $z + h_1$ as a factor.

If there is a quadratic approximation $z = h$ over f for $h_2 \in R'$ with imbalance at least $22/32$ then one of the polynomials in T_3 will have h_2 as a factor.

Proof The corollary follows from Theorem 8.5 using $s_1 = 9$ and $s_2 = 1$. In both cases, $d - \deg g_1 + g_2 = d - 1$ and we get

$$\tau \leq \sum_{i=0}^{2} \left(\begin{array}{c} 9 \\ i \end{array} \right) = \left(\begin{array}{c} 9 \\ 0 \end{array} \right) + \left(\begin{array}{c} 9 \\ 1 \end{array} \right) = 10.$$

Consequently, the approximation has to hold for $n - \tau = 64 - 10 = 54$ of the possible values of the argument. This corresponds to an imbalance of at least $2 \cdot \left| \frac{54}{64} - \frac{1}{2} \right| = \frac{22}{32}$. □

8.4 An Interpretation Involving Algebraic-Geometry Codes

The parallels between S-box analysis and decoding of linear block codes appear even more striking when we consider the class of algebraic-geometry codes (AG-codes, see [2]). An AG-code is somewhat reminiscent to a Reed-Solomon code. AG-codewords are constructed by evaluating over the rational points of a plane curve the functions from the function field defined by the curve. Shokrollahi and Wasserman have generalized Sudan's algorithm to the case of AG-codes.

In the cryptographic setting, we may consider the curve to be defined by $y = f(x)$ where f is the S-box. Then the evaluation functions will be evaluated exactly at points describing the S-box, $\{(x_1, y_1), \ldots, (x_s, y_s)\}$.

8.5 Security against Probabilistic boolean Interpolation Attacks

Just as immunity against linear and differential attacks can be achieved by using bent functions, it is possible to obtain security against higher-order attacks that try to obtain nonlinear approximations. Recall that bent functions are multivariate boolean functions that are defined as having maximum distance from the set of affine functions. Bent functions have been described and constructed by several people [3, 18, 20].

In a similar fashion, one may ask how to construct or characterize functions that are far away from functions of degree higher than 1. We propose the name *ultrabent functions* [2] for such functions.

Definition 8.7 *A function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is called ultrabent of order r if it attains the maximum distance from the set of functions of algebraic degree at most r.*

Note that the usual set of bent functions coincide with the set of ultrabent functions of order 1.

In the following, let $\mathrm{dist}_r(f)$ denote the distance of the function f to the set of multivariate, boolean polynomials of degree at most r. The attainable maximum distance of an rth-order ultrabent function of m variables equals the covering radius of the (r, m) Reed-Muller code (for the same reason that the distance of a bent function to the affine functions equals the covering radius of the first-order Reed-Muller codes). I.e., we have the following result:

Theorem 8.8 *Let f be an ultrabent function of order r with m variables. Then its distance to the set functions of degree r is*

$$\mathrm{dist}_r(f) = \mathrm{CR}\left[\mathcal{RM}(r, m)\right]$$

where CR *denotes the covering radius of a code, and $\mathcal{RM}(r, m)$ denotes the Reed-Muller code of order r and length 2^m.*

Proof Follows directly from the definition of covering radius and the fact that the codewords in the Reed-Muller code of order r are the evaluations of all functions of degree at most r. □

We haven't been able to construct ultrabent functions or characterize them further. There is the following weak result, however, lower-bounding the maximum distance of a function in m variables (and hence also a hyperbent function) to the set of functions of degree r.

Theorem 8.9 *Let f be a binary function in m variables. Then*

$$\max_f \mathrm{dist}_r(f) \geq 2^{m-r-1}.$$

Proof Simply choose any f with $\deg f = r + 1$. The minimum distance of the $\mathcal{RM}(r + 1, m)$ code (which contains $\mathcal{RM}(r, m)$) is 2^{m-r-1}. Consequently, this serves as a lower-bound for the maximum distance. □

[2] The notions "hyperbent" and "generalized bent" were already taken [4, 12]; both meanings differ from that of ultrabent.

Chapter 9

Conclusion

Through cryptosystems
my plaintext flows silently.
The key is hidden.

- Cryptologist's haiku, 1999

On the basic level, cryptography deals with methodologies for authenticating information or keeping it confidential. The cryptographer's goal is to avoid the effects of adversarial behaviour from hostile counterparts. In an ideal world, such an information-theoretical lock smith would of course be without a job. But with information interconnectivity exploding globally in a wonderful but imperfect world, a secure infrastructure is needed more than ever.

Aside from the intellectual disappointment that no absolute proof of security has been found for a practical system, the threat of having to change a standard cipher world-wide overnight because it has been broken is enervating to say the least. Even reductions to easily characterized, hard problems as known from public-key cryptography seem difficult for practical block ciphers.

Nevertheless, it is interesting to see that more theoretically oriented, algebraic notions like factorization and the counting of points on certain curves[1] that have been known mostly from public-key cryptography are now slowly making their way into the theory of block ciphers.

As mentioned several times, however, the best strategy one can adopt presently to ensure a high level of security is to continue the iterated process of proposing new cipher constructions and attack them to see if they withstand.

Often, within the block-cipher community, this is carried out by looking at a specific cipher and attacking it using special tricks applicable maybe to that instance only. While this is very important from a practical, engineering point of view (nobody wants to use a bad cipher!), the theoretical knowledge

[1] The exponential sums over characteristic 2 considered in Chapter 8 are actually doing exactly this.

gained is often very limited. Sometimes a successful attacker learns only one bit of information (aside from eventual plaintext or key bits), namely that the considered cipher is insecure. No new general design principles can be extracted from such a process although sometimes an interesting design flaw or principle can be pointed out.

While the value of such narrow attacks should be acknowledged in many cases for raising the level of security, one may choose to consider broader (and maybe more theoretical) classes of attacks in order to gain more constructive knowledge. This has been the scope of this thesis. On the other hand, a theoretical result is of no engineering value if it does not apply to some real-world problem. Luckily, while managing to devise new theoretical classes of attacks here, the practical use has also been demonstrated on several actual proposed ciphers.

Some of the presented cryptological results have drawed heavily on recent advances in coding theory and we hope that this thesis contributes towards bringing the two fields closer together. They share the same mathematical foundation and both are models of communications — a fact which is often neglected although Shannon pioneered both areas using the same information-theoretical approach. In any case, coding has a much longer history of (public) research than cryptography and as such one might hope to learn something from the guy who has been around longer.

9.1 Main Results

The major achievements of this thesis are:

- **The interpolation attack**. A new class of attacks on block ciphers has been described. With both a deterministic and a probabilistic version, the attack exploits nonlinear relations of low degree between plaintext and ciphertext. It is particularly efficient against ciphers relying on simple algebraic functions in order to provide security against differential or linear cryptanalysis. The attack demonstrates that new design criteria are needed in order to obtain security.

- **A new Reed-Muller decoder**. An algorithm for decoding Reed-Muller codes beyond half the minimum distance has been developed. The algorithm has some similarities with the Sudan and the Welch-Berlekamp algorithms for Reed-Solomon codes. The decoder is useful for error-correction in settings with a very high noise rate (and correspondingly low information rate). In addition, as decoding Reed-Muller codes corresponds to approximation with multivariate polynomials of low degree, we also immediately have a procedure which could be very useful in the cryptanalysis of bit-oriented block-ciphers.

- **A new higher-order differential attack.** A novel generalization of differential cryptanalysis has been described. The attack quickly breaks

ciphers in an interpolation-like fashion when ciphertext bits are expressible as low-degree evaluations of plaintext bits.

- **Demonstration of weaknesses in several proposed ciphers**. Several proposed ciphers that aim to defeat the threat of differential or linear cryptanalysis have been shown by the above approaches to be either completely insecure or to exhibit theoretical weaknesses.

- **Links to coding theory**. A kind of dictionary between S-box design and coding theory has been provided. This allows several results from coding theory to be used almost directly in the area of block cipher design. In particular it has been demonstrated how the existence of decoding algorithms leads to procedures that can find simple relations over S-boxes or whole ciphers, and a few coding-theoretical bounds have been shown to apply to the cryptographical setting as well.

- **Factorization of boolean polynomials**. A simple algorithm for factorization of boolean, multivariate polynomials has been described. The algorithm allows by an easy extension the search for factors of a certain form (degree bound or similar linear restrictions). Also a number of results about approximations of binary functions appearing as factors in certain polynomials have been shown, especially in the case of Reed-Muller decoding.

9.2 Ideas for Further Research

The following list contains examples of interesting areas where one might build upon the results in this thesis or fill out gaps. The analysis of block ciphers should benefit greatly from additional advances in these areas.

- **Proving the Reed-Muller conjectures**. Although it works in practice, the presented Reed-Muller decoding algorithm still has two major conjectures related to it. Showing these to be true will immediately prove the efficiency and the error-correcting capability of the algorithm. The conjectures are closely related to the generally unknown weight distribution of Reed-Muller codes, however, and they both seem to be hard problems; fortunately, they are both supported by emperical results. Possibly, Gleason's theorem [2] about the weight distribution of selfdual codes could be of help in the case of the selfdual $\mathcal{RM}(\frac{m-1}{2}, m)$ code, m odd. To underline the practical usefulness of the algorithm more simulations would also be welcome.

- **Generalizing the Reed-Muller decoder**. It might be possible to generalize the decoder to generalized Reed-Muller codes, which operate on non-binary alphabets. Maybe also generalizations to other algebraic-geometry codes, BCH codes, or subfield codes are possible.

- **A wider class of attacks**. Since most block ciphers are based directly on the manipulation of bits at some level, interpolation-like attacks utilizing the Reed-Muller decoding algorithm will apply to a very general class of cryptosystems.

  But notably within the block-cipher setting where large finite fields are used, there is room for improvement. For instance, consider a notion of approximability that does not distinguish hard in the following sense. Assume that we want to approximate the function f by another simpler function g. Our current Hamming-weight-like measure of success is the probability that $f(x) = g(x)$, a value proportional to the metric

$$\mathrm{dist}(f, g) = \sum_x \delta(f(x), g(x)),$$

  where δ is Kronecker's delta. But for functions that are not binary-valued we may choose less "digital" metrics by defining a more gradual distance between elements, for instance

$$\mathrm{dist}(f, g) = \sum_x (\log_\alpha f(x) - \log_\alpha g(x))^2$$

  where the logarithm is taken with respect to some primitive element α of the underlying field. Or for finite fields with characteristic different from 2, it might be worth looking at the metric

$$\mathrm{dist}(f, g) = \left| \sum_x \chi_1(f(x)) - \chi_2(g(x)) \right|^2$$

  with χ_1 and χ_2 being appropriate, additive characters (this worked out very well in Chapter 4 and in [10] for the linear case). For characteristic 2 this simplifies more or less to the Reed-Muller problem already considered since in this situation characters have only two possible values.

  In the coding-theoretical interpretation, the introduction of such a gradual metric would somehow correspond to doing soft-decision decoding instead of hard-decision decoding.

- **More complexity-theoretical attack models**. In this thesis we have developed methods that can discover the existence of certain simple relations over a block cipher. Here, "simple" means "approximately expressible as a polynomial with a low number of coefficients". But one might use other characterizations of simple.

  Of course more general definitions give rise to more powerful attacks and at the extreme end of the spectrum we have the complexity theoretical definition. Take simple to mean "computationally simple" and if you can prove the nonexistence of an algorithm that will find such computationally simple relations over a block cipher, then you have effectively proved in

an absolute sense the security of the system. This is not likely to happen any day soon of course but one may pursue less general definitions that are still broader than the polynomial one and closer to a true complexity-theoretical definition of simplicity.

- **Other coding-theoretical connections**. It might be possible to take advantage of results in other coding-theoretical areas, e.g. cyclic codes, convolutional codes, codes with low-density parity check matrices, or algebraic geometry codes, in order to obtain tools for the cryptanalysis and design of block ciphers. The stream-cipher community has been doing this with great success for quite a while.

- **Cryptanalysis of AES candidates**. Although theoretical results are important, they are worthless if they cannot be used in practice. At the time of writing, a kind of competition is taking place to find a replacement for DES. The new algorithm, the Advanced Encryption Standard (AES), will be found among several candidates that have been submitted. These proposals are of course excellent targets and provide useful testing grounds for new attacks, e.g., a bit-oriented, probabilistic interpolation attack based on decoding Reed-Muller codes beyond half the minimum distance.

- **Security against higher-order attacks**. Just as bent functions are useful to provide security against linear attacks there exists functions that have a high distance to the set of polynomials of low degree. They would be useful in providing security against the higher-order attacks described in this thesis. The overall goal, of course, is to develop new design criteria that yield secure ciphers.

  In the multivariate case, this corresponds to finding words that are as far away as possible from the codewords in higher-order Reed-Muller codes. We have proposed the name *ultrabent* for such functions. While several constructions and classifications of bent functions are known, a treatment of ultrabent functions is still to come.

Bibliography

[1] Eli Biham and Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Berlin: Springer-Verlag, 1993.

[2] Richard Brualdi, William Cary Huffman and Vera S. Pless, editors, *Handbook of Coding Theory*, Elsevier, 1998.

[3] Claude Carlet and Philippe Guillot, A characterization of binary bent functions, *Journal of Combinatorial Theory*, Series A, vol. 76, no. 2, 328–335 (1996).

[4] Claude Carlet, Hyperbent functions, *Proceedings of Pragocrypt '96*, ed. J. Pribyl, Prague, 1996.

[5] L. Carlitz and S. Uchiyama, Bounds for exponential sums, *Duke Math. J.*, **24** (1957), pp. 37–41.

[6] Carlo Harpes, *Cryptanalysis of iterated block ciphers*, Ph.D. thesis, ETH Series in Information Processing, Ed. J. L. Massey, Hartung-Gorre Verlag Konstanz, 1996.

[7] Carlo Harpes, Gerhard Kramer, and James Massey, A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma, *Advances in Cryptology — EUROCRYPT '95*, pp. 24–38.

[8] Thomas Jakobsen, *A Fast Method for Cryptanalysis of Substitution Ciphers*, Cryptologia 19(3), July 1995.

[9] Thomas Jakobsen, *Security Against Generalized Linear Cryptanalysis and Partitioning Cryptanalysis*, Semester Project at Signal and Information Processing Laboratory, Swiss Federal Institute of Technology Zurich, Zürich 1995.

[10] Thomas Jakobsen, *Correlation Attacks on Block Ciphers*, Master's Thesis, Department of Mathematics, Technical University of Denmark, February 1996.

[11] L. R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 196–211. Springer Verlag, 1995.

[12] P. V. Kumar, R. A. Scholtz, and L. Welch, Generalized Bent Functions and their Properties, *Journal of Combinatorial Theory*, September 1985.

[13] X. Lai. Higher order derivatives and differential cryptanalysis. In R. Blahut, editor, *Communication and Cryptography, Two Sides of One Tapestry.* Kluwer Academic Publishers, 1994.

[14] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", Proceedings of Crypto '94, Lecture Notes in Computer Science.

[15] W. Meier and O. Staffelbach, Fast Correlation Attacks on Stream Ciphers, Advances in Cryptology — EUROCRYPT '88, pp. 301–314, Springer-Verlag.

[16] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press Series on Discrete Mathematics and Its Applications, 1996.

[17] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *The Journal of Cryptology*, 8(1):27–38, 1995.

[18] J. Olsen, R. A. Scholtz, and L. Welch, Bent Function Sequences, IEEE Trans. on Info. Th., vol. 28, no. 6, November 1982.

[19] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher SHARK. In Gollmann D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 99–112. Springer Verlag, 1996.

[20] O. S. Rothaus, On Bent Functions, *Journal of Combinatorial Theory*, vol. A20, pp. 300-305, 1976.

[21] Takeshi Shimoyama, Private communication.

[22] Takeshi Shimoyama, Toshinobu Kaneko, Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES. In Hugo Krawczyk (Ed.): *Advances in Cryptology - CRYPTO '98*, Proceedings. Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 200–211.

[23] Douglas Stinson, *Cryptography: Theory and Practice*, CRC Press, Inc., Boca Raton, 1995.

[24] Madhu Sudan, Decoding of Reed-Solomon codes beyond the error-correction bound, *Journal of Complexity*, 13(1):180–193, March 1997.

[25] Alexander Vardy, The intractability of computing the minimum distance of a code, *IEEE Trans. on Info. Th.*, vol. 43, no. 6, Nov. 1997.

[26] David Wagner. Personal communication.