KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Automatic SAT-solver Based Search Tools for Cryptanalysis

Glenn De Witte

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: wiskundige ingenieurstechnieken

Promotor:
Prof. dr. ir. Vincent Rijmen

Assessoren:
Prof. dr. ir. J. Suykens
Prof. dr. ir. M. Van Barel

Begeleiders:
T. Ashur
Y. Liu

Academiejaar 2016 – 2017

© Copyright KU Leuven

Without written permission of the thesis supervisor and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to the Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 or by email `info@cs.kuleuven.be`.

A written permission of the thesis supervisor is also required to use the methods, products, schematics and programs described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail `info@cs.kuleuven.be`.

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Preface

In this preface I would like to take the opportunity to thank the individuals who helped me finish this master's thesis, without their support this would not be possible.

I would like to thank my promotor, Prof. V. Rijmen, to provide me with such an interesting thesis topic and giving me the opportunity to explore my interests in cryptography.

Enormous gratitude to my two supervisors, Tomer Ashur and Yunwen Liu, who were there for me on every moment of this journey. The door to their office was always open, ready to help me in every aspect of this thesis, from improving my presentation skills or proof reading my thesis text to discussing various topics in cryptography.

I would also like to thank my two assessors, Prof. J. Suykens and Prof. M. Van Barel to read this thesis, I hope they enjoy the fruits of my labor.

And lastly thank you to my family to be there for me, in good and bad times, especially my sister Emma, and my mum and dad, Nandy and Kris.

Glenn De Witte

Contents

Preface i

Abstract iii

List of Figures and Tables iv

List of Abbreviations and Symbols vi

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Goals and Modus Operandi . . . 1
- 1.3 Outline . . . 2

2 Statistical Cryptanalysis for ARX primitives 3
- 2.1 Block ciphers . . . 4
- 2.2 Cryptanalytic Techniques . . . 7
- 2.3 Conclusion . . . 14

3 Automated Search of RX-characteristics 15
- 3.1 Boolean Satisfiability problem . . . 15
- 3.2 From RX-characteristic to SAT problem . . . 16
- 3.3 Application to SPECK . . . 17
- 3.4 Conclusion . . . 21

4 Results of the automated search for SPECK 23
- 4.1 Used software and hardware . . . 23
- 4.2 Probability of optimal RX-characteristics for SPECK . . . 23
- 4.3 Optimal RX-characteristics for SPECK . . . 27
- 4.4 Experimental verification . . . 30
- 4.5 Comparison to other cryptanalytic techniques . . . 30
- 4.6 Conclusion . . . 34

5 Conclusion 35

A Paper submitted to the "2017 Symposium on Information Theory and Signal Processing in the Benelux" 39

B Poster presented on the "masterproefbeurs" 49

Bibliography 51

Abstract

Cryptographic primitives are essential building blocks in modern cryptographic systems, used for providing confidentiality, authenticity and data integrity. In symmetric key cryptography, where encryption and decryption happens with the same fixed key K, these primitives are block ciphers and stream ciphers. One of the most popular designs for these primitives is ARX, which stands for Addition-Rotation-XOR. An advantage of these designs is efficiency, especially important in the advent of the *Internet of Things*.

However, before such designs are implemented in practice, it is of critical importance to test the security of these primitives. The field in cryptography aimed at investigating the security of cryptographic systems is cryptanalysis. Security proofs often do not exist, and the only way to know if a certain primitive can be considered secure is to check if it is resistant against known cryptanalytic techniques. Cryptanalytic techniques that exploit probabilistic properties in the primitive belong to the field of statistical cryptanalysis.

Rotational cryptanalysis is a statistical attack effective against ARX designs with (almost) rotational invariant constants. The technique is based on the fact that most rotational pairs propagate with known probability through ARX designs. Through modular additions, rotational pairs propagate with a certain probability, through constants they propagate for sure if the constant is rotational invariant, or not at all if the constant is rotational variant.

To deal with rotational variant constants, which destroy rotational pairs, one can define an RX-difference, which not only keeps track of a rotational offset, but also the translation needed to make a pair a rotational pair. A path of an RX-difference through different rounds of encryption is called an RX-characteristic. For the technique to be successful, the RX-characteristic has to be highly probable. Mathematical methods to evaluate this probability for a certain RX-difference through a specific primitive exist, but because of the large number of possible RX-characteristics it is natural to automate the search for finding good ones.

We investigate how an automatic search tool can be developed for finding good RX-characteristics in ARX primitives. We implement this search tool for the Speck family of block ciphers. This is a family of ARX block ciphers that was developed by the NSA to fulfil the need for lightweight and flexible block ciphers. The results of this automated search are given and compared with results of other statistical techniques on round-reduced Speck.

List of Figures and Tables

List of Figures

2.1 Schematic representation of the encryption and decryption algorithm of a block cipher. . . . 4
2.2 Iterated block cipher design. . . . 6
2.3 Two ARX primitives with different rotational probabilities. . . . 12

3.1 Description of Speck 18
3.2 Notation of RX-differences in Speck. . . . 19

List of Tables

3.1 Truth tables of the logical AND and logical OR. . . . 16
3.2 Truth table of the logical NOT. . . . 16
3.3 Parameters for different versions of Speck. . . . 18

4.1 Probability and size of weak key-class of optimal RX-characteristic through Speck32/64. . . . 24
4.2 Probability and size of weak key-class of optimal RX-characteristics throughout Speck48/96. . . . 25
4.3 Probability and size of weak key-class of optimal RX-characteristics throughout Speck64/128. . . . 25
4.4 Probability and size of weak key-class of optimal RX-characteristics throughout Speck96/144. . . . 26
4.5 Probability and size of weak key-class of optimal RX-characteristics throughout Speck128/256. . . . 26
4.6 Optimal RX-characteristic for 12-round Speck32/64. . . . 27
4.7 Optimal RX-characteristic for 14-round Speck48/96. . . . 28
4.8 Optimal RX-characteristic for 12-round Speck64/128. . . . 28
4.9 Optimal RX-characteristic for 13-round Speck96/144. . . . 29
4.10 Optimal RX-characteristic for 13-round Speck128/256. . . . 29
4.11 Comparison of theoretical probability and empirical probability of 9-round RX-characteristic in Speck32/64. . . . 30

4.12 Comparison of optimal RX-characteristics to optimal linear trails throughout Speck32/64. 31
4.13 Comparison of optimal RX-characteristics to optimal linear trails throughout Speck48/96. 31
4.14 Comparison of optimal RX-characteristics to optimal linear trails throughout Speck64/128. 32
4.15 Comparison of optimal RX-characteristics to optimal linear trails throughout Speck96/144. 32
4.16 Comparison of optimal RX-characteristics to optimal linear trails throughout Speck128/256. 33
4.17 Comparison of the maximum number of rounds analysed with linear cryptanalysis and RX-cryptanalysis. 33

List of Abbreviations and Symbols

Abbreviations

IoT	*Internet of Things*
NSA	National Security Agency
XOR	Exclusive Or
ARX	Addition-Rotation-XOR
SAT problem	Boolean Satisfiability Problem
RX-difference	Rotational-XOR-difference
MILP	Mixed-Integer Linear Programming

Symbols

$x = (x^{n-1}, \ldots, x^1, x^0)$	An n-bit boolean vector.
$\|x\|$	The Hamming weight of the Boolean vector x.
$\#\{x\}$	The number of bits in the Boolean vector x.
$x \ggg \gamma$	A cyclic right shift of x by γ bit(s).
$SHL(x)$	An arithmetic left shift of x by one bit.
$(I \oplus SHL)(x)$	$x \oplus SHL(x)$
$L(x)$	The $n-1$ most significant bits of x: $(x^{n-1}, \ldots, x^2, x^1)$.
$x\|\|y$	The concatenation of the vectors x and y.
$x\|y$	The bitwise OR operation between the vectors x and y.
$1_{x \preceq y}$	Characteristic function which evaluates to 1 if $\forall i : x^i \leq y^i, 0 \leq i < n$, otherwise to 0.
Speck$2n/mn$	Version of Speck with word size n and number of key words of m.
T	logical TRUE
F	logical FALSE
$\wedge$	logical AND
$\vee$	logical OR
$\neg$	logical NOT

Chapter 1

Introduction

1.1 Context

In symmetric key cryptography, primitives, *i.e.* block ciphers and stream ciphers, are essential building blocks for providing confidentiality, authenticity and data integrity. In the advent of the *Internet of Things*, the efficiency of such primitives is critical, along with their security. Some of the most popular designs for these primitives are ARX designs, which are composed of three operations: modular addition, rotation and exclusive or. This popularity stems from the belief that these designs can be efficient and secure at the same time. Because security proofs often do not exist for ARX primitives, they should at least be resistant to all known cryptographic techniques to be considered secure. Statistical cryptanalysis is the field of cryptography that aims to investigate the security of such primitives by trying to find probabilistic relations between known pieces of information and the secret key. Several statistic techniques against ARX primitives exist and in this thesis we investigate such a technique, namely Rotational-XOR cryptanalysis. More precisely, we examine how to maximize the probability that Rotational-XOR cryptanalysis succeeds against a certain ARX design. Maximizing this probability has been successfully done for other statistical techniques, such as differential or linear cryptanalysis, by making use of a crypto oriented SAT-solver. In this thesis, we also take this approach and test it against Speck, a block cipher family designed by the NSA to fulfil the need for an efficient and yet secure block cipher implementable on resource-constrained devices.

1.2 Goals and Modus Operandi

The goal of this thesis is twofold. One part is describing how the problem of maximizing the probability of success for rotational-XOR cryptanalysis can be solved. The other part is developing a program that implements this solution for a specific ARX primitive, namely the Speck block cipher family.

1.3 Outline

In chapter 2, an overview of some statistical cryptanalytic techniques for ARX primitives is given. This overview contains Rotational-XOR cryptanalysis, the technique we are concerned with in this thesis.

In chapter 3, the definition of a SAT problem is given and we explain how statistical relations, in the case of rotational-XOR cryptanalysis called RX-characteristics, in ARX primitives can be found by solving a series of SAT problems. This series of SAT problems is then derived explicitly for Speck.

In chapter 4, the results of the automatic search are given and verified. A comparison is also made with previous results of other statistical attacks for Speck.

Finally, in chapter 5, we conclude this thesis by recapitulating the most important results and conclusions.

Chapter 2

Statistical Cryptanalysis for ARX primitives

Ever since the beginning of written languages, people have sent secret messages to each other. Such messages are private messages from a sender to a recipient which are coded in a way that only the sender and receiver should be able to read them. Since the advent of computer technology and its communication networks, people also desire to secure messages over these communication networks.

A communication protocol is a set of rules that allows two (or more) users to transmit information over a communication network. A cryptographic protocol is the same, but it allows (or at least tries) to protect the information between sender and receiver(s) in the presence of an adversary. If the sender and receiver use the same secret key K to protect the information, the protocol uses a symmetric-key algorithm. Primitives, *i.e.* block ciphers and stream ciphers, are important building blocks for symmetric-key algorithms. Block ciphers define an encryption and decryption algorithm which work on n-bit vectors, also called blocks. The block cipher specifies how blocks are to be encrypted and decrypted using these encryption and decryption algorithms which both have a master key K as parameter.

The security of a certain block cipher is often hard to estimate, let alone to prove. A block cipher's security is often evaluated on how many known cryptanalytic attacks the block cipher has resisted together with how analysable the block cipher is. The most known cryptanalytic attacks for block ciphers are differential and linear cryptanalysis.

The design of many modern block ciphers is an iterated design. A block cipher then consists of multiple equivalent round functions, with the difference that a different round key is injected in every round. In the light of pervasive computing and lightweight cryptography, the round function of some modern block ciphers only consists of three operations: (modular) Addition, Rotation and XOR. This is called an ARX design.

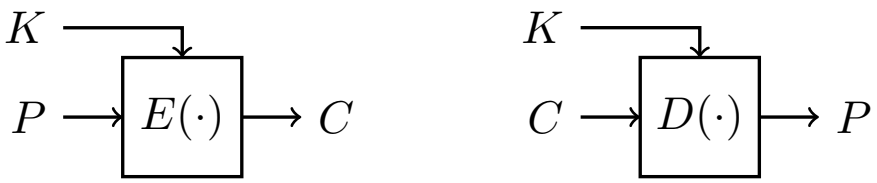
A form of cryptanalysis for ARX designs which successfully revealed some weaknesses is rotational cryptanalysis, first introduced in [9]. One important limitation of using this technique is that it deals badly with constants. In [2] the technique of

rotational cryptanalysis is generalized to take constants into account.

In this chapter, block ciphers are discussed in greater depth along with some cryptanalytic statistical techniques. The techniques are discussed for ARX block ciphers in particular, but are applicable for ARX primitives and primitives in general.

2.1 Block ciphers

Block ciphers specify an encryption algorithm $E_K(\cdot)$ (Figure 2.1a) and a decryption algorithm $D_K(\cdot) = E_K^{-1}(\cdot)$ (Figure 2.1b), both with a fixed key K as a parameter. Both the input and output of these algorithms are strings of n bits, called blocks. The plaintext P is what we call the input to the encryption algorithm and the ciphertext $C = E_K(P)$ its output of the encryption algorithm. Equivalently, C is the input to the decryption algorithm and $P = D_K(C)$ its output. Because the decryption algorithm is the inverse of the encryption algorithm, both the encryption and decryption are permutations on n-bit blocks (or bijective mappings from the set of n-bit blocks to the set of n-bit blocks).



(a) Encryption algorithm. (b) Decryption algorithm.

Figure 2.1: Schematic representation of the encryption and decryption algorithm of a block cipher.

Ideally, we want for every key K that the encryption function $E_K(\cdot)$ is a permutation $\pi_K : \{0,1\}^n \to \{0,1\}^n$ randomly chosen out of the set of all possible permutations $P = \{\pi : \{0,1\}^n \to \{0,1\}^n \mid \pi \text{ is a permutation}\}$. In other words, the plaintext P, the ciphertext C and the key K are not correlated in any way in an ideal block cipher. This means, among other things, that no information about a secret key K can be found, even if we collect a lot of plaintext-ciphertext pairs for that particular key K. It also means that collecting plaintext-ciphertext pairs for a particular key K will give no information about plaintext-ciphertext pairs for another key $K' \neq K$. Although such a block cipher does not exist in practice, this is what we call an ideal block cipher. The idea is that a block cipher in practice should be indistinguishable from the ideal block cipher.

2.1.1 Properties

When trying to find out how suitable a certain block cipher is to use in a specific application, the following properties are often considered as important evaluation criteria:

Block Length and Key Length

Every block cipher has a specified block length and key length, meaning the number of bits of a block and master key respectively. The key length provides an upper bound on the security of the block cipher. This is because an attacker can exhaustively search through all keys until finding the right key. This exhaustive search can be avoided if the key is sufficiently large, how large exactly depends on the desired security level.

Estimated Security

As said in the previous section, the key length provides an upper bound on the level of security in a block cipher. But what about a lower bound? The (maybe disappointing) answer is that in general there is not one. The majority of block ciphers are not proven to be secure. One thing a block cipher should inevitably provide is to be secure against all known cryptanalytic techniques, but aside from that, estimating the security is often educated guesswork. A block cipher that is considered secure today might be broken tomorrow and render the block cipher completely useless.

However, this does not happen too much in practice, and there are some general guidelines so one can consider one cipher to be probably more secure than another cipher. In general, if the cryptographic community spends time and effort in trying to break a cipher and does not succeed, the block cipher is considered more secure than one that is studied little or not at all. This is also the reason why block ciphers that are used in practice are published, that way they can be studied by more people. A block cipher that has a mathematically sound basis and/or of which the encryption/decryption-algorithm is simple, can be analysed more easily by the cryptographic community, which is also an advantage. More complicated algorithms are more likely to have hidden weaknesses, some can go unnoticed.

Throughput

Part of the performance of a block cipher is quantified by the throughput of the block cipher. The throughput of a block cipher is expressed as the maximum number of bits that can be encrypted/decrypted in a certain time interval. This of course depends on how fast a block cipher can encrypt/decrypt one block, which in turn depends on how long the operations in the round function itself take and if some of them can be executed in parallel.

Implementation Cost

The implementation cost of a block cipher expresses how much software and/or hardware an implementation of the block cipher needs. The number of lines of code of an implementation of a block cipher in a specific software language is an example of a cost in software. In the case of a hardware implementation, an example is the number of hardware gates. In many resource-constrained environments, some or all

of these costs cannot be too high. A good and well-thought design is very important, but also the block length and key length are important factors to consider, because this leads often, if not always, to a higher implementation cost. The block length and key length of a certain block cipher are always a trade-off between a desired security level and the intended environment in which the block cipher should be implemented.

Other Evaluation Criteria

There are some other properties of block ciphers that are often considered as evaluation criteria (for example licensing requirements), but the ones described above are considered the most important in an academic context.

2.1.2 Design of block ciphers

The design of modern block ciphers is an iterated design, first proposed in [13]. Encryption is achieved by concatenating multiple, equivalent round functions. The only difference in each round is the use of a different round key, derived from the master key K and often injection via the XOR operation. This is shown in Figure 2.2, the subsequent round functions are denoted by R_i and the round key of round i is denoted by k_{i-1}. Other operations in the round function often include operations to reorder the bits (rotations, permutation-boxes, ...) and a nonlinear operation with respect to the XOR operation (modular addition, S-boxes, ...). The larger the number of rounds, the higher the estimated security. In designs like this, decryption is equivalent to encryption, except that the order of the round keys is reversed.

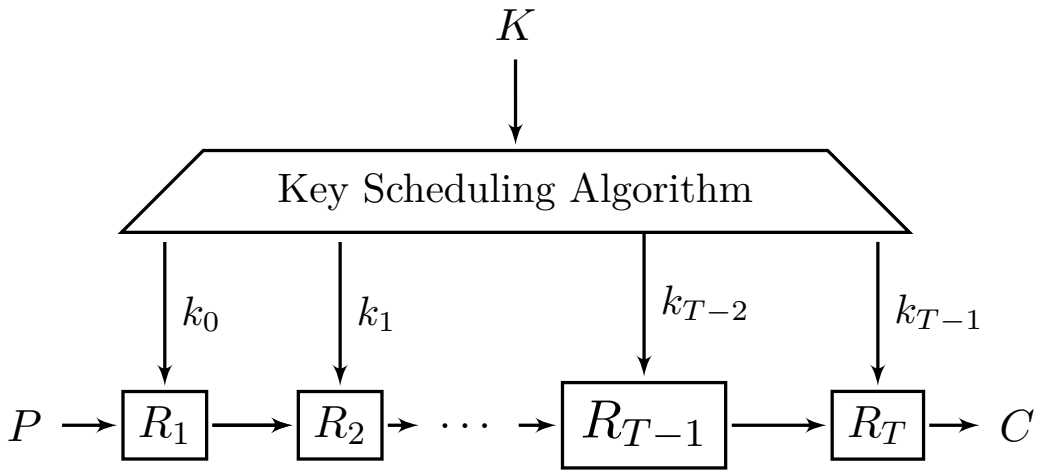


Figure 2.2: Iterated block cipher design.

2.2 Cryptanalytic Techniques

In this section we will shortly discuss some important cryptanalytic techniques relevant for this thesis. It is important to know where the techniques are situated in the context of cryptanalysis, so we give an overview of a classification of cryptanalytic techniques.

2.2.1 Classification

Cryptanalysis can be classified in two types. One type tries to exploit weaknesses in the implementation of a specific cryptographic primitive, which are called side-channel attacks. Examples are timing attacks [1], *i.e.* attacks that exploit how much time it takes for each operation in the implementation to perform, or power-monitoring attacks [8], attacks that measure the varying power consumption during computation. In this case success or failure of the attack is dependent on the implementation of the ARX primitive and not the ARX primitive itself. Side-channel attacks do not concern us in this thesis.

The other is a mathematical analysis of a cryptographic primitive, which exploit a mathematical weakness of the cryptographic primitive. In this case the probability of success is not dependent on the exact implementation of the primitive. Mathematical cryptanalysis includes a number of statistical techniques. It investigates undesirable probabilistic properties in cryptographic primitives.

For a certain cryptanalytic technique, there is often an assumption on which information is available or can be obtained by the attacker. These assumptions are listed below, sorted by an increasing amount of different types of information available to the attacker.

Ciphertext-only A set of ciphertexts is available to the attacker.

Known-plaintext A set of plaintexts with corresponding ciphertexts is available to the attacker.

Chosen-plaintext Similar to known-plaintext but the attacker can choose the set of plaintexts.

Adaptive chosen-plaintext Similar to known-plaintext but the attacker can choose plaintexts adaptively based on previously seen ciphertexts.

Chosen-plaintext/chosen-chiphertext The attacker can choose a set of plaintexts as well as a set of ciphertexts and get the corresponding ciphertexts and plaintexts.

adaptive chosen-plaintext/chosen-ciphertext Identical to chosen-plaintext/chosen-ciphertext but the attacker can choose plaintexts and ciphertexts adaptively.

For the above techniques, a related-key setting or non-related key setting can be assumed. In a related-key setting, the attacker can not only get access to information

encrypted or decrypted under a certain key K, but also under a key K' which is related to key K in a way defined by the attacker. In a non-related key setting, only information encrypted or decrypted under one key can be accessed.

The description of some popular statistical cryptanalytic techniques is given in the following sections.

2.2.2 Linear cryptanalysis

Linear cryptanalysis is a known-plaintext attack introduced by Mitsuru Matsui [11] in 1993. It was introduced as an attack on the Data Encryption Standard (DES). Linear cryptanalysis analyses the probability of linear expressions involving plaintext bits, ciphertext bits and (round) key bits, where the linearity is with respect to the XOR operation. This expressions take the following form:

$$p^i \oplus p^j \oplus ... \oplus p^k \oplus c^l \oplus c^m \oplus ... \oplus c^n \oplus k^o \oplus k^p \oplus k^q = 0,$$

where p is the plaintext, c is the ciphertext, k is one the round keys and the letters denote specific bits. In the case of randomly chosen plaintext, ciphertext and round keys, the expected probability of the above equation to be valid is $1/2$, no matter which specific bits are selected. Linear cryptanalysis exploits expressions in specific block ciphers where this is not the case, and the probability differs from $1/2$. The larger the absolute difference between the probability of the expression and $1/2$, the better the linear expression for our cryptanalysis and the higher the weakness in the block cipher. We call the absolute difference between the probability of the expression and $1/2$ the bias. Now notice that if we remove the key bits from the expression:

$$p^i \oplus p^j \oplus ... \oplus p^k \oplus c^l \oplus c^m \oplus ... \oplus c^n = 0,$$

the probability of the expression might change but the bias will stay exactly the same.

To find expressions of this form, each operation in the round function needs to be analysed independently and a linear approximations need to be derived for it. By concatenating linear approximations of each component, an expression for the whole block cipher can be derived. If the bias of this expression is large enough, it can be used to mount a key recovery attack on the block cipher.

2.2.3 Differential cryptanalysis

Differential cryptanalysis is a chosen-plaintext attack against cryptographic primitives. The technique was published in 1990 by Eli Biham and Adi Shamir [4], and a lot of other statistical attacks have been derived from this technique. It was probably known to some members of IBM and NSA before its publication. This is because the S-boxes in the Data Encryption Standard, a block cipher published in 1975, developed by IBM and NSA are surprisingly resistant against differential cryptanalysis [5]. Randomly chosen S-boxes or small changes to the chosen S-boxes for DES would make the block cipher much more susceptible to differential cryptanalysis.

Differential cryptanalysis analyses the effect of specific differences in plaintext pairs on the differences in the encrypted ciphertext pairs. More specifically, it exploits that a certain difference $\Delta p = p_1 \oplus p_2$ (where p_1 and p_2 are plaintexts) propagates with a higher probability than a random permutation would to a certain ciphertext difference $\Delta c = c_1 \oplus c_2$ (where c_1 and c_2 are p_1 and p_2 encrypted under a certain key).

To apply differential cryptanalysis to a primitive, one has to study the propagation of the differences throughout each operation used in the round function. The operations used in the round function are often restricted to the exclusive or operation (XOR), bit permutations, arithmetic operations and S-boxes. In most primitives the round keys in each round are injected via the XOR operation and for the rest of this thesis we will assume this is the case. It is important to have an operation that is nonlinear with respect to the operation that injects the round keys, because otherwise the primitive can easily be broken. The S-boxes and some arithmetic operations are nonlinear, however often only one nonlinear operation is present in the round function of a primitive.

The probability of the propagation of the differences is often calculated by multiplying the probabilities of the propagation throughout the different operations in the primitive assuming that the Markov assumption holds, *i.e.* that each round key is an independent uniform random variable, which is in practice not the case because each round key is derived from the same master key. However, the empirical probabilities observed are often very close to the calculated probabilities. Let's assume $\Delta(x, x')$ is the difference between two bit vectors x and x', then the following rules of propagation apply for this difference if the Markov assumption holds:

- For a bit rotation with rotation amount α, the difference after rotation $\Delta(z, z')$ is equal to the difference $\Delta(x, x')$ rotated over the same amount α.

  Proof.

  Let
  $$\begin{aligned}\Delta(x, x') &= x \oplus x',\\ z &= x \ggg \alpha,\\ z' &= x' \ggg \alpha = (x \oplus \Delta(x, x')) \ggg \alpha = (x \ggg \alpha) \oplus (\Delta(x, x') \ggg \alpha),\end{aligned}$$
  then
  $$\Delta(z, z') = z \oplus z' = \Delta(x, x') \ggg \alpha.$$

  □

- For every XOR operation with another difference $\Delta(y, y')$, the output difference $\Delta(z, z')$ is equal to both differences $\Delta(x, x')$ and $\Delta(y, y')$ XORed together.

  Proof.

Let

$$\begin{aligned}\Delta(x, x') &= x \oplus x',\\ z &= x \oplus y,\\ z' &= (x \oplus \Delta(x, x')) \oplus (y \oplus \Delta(y, y')) = (x \oplus y) \oplus (\Delta(x, x') \oplus \Delta(y, y')),\end{aligned}$$

then

$$\Delta(z, z') = z \oplus z' = \Delta(x, x') \oplus \Delta(y, y').$$

□

- For a XOR operation with a constant c, the output difference $\Delta(z, z')$ is equal to $\Delta(x, x')$. This rule is a specific case of the previous rule, where $c = y = y'$ and thus $\Delta(y, y') = 0$.

  Proof.

  Let

  $$\begin{aligned}\Delta(x, x') &= x \oplus x',\\ z &= x \oplus c,\\ z' &= (x \oplus \Delta(x, x')) \oplus c = (x \oplus c) \oplus \Delta(x, x'),\end{aligned}$$

  then

  $$\Delta(z, z') = z \oplus z' = \Delta(x, x').$$

  □

The above rules are deterministic, no probabilities are involved. They are concerned with the linear operations in ARX designs, namely the rotation and exclusive or. In general, if an operation is linear (with respect to the exclusive or operation), the rules of propagation for differences are deterministic. If an operation is nonlinear, the rules of propagation are probabilistic.

2.2.4 Rotational cryptanalysis

A cryptanalytic technique especially effective against ARX primitives that use (almost) rotation-invariant constants is rotational cryptanalysis. The attack analyses the probability of the propagation of a rotational pair, two bit vectors which are rotated versions of each other. The propagation of a rotational pair $(x, x \ggg \gamma)$ throughout the operations of ARX-primitives is as follows:

- For every ARX operation, the rotational offset γ remains constant.
- For a rotation with rotation amount α, the output pair (z, z') is again a rotational pair.

Proof.

The pair (z, z'), which is the result of the pair $(x, x \ggg \gamma)$ rotated over rotation amount α, satisfies:

$$z = x \ggg \alpha,$$
$$z' = (x \ggg \gamma) \ggg \alpha = (x \ggg \alpha) \ggg \gamma = z \ggg \gamma.$$

□

- For an XOR operation with another rotational pair $(y, y \ggg \gamma)$, the output pair (z, z') is again a rotational pair.

  Proof.

  The pair (z, z'), which is the result of the XOR of the pairs $(x, x \ggg \gamma)$ and $(y, y \ggg \gamma)$, satisfies:

  $$z = x \oplus y,$$
  $$z' = (x \ggg \gamma) \oplus (y \ggg \gamma) = (x \oplus y) \ggg \gamma = z \ggg \gamma.$$

  □

- For an XOR operation with a constant c, the output pair (z, z') is again a rotational pair if and only if the constant is rotational invariant with respect to γ, *i.e.* $c = c \ggg \gamma$.

  Proof.

  The pair (z, z'), which is the result of the XOR the pair $(x, x \ggg \gamma)$ and the constant c, satisfies:

  $$z = x \oplus c.$$

  If $c = c \ggg \gamma$, then:

  $$z' = (x \ggg \gamma) \oplus c = (x \ggg \gamma) \oplus (c \ggg \gamma) = (x \oplus c) \ggg \gamma = z \ggg \gamma.$$

  If $c \neq c \ggg \gamma$, then:

  $$z' = (x \ggg \gamma) \oplus c \neq (x \ggg \gamma) \oplus (c \ggg \gamma) = (x \oplus c) \ggg \gamma = z \ggg \gamma.$$

  □

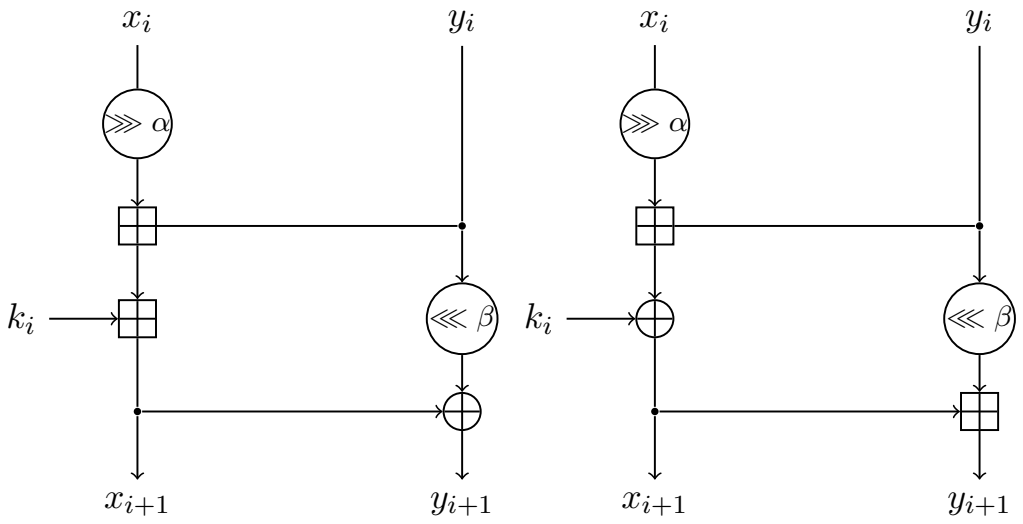
The propagation of a rotational pair through all of the above operations is completely deterministic. For modular addition however, the propagation of the rotational pair is probabilistic and the following theorem holds:

Theorem 1 (Propagation of rotational pairs through modular addition [9])**.** *Suppose* $x, y \in \mathbb{F}_{2^n}$ *are independent uniform random variables. Then*

$$Pr[(x \ggg \gamma) \boxplus (y \ggg \gamma) = (x \boxplus y) \ggg \gamma] = \frac{1}{4}(1 + 2^{\gamma-n} + 2^{-\gamma} + 2^{-n}). \tag{2.1}$$

Proof. The complete proof can be found in [9, Corollary 4.12]. □

The above probability ranges from 1/4 to 3/8, it is minimized when $\gamma = n/2$ and maximized when $\gamma = 1$ for a fixed value of n. At first it was believed that the probability of propagation of rotational pairs through ARX primitives can be calculated by only counting the number of additions. However, when chained modular additions (the output of a previous addition is an input of the next addition) are present, this is incorrect because the assumption that the two inputs are independent uniform variables in theorem 1 does not hold. The reason is that variables part of a rotational pair as output of a previous modular addition cannot be modelled as independent uniformly distributed. To further clarify this issue, Figure 2.3 is given. In 2.3a, an ARX primitive with a modular chain is given, and the probability that a rotational pair propagates throughout the primitive cannot be calculated by multiplying the probabilities through each modular addition operation independently. In Figure 2.3b, no modular chain is present and multiplying the probabilities as computed in Theorem 1 gives the correct probability.



(a) ARX primitive with a modular chain. (b) ARX primitive with no modular chain.

Figure 2.3: Two ARX primitives with different rotational probabilities.

2.2.5 RX-cryptanalysis

One disadvantage of rotational cryptanalysis is its inability to deal well with rotation-variant constants. To remedy this, in [9] the idea is mentioned to keep track of a

rotation error, which defines how close two bitvectors are to being a rotational pair. In [2], this is formalized by introducing RX-differences and formulas to compute the propagation of RX-differences through ARX operations. We revisit the theory here, with a slightly different notation more suited for this thesis.

Definition 1. *A Rotational-XOR-difference (in short, RX-difference)* $\Delta_\gamma(x, x')$ *is defined as the difference* $\Delta_\gamma(x, x') = x \oplus (x' \lll \gamma)$ *between two bit vectors* x *and* x'*.*

From the definition follows that if a vector x and a particular RX-difference $\Delta_\gamma(x, x')$ is given, the vector x' can easily be constructed: $x' = (x \oplus \Delta_\gamma(x, x')) \ggg \gamma$. As with (XOR-)differences and rotational pairs, the propagation of RX-differences is completely deterministic, except through modular addition. The following rules apply:

- For every ARX operation, the rotational offset γ remains constant.
- For a bit rotation with rotation amount α, the RX-difference after rotation $\Delta_\gamma(z, z')$ is equal to the RX-difference $\Delta_\gamma(x, x')$ rotated over the same amount α.

  Proof.

  Let
  $$\Delta_\gamma(x, x') = x \oplus (x' \lll \gamma),$$
  $$z = x \ggg \alpha,$$
  $$z' = (x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \ggg \alpha = ((x \ggg \alpha) \oplus (\Delta_\gamma(x, x') \ggg \alpha)) \ggg \gamma,$$
  then
  $$\Delta_\gamma(z, z') = z \oplus (z' \lll \gamma) = \Delta_\gamma(x, x') \ggg \alpha.$$

  □

- For every XOR operation with another RX-difference $\Delta_\gamma(y, y')$, the output RX-difference $\Delta_\gamma(z, z')$ is equal to both RX-differences $\Delta_\gamma(x, x')$ and $\Delta_\gamma(y, y')$ XORed together.

  Proof.

  Let
  $$\begin{aligned} \Delta_\gamma(x, x') &= x \oplus (x' \lll \gamma), \\ z &= x \oplus y, \\ z' &= ((x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \oplus ((y \oplus \Delta_\gamma(y, y')) \ggg \gamma) \\ &= ((x \oplus y) \oplus (\Delta_\gamma(x, x') \oplus \Delta_\gamma(y, y')) \ggg \gamma), \end{aligned}$$
  then
  $$\Delta_\gamma(z, z') = z \oplus (z' \lll \gamma) = \Delta_\gamma(x, x') \oplus \Delta_\gamma(y, y').$$

  □

- For a XOR operation with a constant, the RX-difference is XORed with the constant and the constant rotated over an amount γ:

  Proof.

  Let

  $$\begin{aligned}\Delta_\gamma(x, x') &= x \oplus (x' \lll \gamma),\\ z &= x \oplus c,\\ z' &= ((x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \oplus c = (x \oplus \Delta_\gamma(x, x') \oplus (c \lll \gamma)) \ggg \gamma\\ &= ((x \oplus c) \oplus \Delta_\gamma(x, x') \oplus (c \lll \gamma) \oplus c) \ggg \gamma,\end{aligned}$$

  then

  $$\Delta_\gamma(z, z') = z \oplus (z' \lll \gamma) = \Delta_\gamma(x, x') \oplus (c \lll \gamma) \oplus c.$$

  □

The propagation probability through modular addition is defined below, assuming independent uniform inputs:

Theorem 2 (Propagation of RX-differences through modular Addition [2])**.** *Suppose $x, y \in \mathbb{F}_{2^n}$ are independent uniform random variables, $x \boxplus y = z$, $\Delta_1(x, x')$, $\Delta_1(y, y')$ and $\Delta_1(z, z')$ are constants in $\mathbb{F}_{2^n}$ and $x' = (x \oplus \Delta_1(x, x')) \ggg 1$, $y' = (y \oplus \Delta_1(y, y')) \ggg 1$, $z' = (z \oplus \Delta_1(z, z')) \ggg 1$. Then,*

$$\begin{aligned}&Pr[x' \boxplus y' = z']\\ &\quad = 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \oplus 1 \preceq SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))} \cdot 2^{-|SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))|} \cdot 2^{-3}\\ &+ 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \preceq SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))} \cdot 2^{-|SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))|} \cdot 2^{-1.415}, \quad (2.2)\end{aligned}$$

where

$$\delta_z = L(\Delta_1(x, x')),\ \delta_z = L(\Delta_1(y, y'))\ \textit{and}\ \delta_z = L(\Delta_1(z, z')).$$

Proof. The complete proof can be found in [2, Theorem 1]. □

When modular chains are present, the assumption of independent uniform random inputs does not hold and in that case, the probability of propagation cannot be calculated with the above theorem.

2.3 Conclusion

The importance of symmetric key primitives is explained, especially in the case of block ciphers, along with the desirable properties of such primitives. One requirement of a primitive is to be secure, however, for most primitives security proofs do not exist. Because of this, the necessary requirement for a primitive is to be resistant against all known cryptanalytic techniques. Three statistical techniques are described, differential cryptanalysis, rotational cryptanalysis and Rotational-XOR cryptanalysis, which is a generalization of differential and rotational cryptanalysis.

Chapter 3

Automated Search of RX-characteristics

The complete theory to compute the probability of a specific RX-characteristic through an ARX primitive is given in the previous chapter. However, the question remains on how to find highly probable RX-characteristics, which is only a small portion of all possible RX-characteristics, especially when the number of rounds is large. Manually searching for highly probable RX-characteristics in ARX primitives is possible, but is slow and of low efficiency. To save time and increase efficiency, automating this process through computer programs is natural. In this chapter we describe a possible approach to automate the search of highly probable RX-characteristics in ARX primitives.

Automated search has been successful for other cryptanalytic techniques, for example differential cryptanalysis. The problem of finding a highly probable differential characteristic can be reformulated as a series of Boolean Satisfiability Problems (SAT problems) [10, 14] or as Mixed-Integer Linear Programming (MILP) [7]. In this thesis, the SAT-approach is taken.

3.1 Boolean Satisfiability problem

A boolean satisfiability problem is the problem of determining if there exists a solution for a given boolean expression. A boolean expression consists of variables, which can take the logical values TRUE (T) or FALSE (F), three logical operations, namely the logical OR ($\vee$), logical AND ($\wedge$) and logical NOT ($\neg$) and parentheses, which decide on the order of operations in the boolean expression. The truth tables for the logical OR and logical AND are given in Table 3.1, the truth table for the logical NOT is given in Table 3.2.

An example of a boolean expression is the following:

$$(p_1 \vee p_2) \wedge (p_1 \vee p_3).$$

When p_1 is true, the boolean expression evaluates to true, independent of the values of p_2 and p_3. When p_1 and p_2 are false, the boolean expression evaluates to

p	q	$p \vee q$	$p \wedge q$
F	F	F	F
F	T	T	F
F	F	T	F
T	T	T	T

Table 3.1: Truth tables of the logical AND and logical OR.

p	$\neg p$
F	T
T	F

Table 3.2: Truth table of the logical NOT.

false. One can now ask the question: can a given boolean expression evaluate to true if the variables are consistently replaced by the values TRUE or FALSE. In the above example, the answer is yes and the expression is satisfiable. The expression $p_1 \wedge (\neg p_1)$ on the other hand is unsatisfiable, because no value of p_1 can make this expression satisfiable.

The boolean satisfiability problem is to determine if a given expression is satisfiable or unsatisfiable. This problem was one of the first problems that was proven to be NP-complete. Because of this, there is currently no known algorithm that solves all SAT problems in polynomial time. However, the problem is extensively studied and heuristic SAT-solvers are able to solve a lot of problem instances that arise in practical problems, ranging from artificial intelligence to cryptography [12].

3.2 From RX-characteristic to SAT problem

Modern SAT-solvers often work with n-bit Boolean vectors, a vector of n Boolean variables. In this case an RX-difference can be expressed as an n-bit Boolean vector. Suppose x, y and z are n-bit vectors, some operations and characteristic functions are then easily expressed with Boolean symbols. For example, if $z = x \ggg \alpha$ the equivalent Boolean expression is $z = (x^{n-1}, \ldots, x^1, x^0) = (x^{\alpha-1}, \ldots, x^0, x^n, \ldots x^{\alpha+1}, x^\alpha)$. If $z = x \oplus y$, the equivalent Boolean expression is $z = (x \vee y) \wedge (\neg x \vee \neg y)$. The characteristic function $1_{x \preceq y}$ is equivalent with the evaluation of the Boolean formula $y = x \vee y$. With this knowledge, it is clear that the deterministic rules of propagation for RX-differences, derived in section 2.2.5, can be expressed as Boolean expressions, composed of Boolean operations and n-bit vectors.

The propagation of RX-differences through modular addition is a probabilistic rule and Boolean constraints can be derived from Theorem 2.2. Suppose that $\Delta_1(x)$ and $\Delta_1(y)$ are n-bit vectors that express RX-differences and the modular addition of these two RX-differences is expressed as the n-bit vector $\Delta_1(z)$. The following Boolean expression should be satisfied for the propagation of RX-differences to be possible:

$$(i \oplus shl)(\Delta_1 x \oplus \Delta_1 y \oplus \Delta_1 z) \oplus 1 \preceq shl((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \oplus \Delta_1 z))$$
$$\text{OR}$$
$$(i \oplus shl)(\Delta_1 x \oplus \Delta_1 y \oplus \Delta_1 z) \preceq shl((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \Delta_1 z)).$$

The above equation expresses a possible transition of RX-differences through the modular addition, but it tells nothing about the probability of that transition. From Theorem 2.2, an expression for the probability can be derived. This probability is given by 2^{-w} with the w equal to the following expression:

$$w = \begin{cases} |SHL((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \oplus \Delta_1 z))| + 3 & \Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 1 \\ |SHL((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \oplus \Delta_1 z))| + 1.415 & \Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 0. \end{cases}$$

Because SAT solvers often cannot work with real numbers, the 1.415 is rounded to 1.5 in the formula above and the expressions are scaled by a factor 2. The constraint concerned with the probability through modular addition then becomes:

$$w_c = \begin{cases} 2|SHL((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \oplus \Delta_1 z))| + 6 & \Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 1 \\ 2|SHL((\Delta_1 x \oplus \Delta_1 z)|(\Delta_1 y \oplus \Delta_1 z))| + 3 & \Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 0. \end{cases}$$

Notice that the exact probability can be derived from this constraint if the SAT-solver returns a solution. If $\Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 1$, then $w = w_c/2$, if $\Delta_1^0 x \oplus \Delta_1^0 y \oplus \Delta_1^0 z = 0$, then $w = w_c/2 - 0.085$.

3.3 Application to SPECK

3.3.1 Description of SPECK

The Speck family of block ciphers is a collection of 10 block ciphers designed by the NSA in 2013 to fulfil the need for efficient, flexible, lightweight block ciphers with varying block length and key length. The parameters for every version of the Speck family are given in Table 3.3.

The output $(x_{i+1}||y_{i+1})$ of one round of encryption can be calculated from the input $(x_i||y_i)$ as follows:

$$x_{i+1} = ((x_i \ggg \alpha) \boxplus y_i) \oplus k_i \tag{3.1}$$
$$y_{i+1} = (y_i \lll \beta) \oplus x_{i+1}. \tag{3.2}$$

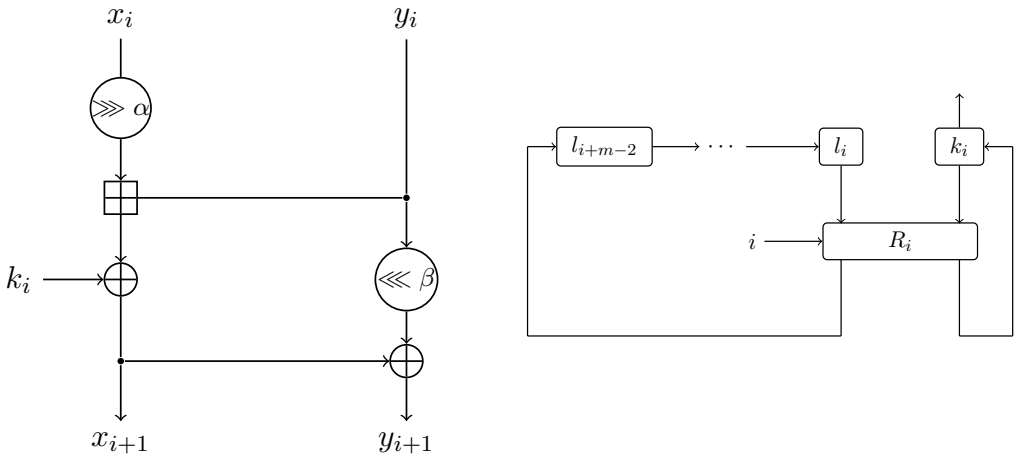
The Speck key schedule reuses the round function to generate the next round key k_{i+1} from the previous one k_i:

Block Size $2n$	Key Size mn	rot α	rot β	Number of rounds T
32	64	7	2	22
48	72	8	3	22
	96			23
64	96	8	3	26
	128			27
96	96	8	3	28
	144			29
128	128	8	3	32
	192			33
	256			34

Table 3.3: Parameters for different versions of Speck.

$$l_{i+m-1} = (k_i \boxplus (l_i \ggg \alpha)) \oplus i \tag{3.3}$$
$$k_{i+1} = (k_i \lll \beta) \oplus l_{i+m-1}. \tag{3.4}$$

A visual representation of the Speck round function and the key schedule is given in Figure 3.1a and Figure 3.1b respectively. For more information, see [3].



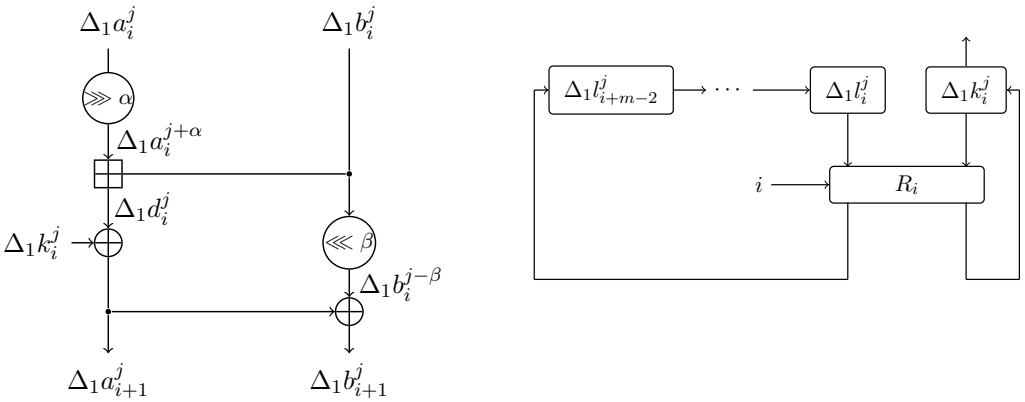
(a) Round function of Speck (b) Key Schedule of Speck

Figure 3.1: Description of Speck

3.3.2 Boolean expressions for RX-characteristics

The notations in this section are as follows: $\Delta_1 x_i^j$ is an RX-difference in round i and $\Delta_1 x_i^{j+\alpha}$ is the RX-difference $\Delta_1 x_i^j$ bitwise rotated over an amount α.

The following theorem is a corollary from the Boolean constraints for RX-differences derived in section 3.2 and the round function of Speck.



(a) Notation of RX-differences in the round function of SPECK.

(b) Notation of RX-differences in the key schedule of SPECK.

Figure 3.2: Notation of RX-differences in SPECK.

Theorem 3 (Boolean constraints concerning the round function to find a valid r-round RX-characteristic for SPECK)**.** *To obtain a valid r-round RX-characteristic for* SPECK*, the boolean bits of RX-differences in round i, with a notation used as in Figure 3.2a, need to satisfy the following constraints:*

$$(I \oplus SHL)(\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 b_i^j \oplus \Delta_1 d_i^j) \oplus 1 \preceq SHL((\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 b_i^j \oplus \Delta_1 d_i^j))$$

$$OR$$

$$(I \oplus SHL)(\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 b_i^j \oplus \Delta_1 d_i^j) \preceq SHL((\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 b_i^j \oplus \Delta_1 d_i^j)),$$
$$\Delta_1 a_{i+1}^j = \Delta_1 d_i^j \oplus \Delta_1 k_i^j,$$
$$\Delta_1 b_{i+1}^j = \Delta_1 b_i^{j-\beta} \oplus \Delta_1 a_{i+1}^j,$$

where $0 \leq i \leq r$.

Likewise, the following theorem concerns the propagation of RX-differences through the SPECK key schedule.

Theorem 4 (Boolean constraints concerning the key schedule to find a valid r-round RX-characteristic for SPECK)**.** *To obtain a valid r-round RX-characteristic for* SPECK*, the boolean bits of RX-differences in round i of the key schedule, with a notation used as in Figure 3.2b, need to satisfy the following constraints:*

$$(I \oplus SHL)(\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 k_i^j \oplus \Delta_1 e_i^j) \oplus 1 \preceq SHL((\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 e_i^j)|(\Delta_1 k_i^j \oplus \Delta_1 e_i^j))$$

$$OR$$

$$(I \oplus SHL)(\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 k_i^j \oplus \Delta_1 e_i^j) \preceq SHL((\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 e_i^j)|(\Delta_1 k_i^j \oplus \Delta_1 e_i^j)),$$
$$\Delta_1 l_{i+1}^j = \Delta_1 e_i^j \oplus (c \lll 1) \oplus c,$$
$$\Delta_1 k_{i+1}^j = \Delta_1 k_i^{j-\beta} \oplus \Delta_1 l_{i+1}^j,$$

where $0 \leq i \leq r$ *and* $\Delta_1 e_i^j = \Delta_1 l_{i+1}^j \oplus (c \lll 1) \oplus c$.

The above two theorems allow to find valid RX-characteristics with a SAT-solver, the following theorem allows to calculate the data probability of an RX-characteristic.

Theorem 5 (Data probability of a valid r-round RX-characteristic for Speck)**.** *The probability of the a valid RX-characteristic is given by*

$$2^{-\sum_{i=1}^{r} w_i}, \tag{3.5}$$

where

$$w_i = \begin{cases} |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))| + 3 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 1 \\ |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))| + 1.415 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 0 \end{cases}.$$

if the input of each modular addition is an independent uniform random variable.

The key probability of an r-round RX-characteristic can be calculated with the following theorem.

Theorem 6 (Key probability of a valid r-round RX-characteristic for Speck)**.** *The key probability of a valid RX-characteristic is given by*

$$2^{-\sum_{i=1}^{r} w_i^k}, \tag{3.6}$$

where

$$w_i^k = \begin{cases} |SHL((\Delta_1 l_{i+7}^r \oplus \Delta_1 l_i^{r+1})|(\Delta_1 k_i^r \oplus \Delta_1 l_i^{r+1}))| + 3 & \Delta_1 l_i^\alpha \oplus \Delta_1 k_i^0 \oplus \Delta_1 e_i^0 = 1 \\ |SHL((\Delta_1 l_{i+7}^r \oplus \Delta_1 l_i^{r+1})|(\Delta_1 k_i^r \oplus \Delta_1 l_i^{r+1}))| + 1.415 & \Delta_1 l_i^\alpha \oplus \Delta_1 k_i^0 \oplus \Delta_1 e_i^0 = 0 \end{cases}.$$

if the input of each modular addition is an independent uniform random variable.

In the automated search program, these probabilities are expressed as objective functions $\sum_{i=1}^{r} w_{i,c}$ and $\sum_{i=1}^{r} w_{i,c}^k$ with $w_{i,c}$ and $w_{i,c}^k$ defined as w_c in the previous section.

3.3.3 Search strategy for RX-characteristics

To find good RX-characteristics for Speck, the objective functions $\sum_{i=1}^{r} w_{i,c}$ and $\sum_{i=1}^{r} w_{i,c}^k$ need to be minimized. Minimizing $\sum_{i=1}^{r} w_{i,c}$, maximizes the data probability as defined in theorem 5 and leads to fewer data requirements. For example, if $2^{-\sum_{i=1}^{r} w_i} = 2^{-\psi}$, then at least 2^ψ plaintext-ciphertext pairs need to be available to observe the RX-characteristic. Minimizing $\sum_{i=1}^{r} w_{i,c}^k$ on the other hand, maximizes the key probability

as defined in theorem 6 and increases the chance to find two related keys. The size of the weak key-class can then be calculated as $2^{-\sum_{i=1}^{r} w_i + \#\{\text{master key bits}\}}$. This makes it clear that the upper bound on $\sum_{i=1}^{r} w_i^k$ the number of bits in the master key is and that the upper bound on $\sum_{i=1}^{r} w_i$ the block size is. Minimizing the objective functions happens as follows: We check if the SAT problem is satisfiable for a certain value of the objective function. If it is satisfiable, we divide the value of the objective function by two and solve the SAT problem again. If it is not satisfiable, we add the upper bound to the value of the objective function, divide by two and solve the SAT problem again. Using this method, the objective function can be minimized by solving as few SAT problems as possible. In this thesis, the data probability is first minimized before the key probability. Other trade-offs are possible and are worth investigating.

3.4 Conclusion

A definition of the Boolean satisfiability problem is given and constraints in the form of Boolean expressions are derived for the propagation of RX-differences throughout the different operations in ARX designs. This includes a Boolean expression for the propagation probability through modular addition. By solving these expressions, valid RX-characteristics with a lower bound on the probability can be derived for a certain ARX primitive. Explicit expressions to find valid RX-characteristics with a specified lower bound on the probability are derived for the Speck family of block ciphers.

Chapter 4

Results of the automated search for SPECK

In this chapter the results of the program for finding optimal RX-characteristics for different versions of Speck are given and compared with previous results on statistical cryptanalysis of Speck. A short description of the used hardware and software is also given.

4.1 Used software and hardware

The "STP Constraint Solver" is used for the automated search. It is a constraint solver that accepts expressions in bit vector arithmetic and can solve many kinds of problems, *e.g.* problems generated by program analysis tools, theorem provers, automated bug finders, cryptographic algorithms, intelligent fuzzers and model checkers. The STP program accepts problems in SMT-LIB2 format, a language in which the bit vector expressions can be stated along with the objective functions derived in the previous chapter. The STP Constraint solver parses the given problem in SMT-LIB2 format to a SAT-problem in CNF format. This SAT-problem in CNF format is then solved by a SAT-solver, for example "MiniSat" or "CryptoMiniSat". In this case we use "CryptoMinisat" (version 5), it uses heuristics that solve problems with XOR operations faster on average and in our problems XOR operations are common. On top of that, it is a SAT-solver competitive with other SAT-solvers, it won some categories of the SAT Race in the previous years.

The hardware used is the "Google cloud platform" with 24 cores and 96Gb of RAM. The system was running Ubuntu 16.04 LTS.

4.2 Probability of optimal RX-characteristics for SPECK

For five versions of Speck, the optimal RX-characteristics are found for an increasing number of rounds, until no RX-characteristics with sufficient data probability could

R	Data probability	Size weak key-class
1	$2^{-1.415}$	2^{64}
2	$2^{-2.83}$	$2^{62.55}$
3	$2^{-4.245}$	$2^{61.17}$
4	$2^{-5.66}$	$2^{58.755}$
5	$2^{-7.075}$	$2^{56.34}$
6	$2^{-8.49}$	$2^{53.925}$
7	$2^{-11.905}$	$2^{51.51}$
8	$2^{-13.32}$	$2^{38.01}$
9	$2^{-15.735}$	$2^{31.51}$
10	$2^{-19.15}$	$2^{28.095}$
11	$2^{-22.065}$	$2^{18.68}$
12	$2^{-25.565}$	$2^{4.925}$

Table 4.1: Probability and size of weak key-class of optimal RX-characteristic through Speck32/64.

be found any more or the program used too many computational resources. For Speck32/64, the results are given in Table 4.1, for Speck48/96 in Table 4.2, for Speck64/128 in Table 4.3, for Speck96/144 in Table 4.4 and for Speck128/256 in Table 4.5. In the case of Speck32/64, no RX-characteristics with sufficient data probability (2^{-32}) could be found if the number of rounds was bigger than 12. For the other versions of Speck that we studied, the computational resources to further increase the number of rounds were too demanding.

In this thesis we are mainly concerned with the data probability of the RX-characteristics. For that reason and because of requirements on computational resources, results for the other versions of Speck were not computed. RX-characteristics for other versions with equivalent word size but a lower number of keywords can never have better data probabilities than the versions we studied.

R	Data probability	Size weak key-class
1	$2^{-1.415}$	2^{96}
2	$2^{-2.83}$	$2^{94.585}$
3	$2^{-4.245}$	$2^{93.17}$
4	$2^{-5.66}$	$2^{90.855}$
5	$2^{-7.075}$	$2^{87.34}$
6	$2^{-9.49}$	$2^{82.925}$
7	$2^{-12.49}$	$2^{75.34}$
8	$2^{-13.905}$	$2^{69.34}$
9	$2^{-16.32}$	$2^{63.51}$
10	$2^{-18.735}$	$2^{57.925}$
11	$2^{-23.15}$	$2^{14.925}$
12	$2^{-26.565}$	$2^{43.51}$
13	$2^{-31.98}$	$2^{24.51}$
14	$2^{-37.395}$	$2^{0.34}$

Table 4.2: Probability and size of weak key-class of optimal RX-characteristics throughout Speck48/96.

R	Data probability	Size weak key-class
1	$2^{-1.415}$	2^{128}
2	$2^{-2.83}$	$2^{126.585}$
3	$2^{-4.245}$	$2^{125.17}$
4	$2^{-5.66}$	$2^{122.755}$
5	$2^{-7.075}$	$2^{119.34}$
6	$2^{-9.49}$	$2^{114.925}$
7	$2^{-12.49}$	$2^{107.34}$
8	$2^{-13.905}$	$2^{101.34}$
9	$2^{-16.32}$	$2^{98.51}$
10	$2^{-18.735}$	$2^{89.925}$
11	$2^{-23.15}$	$2^{30.925}$
12	$2^{-26.565}$	$2^{75.51}$
13	$2^{-31.98}$	$2^{54.51}$

Table 4.3: Probability and size of weak key-class of optimal RX-characteristics throughout Speck64/128.

R	Data probability	Size weak key-class
1	$2^{-1.415}$	2^{144}
2	$2^{-2.83}$	$2^{142.585}$
3	$2^{-4.245}$	$2^{141.17}$
4	$2^{-5.66}$	$2^{138.755}$
5	$2^{-7.075}$	$2^{133.755}$
6	$2^{-10.49}$	$2^{131.34}$
7	$2^{-12.49}$	$2^{123.34}$
8	$2^{-15.905}$	$2^{112.755}$
9	$2^{-19.32}$	$2^{105.925}$
10	$2^{-22.735}$	$2^{98.34}$
11	$2^{-26.15}$	$2^{90.34}$
12	$2^{-31.565}$	$2^{8.975}$
13	$2^{-37.98}$	$2^{37.925}$

Table 4.4: Probability and size of weak key-class of optimal RX-characteristics throughout Speck96/144.

R	Data probability	Size weak key-class
1	$2^{-1.415}$	2^{256}
2	$2^{-2.83}$	$2^{254.585}$
3	$2^{-4.245}$	$2^{253.17}$
4	$2^{-5.66}$	$2^{250.755}$
5	$2^{-7.075}$	$2^{247.34}$
6	$2^{-9.49}$	$2^{242.925}$
7	$2^{-12.49}$	$2^{235.34}$
8	$2^{-13.905}$	$2^{229.34}$
9	$2^{-16.32}$	$2^{226.51}$
10	$2^{-18.735}$	$2^{217.925}$
11	$2^{-23.15}$	$2^{94.925}$
12	$2^{-26.565}$	$2^{203.51}$
13	$2^{-31.98}$	$2^{182.51}$

Table 4.5: Probability and size of weak key-class of optimal RX-characteristics throughout Speck128/256.

4.3 Optimal RX-characteristics for SPECK

In this section, the exact values of RX-characteristics throughout the Speck block cipher are given before and after each round for the different versions of Speck. For Speck32/64, the results are given in Table 4.6, for Speck48/96 in Table 4.7, for Speck64/128 in Table 4.8, for Speck96/144 in Table 4.9 and for Speck128/256 in Table 4.10. The RX-characteristics took respectively 796, 470852, 44473, 1037717 and 25888 seconds. From these results, it can be noticed that the search strategy explained in Section 3.3.3 can be improved by a better prediction of the data probability of the optimal RX-characteristic in round $i+1$ if the data probability of the optimal RX-characteristic in round i is known. There also seem to be connections between optimal RX-characteristics of different versions of Speck. Especially the Tables 4.8 of Speck64/128 and 4.10 of Speck128/256 seem similar, both versions of Speck have the same rotation parameters α and β and the block size of Speck128/256 is exactly the double of Speck64/128. These similarities might be worth investigating further.

R	$\Delta_1 a$	$\Delta_1 b$	$\Delta_1 l$	$\Delta_1 k$
0	0050	2000	8000	0000
1	8000	0000	8102	0100
2	0000	0000	0100	0000
3	0000	0000	0100	0000
4	0000	0000	0400	0001
5	0000	0000	0000	0000
6	0000	0000	0001	0001
7	0000	0000	0004	0200
8	0200	0200	0001	0206
9	0000	0800	0204	0800
10	0000	2000	0a06	2001
11	0000	8000	0018	a40e
12	240e	240c	/	/

Table 4.6: Optimal RX-characteristic for 12-round Speck32/64.

R	$\Delta_1 a$	$\Delta_1 b$	$\Delta_1 l$	$\Delta_1 k$
0	000000	000008	020800	000008
1	000000	000040	104000	000240
2	000200	000000	03f800	000000
3	000000	000000	000200	000000
4	000000	000000	001200	000000
5	000000	000000	000000	000000
6	000000	000000	000000	000001
7	000000	000000	000000	000001
8	000000	000000	000001	000000
9	000000	000000	000009	010018
10	010019	010019	000008	1000e0
11	0801e8	000120	010018	680021
12	000900	000000	180020	000009
13	000000	000000	e80721	202844
14	202844	202844	/	/

Table 4.7: Optimal RX-characteristic for 14-round Speck48/96.

R	$\Delta_1 a$	$\Delta_1 b$	$\Delta_1 l$	$\Delta_1 k$
0	00000000	00000008	00020800	00000008
1	00000000	00000040	00104200	00000240
2	00000200	00000000	00000200	00000000
3	00000000	00000000	00000200	00000000
4	00000000	00000000	00001200	00000000
5	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000001
7	00000000	00000000	00000000	00000001
8	00000000	00000000	00000001	00000000
9	00000001	00000001	00000009	07000018
10	00000018	00000010	00000008	380000f1
11	00000080	00000000	07000018	80000001
12	00000000	00000000	00000031	88010011
13	88010011	88010011	/	/

Table 4.8: Optimal RX-characteristic for 12-round Speck64/128.

R	$\Delta_1 a$	$\Delta_1 b$	$\Delta_1 l$	$\Delta_1 k$
0	000002080000	000000000001	000012480000	000000020801
1	000000000000	000000000008	000000020a00	000000000008
2	000000000000	000000000040	000000104000	000000000240
3	000000000200	000000000000	000000000200	000000000000
4	000000000000	000000000000	000000001200	000000000000
5	000000000000	000000000000	000000000000	000000000000
6	000000000000	000000000000	000000000000	000000000001
7	000000000000	000000000000	000000000001	000000000002
8	000000000003	000000000003	00000000000a	07000000001e
9	000000000018	000000000000	07000000000e	390000000001
10	010000000000	010000000000	0100000000f1	090100000010
11	080000000010	000000000010	c10100000019	100800000091
12	000000000080	000000000000	580000000011	767707000425
13	f67707000425	f67707000425	/	/

Table 4.9: Optimal RX-characteristic for 13-round Speck96/144.

R	$\Delta_1 a$	$\Delta_1 b$	$\Delta_1 l$	$\Delta_1 k$
0	0000000000000000	0000000000000008	0000000000020800	0000000000000008
1	0000000000000000	0000000000000040	0000000000104200	0000000000000240
2	0000000000000200	0000000000000000	0000000000000200	0000000000000000
3	0000000000000000	0000000000000000	0000000000000200	0000000000000000
4	0000000000000000	0000000000000000	0000000000001200	0000000000000000
5	0000000000000000	0000000000000000	0000000000000000	0000000000000000
6	0000000000000000	0000000000000000	0000000000000000	0000000000000001
7	0000000000000000	0000000000000000	0000000000000000	0000000000000001
8	0000000000000000	0000000000000000	0000000000000001	0000000000000000
9	0000000000000001	0000000000000001	0000000000000009	0700000000000018
10	0000000000000018	0000000000000010	0000000000000008	0700000000000018
11	0000000000000080	0000000000000000	0700000000000018	8000000000000001
12	0000000000000000	0000000000000000	0000000000000031	8801000000000011
13	8801000000000011	8801000000000011	/	/

Table 4.10: Optimal RX-characteristic for 13-round Speck128/256.

R	Theoretical data probability	Empirical data probability
1	$2^{-3.415}$	$2^{-3.415}$
2	$2^{-5.83}$	$2^{-5.79}$
3	$2^{-7.245}$	$2^{-7.245}$
4	$2^{-8.66}$	$2^{-8.717}$
5	$2^{-10.075}$	$2^{-10.08}$
6	$2^{-11.49}$	$2^{-11.52}$
7	$2^{-12.905}$	$2^{-12.90}$
8	$2^{-14.32}$	$2^{-14.35}$
9	$2^{-15.735}$	$2^{-15.81}$

Table 4.11: Comparison of theoretical probability and empirical probability of 9-round RX-characteristic in Speck32/64.

4.4 Experimental verification

In Table 4.11, an experimental verification of a 9-round RX-trail is done in Speck32/64. This experiment was done by encrypting 2^{16} plaintext pairs with a specific RX-difference. After every round of encryption, we check if the pair satisfies the RX-trail up until that round. The empirical probability after every round can then be calculated by dividing the pairs that satisfy the trail divided by the pairs that do not. From the results in Table 4.11, we can conclude that the empirical data probability follows the calculated data probability very closely, the deviations from the calculated data probability are very minimal and seem to be noise from taking only 2^{16} plaintext pairs instead of all possible 2^{32} plaintext pairs.

4.5 Comparison to other cryptanalytic techniques

We compare optimal RX-characteristics with optimal linear trails for different versions of Speck. The optimal linear trails for different versions of Speck are found in [10]. To make a fair comparison, the correlation of the optimal linear trails needs to be squared and then we can compare them to the data probability of the RX-characteristics. In Tables 4.12, 4.13, 4.14, 4.15 and 4.16 the results are given. We can clearly see that RX-cryptanalysis outperforms linear cryptanalysis on the aspect of data requirements. Comparison with differential cryptanalysis gives similar results. In Table 4.17 a comparison is given for the maximum number of rounds that are analysed for RX-cryptanalysis and linear cryptanalysis respectively. The number of analysed rounds for RX-cryptanalysis is in general higher than for linear cryptanalysis.

R	Data prob.	Sq. cor.
1	$2^{-1.415}$	1
2	$2^{-2.83}$	1
3	$2^{-4.245}$	2^{-2}
4	$2^{-5.66}$	2^{-6}
5	$2^{-7.075}$	2^{-7}
6	$2^{-8.49}$	2^{-14}
7	$2^{-11.905}$	2^{-18}
8	$2^{-13.32}$	2^{-24}
9	$2^{-15.735}$	2^{-28}
10	$2^{-19.15}$	2^{-34}
11	$2^{-22.065}$	2^{-38}
12	$2^{-25.565}$	2^{-40}

Table 4.12: Comparison of optimal RX-characteristics to optimal linear trails throughout Speck32/64.

R	Data prob.	Sq. cor.
1	$2^{-1.415}$	1
2	$2^{-2.83}$	1
3	$2^{-4.245}$	2^{-2}
4	$2^{-5.66}$	2^{-6}
5	$2^{-7.075}$	2^{-12}
6	$2^{-9.49}$	2^{-16}
7	$2^{-12.49}$	2^{-24}
8	$2^{-13.905}$	2^{-30}
9	$2^{-16.32}$	2^{-38}
10	$2^{-18.735}$	2^{-44}
11	$2^{-23.15}$	2^{-50}

Table 4.13: Comparison of optimal RX-characteristics to optimal linear trails throughout Speck48/96.

R	Data prob.	Sq. cor.
1	$2^{-1.415}$	1
2	$2^{-2.83}$	1
3	$2^{-4.245}$	2^{-2}
4	$2^{-5.66}$	2^{-6}
5	$2^{-7.075}$	2^{-12}
6	$2^{-9.49}$	2^{-18}
7	$2^{-12.49}$	2^{-26}
8	$2^{-13.905}$	2^{-34}
9	$2^{-16.32}$	2^{-38}
10	$2^{-18.735}$	2^{-42}
11	$2^{-23.15}$	2^{-48}
12	$2^{-26.565}$	2^{-54}
13	$2^{-31.98}$	2^{-60}

Table 4.14: Comparison of optimal RX-characteristics to optimal linear trails throughout Speck64/128.

R	Data prob.	Sq. cor.
1	$2^{-1.415}$	1
2	$2^{-2.83}$	1
3	$2^{-4.245}$	2^{-2}
4	$2^{-5.66}$	2^{-6}
5	$2^{-7.075}$	2^{-12}
6	$2^{-10.49}$	2^{-18}
7	$2^{-12.49}$	2^{-26}
8	$2^{-15.905}$	2^{-36}
9	$2^{-19.32}$	2^{-44}

Table 4.15: Comparison of optimal RX-characteristics to optimal linear trails throughout Speck96/144.

R	Data prob.	Sq. cor.
1	$2^{-1.415}$	1
2	$2^{-2.83}$	1
3	$2^{-4.245}$	2^{-2}
4	$2^{-5.66}$	2^{-6}
5	$2^{-7.075}$	2^{-12}
6	$2^{-9.49}$	2^{-18}
7	$2^{-12.49}$	2^{-26}
8	$2^{-13.905}$	2^{-36}
9	$2^{-16.32}$	2^{-44}

Table 4.16: Comparison of optimal RX-characteristics to optimal linear trails throughout SPECK128/256.

SPECK-version	cryptanalysis	Nb. of rounds	Prob.
SPECK32/64	RX-cryptanalysis	12	$2^{-25.565}$
SPECK32/64	linear cryptanalysis	9	2^{-28}
SPECK48/96	RX-cryptanalysis	14	$2^{-23.15}$
SPECK48/96	linear cryptanalysis	10	2^{-44}
SPECK64/128	RX-cryptanalysis	12	$2^{-31.98}$
SPECK64/128	linear cryptanalysis	13	2^{-60}
SPECK96/144	RX-cryptanalysis	13	$2^{-19.32}$
SPECK96/144	linear cryptanalysis	9	2^{-44}
SPECK128/256	RX-cryptanalysis	13	$2^{-16.32}$
SPECK128/256	linear cryptanalysis	9	2^{-44}

Table 4.17: Comparison of the maximum number of rounds analysed with linear cryptanalysis and RX-cryptanalysis.

4.6 Conclusion

In this chapter, the results of the automatic search program for five versions of Speck are described and compared against results of other types of cryptanalysis for round-reduced Speck. The conclusion is that Rotational-XOR cryptanalysis in general has a higher data probability than other techniques, however two related keys that are part of the weak key-class need to be available for the technique to work. Future research can include finding RX-characteristics with a lower data probability but larger weak key-class, finding optimal RX-characteristics for other ARX primitives, finding longer RX-characteristics or finding ways to speed up the automatic search.

Chapter 5

Conclusion

Various statistical cryptanalytic techniques exist for ARX primitives. Automated search to find statistical relations for some of these techniques has been published and led to satisfactory results. In this thesis, we describe how automatic search can be performed for rotational-XOR cryptanalysis, a recently proposed statistical technique for ARX primitives, by solving a series of SAT problems. This automatic search is applied to round-reduced versions of the Speck block cipher family, the results are reported and compared to previous results.

The general conclusion is that rotational-XOR cryptanalysis for round-reduced Speck has significantly less data requirements than other statistical techniques, albeit one condition for rotational-XOR cryptanalysis to succeed is that two related keys out of a weak key-class are available. The automatic search for RX-characteristics in this thesis was focused on finding the lowest data requirements possible, which sometimes led to a small weak key-class.

Other trade-offs between data requirements and the size of the weak key-class are possible and may be worth investigating. Other future research can include applying automatic search for Rotational-XOR cryptanalysis on other ARX primitives and obtaining longer RX-characteristics by concatenating two shorter RX-characteristics.

Appendices

Appendix A

Paper submitted to the "2017 Symposium on Information Theory and Signal Processing in the Benelux"

The paper below is an 8-page paper accepted for the "2017 Symposium on Information Theory and Signal Processing in the Benelux". The Symposium took place in Delft, the Netherlands on 11 and 12 May 2017. It aimed at sharing ideas, experiences and information on a lot of topics, one of them being cryptography.

An Automated Tool for Rotational-XOR Cryptanalysis of ARX-based Primitives

Glenn De Witte Tomer Ashur Yunwen Liu
K.U. Leuven
imec-COSIC
Leuven, Belgium
glenn.dewitte@student.kuleuven.be [tomer.ashur, yunwen.liu]@esat.kuleuven.be

Abstract

In ARX structures, constants that are not rotational invariant are often injected into the state, in the form of round constants or as a result of using a fixed key. Rotational cryptanalysis cannot deal with such constants. Rotational cryptanalysis in the presence of constants, also known as rotational-XOR cryptanalysis, is a recently proposed statistical technique to attack ARX primitives. The newly proposed technique investigates how constants affect rotational cryptanalysis by introducing the notion of an RX-difference, which generalizes the idea of a rotational difference. Previously, a 7-round distinguisher for Speck, an ARX block cipher designed by the NSA, was found, mainly to demonstrate the proposed technique.

In this paper, it is shown that longer distinguishers exist for Speck32/64, as well as other versions of speck, by using rotational-XOR cryptanalysis . This was done by means of an automated search tool. More specifically, the propagation of RX-differences through ARX structures are transformed into bitwise equations and these equations are then solved by a SAT-solver. Using this method, distinguishers with more rounds than previously reported are found for Speck32/64 and Speck48/96, as well as a 13-round distinguisher for Speck96/144 with a higher probability than previously reported, under the condition that two related keys are available.

Key words: Automated Search Tools, ARX primitive, Speck, Rotational Cryptanalysis, Rotational-XOR Cryptanalysis, Lightweight Cryptography.

1 Introduction

In the advent of the *Internet of Things*, symmetric primitives not only need to be secure, but also allow for an efficient implementation on resource-constrained devices. A common and popular design, which allows for such implementations is the ARX design. ARX stands for modular Addition, cyclic Rotation and XOR. By using these three operations, good diffusion and confusion can be achieved, as well as a cheap and fast implementation in both hardware and software. Some recent examples of ARX designs include stream ciphers e.g. Salsa20 [4] and block ciphers e.g. TEA [13], Speck [3].

Even though elegant and small ARX designs are often used in practice, the security analysis of these designs often lags behind. No security proofs of ARX structures are known for most cryptanalytic techniques, for example differential or linear cryptanalysis. An automatic search for optimal differential characteristics [6, 12] and linear trails [10] is often the only way to check whether a specific ARX design is resistant against those two types of cryptanalytic techniques.

Another cryptanalytic technique, only applicable against ARX designs that use (almost) rotation invariant constants, is rotational cryptanalysis [8]. This technique studies the propagation of a rotational pair through ARX primitives. First it seemed the probability of success was solely dependent on the number of additions in the primitive, because both cyclic rotation and XOR propagate a rotational pair with probability 1 and modular addition with a probability up to 3/8, only relying on the rotational offset [8]. However, because the Markov assumption used to compute the propagation probability through modular addition is invalid when chained modular additions are present in the primitive, more correct success probabilities for primitives with such chains are computed in [9].

One important drawback of rotational cryptanalysis is that it cannot seem to deal with constants introduced into the state. The idea of not only keeping track of a rotational offset, but also a translation that acts as a correction to make a pair a rotational pair, is present in the original paper on rotational cryptanalysis [8], but no formulas to compute the success of propagation are derived. In [2], this idea is formalized in the notion of an RX-difference and probabilities for the propagation of RX-differences through ARX primitives that satisfy the Markov property and have no modular chains are rigorously defined.

The propagation of these RX-differences through ARX primitives can be reformulated as a SAT-problem and an automated search tool based on SAT-solvers is developed to find good RX-characteristics. Automated search tools based on SAT-solvers have been made for linear cryptanlysis [1] and differential cryptanalysis [5, 12] and have proven to be successful. This paper illustrates the effectiveness of the automated search tool rotational-XOR cryptanalysis by showing that distinguishers exist of 12 and 13 rounds for SPECK versions with word lengths of respectively 16, and 24 bits, under the condition that two related keys are available. Previous distinguishers were only up to 10 and 11 rounds. A 13-round distinguisher for SPECK96/144 with a higher probability than previously reported is also presented, again under the condition that two related keys are available.

The rest of this paper is organized as follows: in Section 2 the definition of an RX-difference is recalled, along with the propagation probabilities through operations of ARX structures. Furthermore a concise description of the SPECK block cipher family is given. In Section 3 is explained how SAT-solvers can be used to find optimal RX-characteristics and distinguishers are presented for different versions of SPECK. Lastly, Section 5 concludes this paper.

2 Preliminaries

2.1 Notations

The used notations in this paper are shown in Table 1.

2.2 Rotational Cryptanalysis in the presence of Constants

In this section rotational-XOR cryptanalysis is recalled [2]. To specifically serve the purpose of this paper, a slightly different notation is used. A rotational-XOR-difference (or RX-difference in short) with rotational offset γ of two bit strings x and x' is defined as $\Delta_\gamma(x, x') = x \oplus (x' \lll \gamma)$. Another way to look at it is defining a string x and an RX-difference $\Delta_\gamma(x, x')$. Then x' can be computed as $x' = (x \oplus \Delta_\gamma(x, x')) \ggg \gamma$. Note that when $\gamma = 0$, the RX-difference $\Delta_0(x, x')$ is the difference that is studied in differential cryptanalysis and when $\Delta_\gamma(x, x') = 0$ the pair (x, x') is a rotational pair.

The following deterministic rules concerning the propagation of RX-differences through linear operations in ARX structures are easy to verify:

Notation	Meaning
$x = (x^{n-1}, \ldots, x^1, x^0)$	An n-bit boolean vector.
$\lvert x \rvert$	The Hamming weight of the boolean vector x.
$x \lll \gamma$	A cyclic left shift of x by γ bit(s).
$x \ggg \gamma$	A cyclic right shift of x by γ bit(s).
$SHL(x)$	An arithmetic left shift of x by one bit.
$(I \oplus SHL)(x)$	$x \oplus SHL(x)$
$L(x)$	The $n-1$ most significant bits of x: $(x^{n-1}, \ldots, x^2, x^1)$.
$x \lvert\rvert y$	The concatenation of the vectors x and y.
$x \lvert y$	The vector bitwise operation between the vectors x and y.
$1_{x \preceq y}$	Characteristic function which evaluates to 1 if $\forall i : x^i \leq y^i, 0 \leq i < n$, otherwise to 0.

Table 1: Notations used throughout this paper.

- ◇ for every ARX operation, the rotational offset γ remains constant.
- ◇ for a bit rotation with rotation amount α, the RX-difference is also rotated over the same amount α:

$$(x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \ggg \alpha = ((x \ggg \alpha) \oplus (\Delta_\gamma(x, x') \ggg \alpha)) \ggg \gamma. \quad (1)$$

- ◇ for every XOR operation with another RX-difference, both RX-differences are XORed together:

$$\begin{aligned} &((x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \oplus ((y \oplus \Delta_\gamma(y, y')) \ggg \gamma) \\ &= ((x \oplus y) \oplus (\Delta_\gamma(x, x') \oplus \Delta_\gamma(y, y')) \ggg \gamma). \end{aligned} \quad (2)$$

- ◇ for every XOR operation with a constant, the RX-difference is XORed with the constant and the constant rotated over an amount γ:

$$\begin{aligned} ((x \oplus \Delta_\gamma(x, x')) \ggg \gamma) \oplus c &= (x \oplus \Delta_\gamma(x, x') \oplus (c \lll \gamma)) \ggg \gamma \\ &= ((x \oplus c) \oplus \Delta_\gamma(x, x') \oplus (c \lll \gamma) \oplus c) \ggg \gamma. \end{aligned} \quad (3)$$

Note that for every RX-difference and every value of γ, the above rules hold, also when $\gamma = 0$ (differential cryptanalysis) and when $\Delta_\gamma(x, x') = 0$ (rotational cryptanalysis). RX-differential cryptanalysis can thus be seen as a generalization of two cryptanalytic techniques, rotational and differential cryptanalysis.

For the nonlinear operation modular addition, the propagation of RX-differences is not deterministic.

Theorem 1 (Propagation of RX-differences through modular addition [2])**.** *Suppose $x, y \in \mathbb{F}_{2^n}$ are independent uniform random variables, $x \boxplus y = z$ and $\Delta_1(x, x')$, $\Delta_1(y, y')$ and $\Delta_1(z, z')$ are constants in $\mathbb{F}_{2^n}$. Then,*

$$\begin{aligned} &Pr[((x \oplus \Delta_1(x, x')) \ggg 1) \boxplus ((y \oplus \Delta_1(y, y')) \ggg 1) = (z \oplus \Delta_1(z, z')) \ggg 1] \\ &\quad = 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \oplus 1 \preceq SHL((\delta_x \oplus \delta_z) \lvert (\delta_y \oplus \delta_z))} \cdot 2^{-\lvert SHL((\delta_x \oplus \delta_z) \lvert (\delta_y \oplus \delta_z)) \rvert} \cdot 2^{-3} \\ &\qquad + 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \preceq SHL((\delta_x \oplus \delta_z) \lvert (\delta_y \oplus \delta_z))} \cdot 2^{-\lvert SHL((\delta_x \oplus \delta_z) \lvert (\delta_y \oplus \delta_z)) \rvert} \cdot 2^{-1.415}, \end{aligned} \quad (4)$$

where

$$\delta_z = L(\Delta_1(x, x')),\ \delta_z = L(\Delta_1(y, y'))\ \textit{and}\ \delta_z = L(\Delta_1(z, z')).$$

The formula only considers RX-differences with $\gamma = 1$. If the constants $\Delta_1(x, x') = \Delta_1(y, y') = \Delta_1(z, z') = 0$, the theorem predicts the case for normal rotational cryptanalysis with rotation amount 1, $Pr[(x \ggg 1) \boxplus (y \ggg 1) = z \ggg 1] = 2^{-2.145}$.

2.3 Description of Speck

The Speck family of block ciphers is a collection of 10 block ciphers designed by the NSA in 2013 to fulfil the need for efficient, flexible, lightweight block ciphers with varying block length and key length. The parameters for every version of the Speck family are given in Table 2.

Block Size $2n$	Key Size mn	rot α	rot β	Number of rounds T
32	64	7	2	22
48	72	8	3	22
	96			23
64	96	8	3	26
	128			27
96	96	8	3	28
	144			29
128	128	8	3	32
	192			33
	256			34

Table 2: Parameters for different versions of Speck.

The output $(x_{i+1}||y_{i+1})$ of one round of encryption can be calculated from the input $(x_i||y_i)$ as follows:

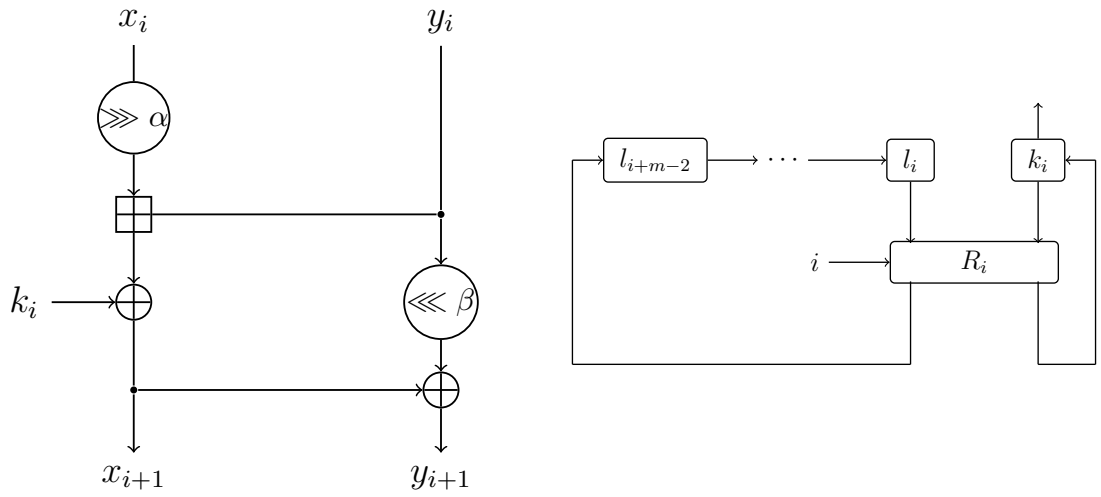
$$x_{i+1} = ((x_i \ggg \alpha) \boxplus y_i) \oplus k_i \tag{5}$$
$$y_{i+1} = (y_i \lll \beta) \oplus x_{i+1}. \tag{6}$$

$$l_{i+m-1} = (k_i \boxplus (l_i \ggg \alpha)) \oplus i \tag{7}$$
$$k_{i+1} = (k_i \lll \beta) \oplus l_{i+m-1}. \tag{8}$$

A visual representation of the Speck round function and the key schedule is given in Figure 1a and Figure 1b, respectively. For more information, we refer to [3].

3 Automated Search of RX-characteristics

A possible path of RX-differences through different encryption rounds of a block cipher is called an RX-characteristic. Finding highly probable RX-characteristics for a specific block cipher can be done manually, but is slow, of low efficiency and does not guarantee



(a) Round function of SPECK (b) Key Schedule of SPECK

Figure 1: Description of SPECK

to find the optimal solution. It is therefore natural to automate this process through computer programs.

Automated Search for differential characteristics has been done, usually by reformulating the problem of finding the highest probable characteristic as a Mixed-Integer Linear Program (MILP) [7] or as a Boolean Satisfiability problem (SAT) [5, 12]. In this paper, the SAT-approach is taken. A SAT-problem is the problem of determining whether a solution exists for a certain Boolean expression. A Boolean expression consists of variables, the operators AND ($\wedge$), OR ($\vee$), NOT ($\neg$) and parentheses. Because a bit can be seen as a Boolean variable, the propagation of RX-characteristics through block ciphers can often be translated as a SAT-problem.

In the next section the finding of good RX-characteristics for SPECK is formulated as a SAT-problem.

3.1 Application to Speck

The following theorem is a corollary from the rules of propagation of RX-characteristics through ARX primitives (Equations 1, 2, 3 and 4) and the round function of SPECK (Equations 5 and 6).

Theorem 2 (Constraints concerning the round function to find a valid r-round RX-characteristic for SPECK)**.** *To obtain a valid r-round RX-characteristic for* SPECK*, the boolean bits of RX-differences in round i, with a notation used as in Figure 2a, need to satisfy the following constraints:*

$$(I \oplus SHL)(\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 b_i^j \oplus \Delta_1 d_i^j) \oplus 1 \preceq SHL((\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 b_i^j \oplus \Delta_1 d_i^j))$$

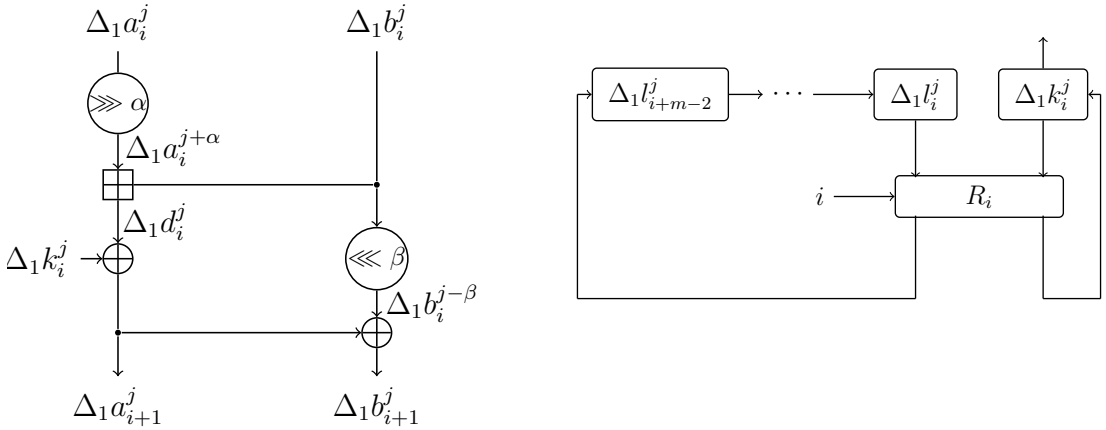
$$OR$$

$$(I \oplus SHL)(\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 b_i^j \oplus \Delta_1 d_i^j) \preceq SHL((\Delta_1 a_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 b_i^j \oplus \Delta_1 d_i^j)),$$

$$\Delta_1 a_{i+1}^j = \Delta_1 d_i^j \oplus \Delta_1 k_i,$$

$$\Delta_1 b_{i+1}^j = \Delta_1 b_i^{j-\beta} \oplus \Delta_1 a_{i+1}^j,$$

where $0 \leq i \leq r$ and $0 \leq j < n$.



(a) Notation of RX-differences in the round function of SPECK.

(b) Notation of RX-differences in the key schedule of SPECK.

Figure 2: Description of SPECK

Theorem 3 (Constraints concerning the key schedule to find a valid r-round RX-characteristic for SPECK)**.** *To obtain a valid r-round RX-characteristic for* SPECK*, the boolean bits of RX-differences in round i of the key schedule, with a notation used as in Figure 2b, need to satisfy the following constraints:*

$$(I \oplus SHL)(\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 k_i^j \oplus \Delta_1 d_i^j) \oplus 1 \preceq SHL((\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 k_i^j \oplus \Delta_1 d_i^j))$$

$$OR$$

$$(I \oplus SHL)(\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 k_i^j \oplus \Delta_1 d_i^j) \preceq SHL((\Delta_1 l_i^{j+\alpha} \oplus \Delta_1 d_i^j)|(\Delta_1 k_i^j \oplus \Delta_1 d_i^j)),$$
$$\Delta_1 l_{i+1}^j = \Delta_1 d_i^j \oplus (c \lll 1) \oplus c,$$
$$\Delta_1 k_{i+1}^j = \Delta_1 k_i^{j-\beta} \oplus \Delta_1 l_{i+1}^j,$$

where $0 \leq i \leq r$ and $0 \leq j < n$.

Because $p \preceq q$, with p and q boolean expressions, is equivalent with $p = p \wedge q$ and $p \oplus q$ equivalent with $(p \vee q) \wedge (\bar{p} \vee \bar{q})$, the above constraints are easily expressed as a Boolean expression.

The above two theorems allow to find valid RX-characteristics with a SAT-solver, the following theorem allows to calculate the data probability of an RX-characteristic, which follows from Equation 4.

$$w_i = \begin{cases} |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))|+3 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 1 \\ |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))|+1.415 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 0 \end{cases}$$

Theorem 4 (Data probability of a valid r-round RX-characteristic for SPECK)**.** *The probability of the a valid RX-characteristic is given by*

$$2^{-\sum_{i=1}^{r} w_i}, \tag{9}$$

where

Version	Number of rounds	Data Probability	Key Probability	Reference
32/64	10	2^{-30}	1	[6]
32/64	10	$2^{-19.15}$	$2^{-35.905}$	This paper
32/64	11	$2^{-22.065}$	$2^{-45.32}$	This paper
32/64	12	$2^{-25.565}$	$2^{-59.075}$	This paper
48/96	11	2^{-40}	1	[6]
48/96	11	$2^{-23.15}$	$2^{-81.075}$	This paper
48/96	12	$2^{-26.565}$	$2^{-52.49}$	This paper
48/96	13	$2^{-31.98}$	$2^{-71.49}$	This paper
96/144	13	$2^{-84.00}$	1	[6]
96/144	13	$2^{-37.98}$	$2^{-106.075}$	This paper

Table 3: Comparison of RX-characteristics with $\gamma = 1$ and previous differentials for different versions of Speck.

$$w_i = \begin{cases} |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))|+3 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 1 \\ |SHL((\Delta_1 a_{i+7}^r \oplus \Delta_1 a_i^{r+1})|(\Delta_1 b_i^r \oplus \Delta_1 a_i^{r+1}))|+1.415 & \Delta_1 a_i^\alpha \oplus \Delta_1 b_i^0 \oplus \Delta_1 d_i^0 = 0 \end{cases}$$

if the input of each modular addition is an independent uniform random variable.

A similar equation for key probability (i.e. the probability of the RX-characteristic through the key schedule) can be derived. This equation has a slightly different interpretation, it expresses the probability that two independent uniform random keys are related, in order to be able to observe the RX-characteristic. The results in Table 3 are found by first minimizing the data probability and then minimizing the key probability. Another trade-off between data probability and key probability is also possible.

In this paper, we develop a method to search for nearly optimal RX-characteristics with $\gamma = 1$. The problem of finding optimal RX-characteristics with $\gamma = 1$ is reformulated as a boolean satisfiability problem and can subsequently be solved by a SAT-solver. The automatic search is applied to different Speck versions and distinguishers with a higher probability than previously reported are found for each of these versions, provided that two related keys are available.

Future research can include Rotational-XOR cryptanalysis where γ differs from 0 and 1, as already mentioned in [2], applying this method to the other versions of Speck and other ARX primitives or obtaining longer RX-characteristics by concatenating two shorter RX-characteristics.

Acknowledgements. This work was supported in part by the Research Council KU Leuven OT/13/071, by the Flemish Government through FWO Thresholds G0842.13 and by European Unions Horizon 2020 research and innovation programme under grant agreement No. H2020-MSCA-ITN-2014-643161 ECRYPT-NET.

References

[1] T. Ashur and D. Bodden. Linear cryptanalysis of reduced-round speck. 2016.

[2] T. Ashur and Y. Liu. Rotational cryptanalysis in the presence of constants. *IACR Trans. Symmetric Cryptol.*, 2016(1):57–70, 2016.

[3] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*, pages 175:1–175:6. ACM, 2015.

[4] D. J. Bernstein. The salsa20 family of stream ciphers. In M. J. B. Robshaw and O. Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.

[5] A. Biryukov, V. Velichkov, and Y. L. Corre. Automatic search for the best trails in ARX: application to block cipher speck. In Peyrin [11], pages 289–310.

[6] I. Dinur. Improved differential cryptanalysis of round-reduced SPECK. In A. Joux and A. M. Youssef, editors, *Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers*, volume 8781 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2014.

[7] K. Fu, M. Wang, Y. Guo, S. Sun, and L. Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In Peyrin [11], pages 268–288.

[8] D. Khovratovich and I. Nikolic. Rotational cryptanalysis of ARX. In S. Hong and T. Iwata, editors, *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, 2010.

[9] D. Khovratovich, I. Nikolic, J. Pieprzyk, P. Sokolowski, and R. Steinfeld. Rotational cryptanalysis of ARX revisited. In G. Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 519–536. Springer, 2015.

[10] Y. Liu, Q. Wang, and V. Rijmen. Automatic search of linear trails in ARX with applications to SPECK and chaskey. In M. Manulis, A. Sadeghi, and S. Schneider, editors, *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, volume 9696 of *Lecture Notes in Computer Science*, pages 485–499. Springer, 2016.

[11] T. Peyrin, editor. *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*. Springer, 2016.

[12] L. Song, Z. Huang, and Q. Yang. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In J. K. Liu and R. Steinfeld, editors, *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*, volume 9723 of *Lecture Notes in Computer Science*, pages 379–394. Springer, 2016.

[13] D. J. Wheeler and R. M. Needham. Tea, a tiny encryption algorithm. In B. Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 363–366. Springer, 1994.

Appendix B

Poster presented on the "masterproefbeurs"

Below is the poster presented on the "masterproefbeurs" on 9 May 2017. This was an event on which the students of the "Master of Mathematical Engineering" presented a poster about their master thesis.

An Automated Tool for Rotational-XOR Cryptanalysis of SPECK

KATHOLIEKE UNIVERSITEIT LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Master Wiskundige ingenieurstechnieken

Masterproef
Glenn De Witte

Promotor
Vincent Rijmen

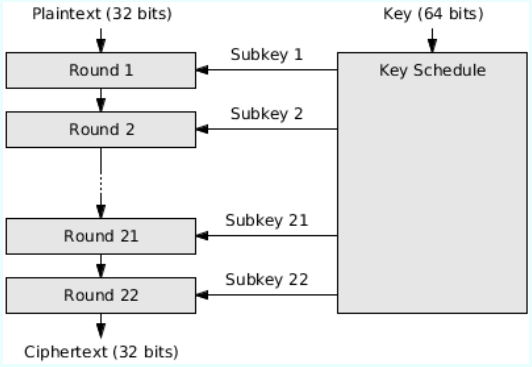
Begeleiders
Tomer Ashur
Yunwen Liu

Academiejaar 2016-2017

Cryptanalysis of Block Ciphers

Statistical Cryptanalysis Find statistical relations between the plaintext, ciphertext, and the key.

Iterated Block Ciphers Modern block ciphers encrypt via multiple, equivalent rounds, each with a different subkey.



Rotational-XOR Cryptanalysis

RX-difference (T. Ashur and Y. Liu)

$$\Delta_\gamma(x, x') = x \oplus (x' \lll \gamma)$$

Deterministic propagation through rotation and XOR.

Probabilistic propagation through modular addition.

Formula to determine the probability of the propagation through modular addition (T. Ashur and Y. Liu)

$$\begin{aligned} Pr[((x \oplus \Delta_1(x,x')) \ggg 1) \boxplus ((y \oplus \Delta_1(y,y')) \ggg 1) = (z \oplus \Delta_1(z,z')) \ggg 1] \\ = 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \oplus 1 \preceq SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))} \cdot 2^{|SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))|} \cdot 2^{-3} \\ + 1_{(I \oplus SHL)(\delta_x \oplus \delta_y \oplus \delta_z) \preceq SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))} \cdot 2^{|SHL((\delta_x \oplus \delta_z)|(\delta_y \oplus \delta_z))|} \cdot 2^{-1.415}, \end{aligned}$$

where

$$\delta_z = L(\Delta_1(x,x')),\ \delta_z = L(\Delta_1(y,y'))\ \textit{and}\ \delta_z = L(\Delta_1(z,z')).$$

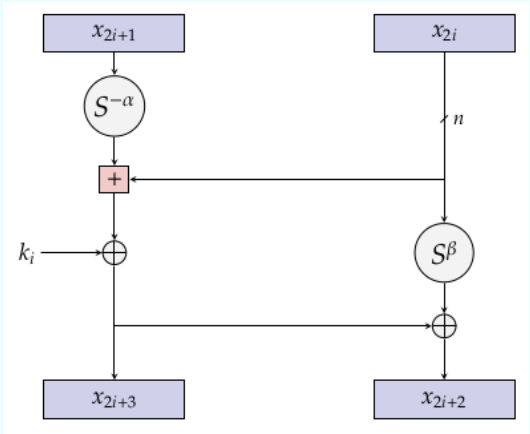
Results

- RX-characteristics for more rounds are found for SPECK32/64 and SPECK48/96.
- RX-characteristics with lower data requirements are found for SPECK96/144.

Under the condition that two related keys are available.

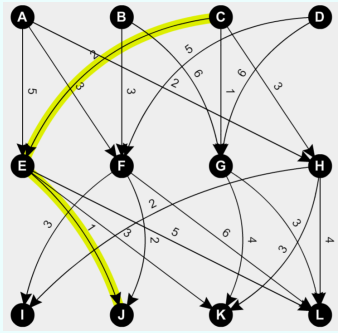
Version	Number of rounds	Data Probability	Key Probability	Reference
32/64	10	2^{-30}	1	Other paper
32/64	10	$2^{-19.15}$	$2^{-35.905}$	This thesis
32/64	11	$2^{-22.065}$	$2^{-45.32}$	This thesis
32/64	12	$2^{-25.565}$	$2^{-59.075}$	This thesis
48/96	11	2^{-40}	1	Other paper
48/96	11	$2^{-23.15}$	$2^{-81.075}$	This thesis
48/96	12	$2^{-26.565}$	$2^{-52.49}$	This thesis
48/96	13	$2^{-31.98}$	$2^{-71.49}$	This thesis
96/144	13	$2^{-84.00}$	1	Other paper
96/144	13	$2^{-37.98}$	$2^{-106.075}$	This thesis

SPECK Block Cipher Family



Automatic Search Tool

- Propagation of RX-differences through multiple rounds of SPECK can be reformulated as a SAT problem.
- The maximal probability is found by checking the satisfiability of the problem for different probabilities.



0	1	2	3	4	5	6
unsat	unsat	unsat	sat	sat	sat	sat

Images taken from URL "https://www.cs.rit.edu/~ark/662/p1/p1.shtml" (upper left) and the paper "The SIMON and SPECK families of Block Ciphers" (lower left).

Bibliography

[1] A. Arjunan, P. Narayanan, and K. Ramu. Securing RSA algorithm against timing attack. *Int. Arab J. Inf. Technol.*, 13(4):471–476, 2016.

[2] T. Ashur and Y. Liu. Rotational cryptanalysis in the presence of constants. *IACR Trans. Symmetric Cryptol.*, 2016(1):57–70, 2016.

[3] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*, pages 175:1–175:6. ACM, 2015.

[4] E. Biham and A. Shamir. Differential cryptanalysis of des-like cryptosystems. *J. Cryptology*, 4(1):3–72, 1991.

[5] D. Coppersmith. The data encryption standard (DES) and its strength against attacks. *IBM Journal of Research and Development*, 38(3):243–250, 1994.

[6] I. Dinur. Improved differential cryptanalysis of round-reduced SPECK. In A. Joux and A. M. Youssef, editors, *Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers*, volume 8781 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2014.

[7] K. Fu, M. Wang, Y. Guo, S. Sun, and L. Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In T. Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 268–288. Springer, 2016.

[8] M. Kar, A. Singh, S. Mathew, A. Rajan, V. De, and S. Mukhopadhyay. 8.1 improved power-side-channel-attack resistance of an AES-128 core via a security-aware integrated buck voltage regulator. In *2017 IEEE International Solid-State Circuits Conference, ISSCC 2017, San Francisco, CA, USA, February 5-9, 2017*, pages 142–143. IEEE, 2017.

[9] D. Khovratovich and I. Nikolic. Rotational cryptanalysis of ARX. In S. Hong and T. Iwata, editors, *Fast Software Encryption, 17th International Workshop,*

FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, 2010.

[10] Y. Liu, Q. Wang, and V. Rijmen. Automatic search of linear trails in ARX with applications to SPECK and chaskey. In M. Manulis, A. Sadeghi, and S. Schneider, editors, *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, volume 9696 of *Lecture Notes in Computer Science*, pages 485–499. Springer, 2016.

[11] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[12] O. Ohrimenko, P. J. Stuckey, and M. Codish. Propagation = lazy clause generation. In C. Bessiere, editor, *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007, Providence, RI, USA, September 23-27, 2007, Proceedings*, volume 4741 of *Lecture Notes in Computer Science*, pages 544–558. Springer, 2007.

[13] C. E. Shannon. Communication theory of secrecy systems. 28(4):656–715, Oct. 1949. A footnote on the initial page says: "The material in this paper appeared in a confidential report, 'A Mathematical Theory of Cryptography', dated Sept. 1, 1945 ([**?**]), which has now been declassified.".

[14] L. Song, Z. Huang, and Q. Yang. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In J. K. Liu and R. Steinfeld, editors, *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*, volume 9723 of *Lecture Notes in Computer Science*, pages 379–394. Springer, 2016.

Fiche masterproef

Student: Glenn De Witte

Titel: Automatic SAT-solver Based Search Tools for Cryptanalysis

Nederlandse titel: Automatische SAT-solver gebaseerde zoekprogramma's voor cryptanalyse

UDC: 621.3

Korte inhoud:

In ARX structures, constants that are not rotational invariant are often injected into the state, in the form of round constants or as a result of using a fixed key. Rotational cryptanalysis cannot deal with such constants. Rotational cryptanalysis in the presence of constants, also known as rotational-XOR cryptanalysis, is a recently proposed statistical technique to attack ARX primitives. The newly proposed technique investigates how constants affect rotational cryptanalysis by introducing the notion of an RX-difference, which generalizes the idea of a rotational difference. Previously, a 7-round distinguisher for Speck, an ARX block cipher designed by the NSA, was found, mainly to demonstrate the proposed technique. In this paper, it is shown that longer distinguishers exist for Speck32/64, as well as other versions of Speck, by using rotational-XOR cryptanalysis. This was done by means of an automated search tool. More specifically, the propagation of RX-differences through ARX structures are transformed into bitwise equations and these equations are then solved by a SAT-solver.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: wiskundige ingenieurstechnieken

Promotor: Prof. dr. ir. Vincent Rijmen

Assessoren: Prof. dr. ir. J. Suykens
Prof. dr. ir. M. Van Barel

Begeleiders: T. Ashur
Y. Liu