DIFFERENTIAL AND LINEAR CRYPTANALYSIS OF LIGHTWEIGHT BLOCK CIPHERS WITH MILP APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT BURHAN İLTER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

SEPTEMBER 2023

Approval of the thesis:

DIFFERENTIAL AND LINEAR CRYPTANALYSIS OF LIGHTWEIGHT BLOCK CIPHERS WITH MILP APPROACH

submitted by **MURAT BURHAN İLTER** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Cryptography Department, Middle East Technical University** by,

Prof. Dr. A. Sevtap Selçuk Kestel
Dean, Graduate School of **Applied Mathematics** _______________

Assoc. Prof. Dr. Oğuz Yayla
Head of Department, **Cryptography** _______________

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Mathematics, METU** _______________

Prof. Dr. Ali Aydın Selçuk
Co-supervisor, **Dept. of Computer Engineering, TOBB ETU** _______________

Examining Committee Members:

Prof. Dr. Zülfükar Saygı
Mathematics, TOBB ETU _______________

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics, METU _______________

Assoc. Prof. Dr. Oğuz Yayla
Crpytography, METU _______________

Assoc. Prof. Dr. Fatih Sulak
Mathematics, ATILIM University _______________

Assoc. Prof. Dr. Cihangir Tezcan
Cyber Security, METU _______________

Date: _______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MURAT BURHAN İLTER

Signature :

ABSTRACT

DIFFERENTIAL AND LINEAR CRYPTANALYSIS OF LIGHTWEIGHT BLOCK CIPHERS WITH MILP APPROACH

İLTER, MURAT BURHAN
Ph.D., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy
Co-Supervisor : Prof. Dr. Ali Aydın Selçuk

September 2023, 54 pages

The security of block ciphers can be evaluated using cryptanalysis methods. The use of Mixed-Integer Linear Programming (MILP) has gained prominence due to its effectiveness in analyzing the security aspects of block ciphers. In this thesis, we explore the application of MILP techniques for conducting comprehensive differential and linear cryptanalysis. Our research specifically addresses fundamental challenges in the realm of differential and linear cryptanalysis.

In this work, we study the cipher resistance against differential and linear attacks taking into account that ciphers need to be resistant to these attacks. In this context, aiming to identify the best differential and linear characteristics of a block cipher is a challenging problem. To tackle these challenges, our work introduces innovative MILP modeling methods for equations involving multiple xor operations. These models, denoted as Model 1 and Model 2, offer alternatives with fewer variables and constraints, respectively. Model 1 and Model 2 generally provide shorter solution times compared to the standard xor model. Importantly, these proposed models have broad applicability beyond differential and linear cryptanalysis, enhancing their utility in various cryptanalysis methods.

We model well-known ciphers such as KLEIN, PRINCE, FUTURE, and IVLBC with MILP. The resulting models enable us to precisely determine the exact minimum number of active S-boxes, and the best differential and linear characteristics. Applying our developed MILP models provides improvements in the best single-key differential and linear characteristics for the examined ciphers.

Keywords: Block Ciphers, Mixed-Integer Linear Programming (MILP), Differential Cryptanalysis, Linear Cryptanalysis

ÖZ

KAYNAK KISITLI BLOK ŞİFRELERİN KTLP YAKLAŞIMI İLE
DİFERANSİYEL VE LİNEER KRİPTANALİZİ

İLTER, MURAT BURHAN
Doktora, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy
Ortak Tez Yöneticisi : Prof. Dr. Ali Aydın Selçuk

Eylül 2023, 54 sayfa

Blok şifrelerin güvenliği kriptanaliz yöntemleri kullanılarak analiz edilebilir. Karma Tamsayılı lineer Programlamanın (KTLP) kullanımı, blok şifrelerin güvenlik yönlerini analiz etmede oldukça etkin olması nedeniyle önem kazanmıştır. Bu tezde, kapsamlı diferansiyel ve lineer kriptanaliz yöntemleri için KTLP tekniklerinin uygulanması araştırılmaktadır. Bu çalışma, özellikle diferansiyel ve lineer kriptanaliz alanındaki temel zorlukları ele almaktadır.

Bu çalışmada şifrelerin diferansiyel ve lineer saldırılara karşı dayanıklı olması gerektiği dikkate alınarak bu saldırılara karşı şifre dirençleri incelenmektedir. Bu bağlamda bir blok şifrenin en iyi diferansiyel ve lineer karakteristiklerini bulmayı hedeflemek zor bir problemdir. Çalışmamızda bu problemin çözümüne yönelik olarak çoklu xor işlemlerini içeren denklemler için yenilikçi KTLP modelleme yöntemleri sunulmaktadır. Model 1 ve Model 2 olarak adlandırılan bu modeller sırasıyla daha az değişken ve kısıtla alternatifler sunmaktadır. Model 1 ve Model 2 genellikle standart xor modeline göre daha kısa çözüm süreleri sağlar. Önerilen bu modeller, diferansiyel ve lineer kriptanalizin ötesinde geniş bir uygulanabilirliğe sahiptir ve çeşitli kriptanaliz yöntemlerindeki verimliliği artırır.

Bu tezde, KTLP ile KLEIN, PRINCE, FUTURE ve IVLBC gibi iyi bilinen şifreler modellenmektedir. Sunulan modeller, kesin minimum diferansiyel aktif S-kutularının sayısının ve en iyi diferansiyel ve lineer karakteristiklerin belirlenmesini sağlamaktadır. Geliştirilen KTLP modelleri, incelenen şifreler için literatürde yer alan en iyi tek anahtarlı diferansiyel ve lineer karakteristiklerde iyileştirmeler ortaya koymaktadır.

Anahtar Kelimeler: Blok Şifre, Karmaşık Tamsayılı Lineer Programlama, Diferansiyel Kriptanaliz, Lineer Kriptanaliz

To my family

ACKNOWLEDGMENTS

I am pleased to thank my advisor, Assoc. Prof. Dr. Ali Doğanaksoy, for his invaluable insights and guidance throughout my entire graduate studies. I am deeply indebted to my co-advisor, Prof. Dr. Ali Aydın Selçuk, for his insightful feedback, thoughtful suggestions, and continuous support. I am profoundly grateful for the immeasurable contributions he made to my development. I would also like to express my sincere gratitude to the Ph.D. thesis committee members for providing valuable feedback.

I am grateful to Siwei Sun and Yu Sasaki for their helpful advice.

I sincerely appreciate my friends and colleagues who have been a constant source of inspiration and motivation. Special thanks to Tayfun Doğrar, Dr. Hakan Genç, Sura Imren, Dr. Cemre Bozyiğit, Melis Yünüak, Erkan Uslu and Dobi.

I am thankful to ASELSAN Inc. for supporting this thesis.

I would like to express my appreciation to Anıl Aslan, who constantly supported my work with hardware equipment, and to Berksu Kısmet for his expertise in Python.

My deepest thanks belong to Dr. Neşe Koçak and Dr. Onur Koçak who provided exceptional support through my research.

A very special appreciation goes to my friend H. Bartu Yünüak to whom I am truly thankful for his invaluable knowledge and genuine guidance.

I would like to thank my parents Altan İlter, Reha İlter, and my brother Umut İlter for always encouraging and motivating me. I want to extend my heartfelt gratitude to my wife, Cansu Aktepe İlter, whose constant support and love sustained me during the challenging years of my Ph.D. Your presence and encouragement made the challenges bearable and the successes sweeter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
DDT	Difference Distribution Table
DES	Data Encryption Standard
LAT	Linear Approximation Table
LP	Linear Programming
MDS	Maximum Distance Separable
MILP	Mixed Integer Linear Programming
SPN	Substitution Permutation Network

CHAPTER 1

INTRODUCTION

Cryptanalysis of block ciphers has been studied in the literature for many years. Block ciphers are designed to provide security requirements for different platforms. For instance, these ciphers are used in the Internet of Things, RFID, Smart Cards, and sensor technologies. Security requirements differ across these platforms, and in some cases, using AES [8] is not suitable due to resource limitations such as area, energy, and code size. Consequently, comprehensive cryptanalysis of these ciphers is a mandatory requirement.

Differential [3] and linear [23] cryptanalysis are two cornerstone methods in the field. Ciphers must be resistant to these attacks. Due to their time-consuming nature, demonstrating resistance against all cryptanalysis methods using manual approaches is exceedingly challenging. Modifying the building blocks of the cipher can significantly change how the system behaves and will require cryptanalysis to be done from the beginning. For instance, in differential and linear cryptanalysis, changing the S-box or permutation requires a complete restart of the analysis. These demands have prompted the utilization of semi-automatic or automatic tools in cryptanalysis to provide resilience against these attack methods. Automated tools play a pivotal role in cryptanalysis. In recent literature, newly designed ciphers are increasingly being analyzed using automated tools. Among these, the Mixed-Integer Linear Programming (MILP) method has gained importance as a powerful tool for analyzing the security of block ciphers. Prior to the emergence of MILP applications, cipher-specific automated search algorithms were common, but often challenging to implement. In contrast, MILP methods are easier to implement and more effective, establishing MILP as a crucial tool for cipher analysis and attacks.

In this thesis, our aim is to model matrix multiplication in the MILP models in an efficient way. Matrix multiplication can be demonstrated as multiple xor operations with the primitive representation of a given finite field. We effectively model lightweight block ciphers using the MILP approach in order to determine the exact minimum number of differentially active S-boxes and the best single-key differential and linear characteristics.

1.1 Literature Survey

Block ciphers, stream ciphers, and hash functions have been analyzed using MILP models. Mouha et al. [24] suggested a method to find the minimum number of active S-boxes for word-oriented ciphers using the MILP approach. They analyzed the minimum number of active S-boxes for linear and differential cryptanalysis of the AES and Enocoro ciphers.

Following Mouha et al.'s work, much research in cryptanalysis has been done using MILP. Various cryptanalysis methods such as differential [37], linear [10], impossible differential [27], and conditional cube attacks [22] have been also modeled by this approach.

Sun et al. [30] proposed a MILP model to find the minimum number of active S-boxes for bit-oriented block ciphers. In that work, PRESENT-80 was modeled with MILP for single-key and related-key differential cryptanalysis.

Sun et al. [32] gave the first analysis using the H-representation and logical condition modeling to give an exact representation of an S-box with a greedy algorithm to model S-boxes. The authors analyzed the ciphers SIMON, Serpent, LBlock, and DESL. They obtained significant results of differential cryptanalysis and related key attacks on these ciphers.

Sun et al. [31] recommended a method to find the best characteristic. In this work, the probability information of possible differential patterns was added to the S-box representation. The authors studied the SIMON48, LBlock, DESL, and PRESENT-128 ciphers and obtained improved results on differential cryptanalysis, linear cryptanalysis, and related key attacks on these ciphers.

Different types of ciphers, besides bit-oriented, lightweight ciphers, have been also analyzed by MILP: Sun et al. [28] applied the technique to analyze ARX-based ciphers. Abdelkhalek et al. [1] and Boura and Coggia [7] modeled ciphers with 8×8 S-boxes by MILP.

The solution performance of a MILP model is highly dependent on the complexity of the model and the number of constraints and variables involved [21]. More efficient models need to be constructed in order to analyze a higher number of rounds of a given cipher. This has been the focus of many MILP-based studies in the literature.

For instance, Sasaki and Todo [26] developed a novel method to represent an S-box with fewer constraints; Fu et al. [10] presented a methodology wherein a single constraint is utilized to model xor operations; and Yin et al. [36] modeled xor operations with fewer variables.

In a block cipher that uses (MDS) matrix multiplication operations over $GF(2^n)$ for diffusion, such as AES, the multiplication of a vector by the matrix can be expressed in a set of xor operations. Sun et al. [29] showed how to model differential propagation over an MDS matrix multiplication by MILP. In the MILP modeling of such ciphers, the performance of the resulting MILP model can be significantly improved by reducing the complexity of the combined xor operations within the model.

1.2 Our Contributions

Our work proposes two novel methods in order to model multiple xor operations in the MILP approach. These models are called Model 1 and Model 2. In the thesis, in addition to these proposed models, the standard xor model is also implemented in order to make a comparison of the solution times of the MILP models.

We model lightweight block ciphers KLEIN, PRINCE, FUTURE, and IVLBC with three alternative ways, namely Model 1, Model 2, and standard xor models via the MILP approach. Our main contributions are listed as follows:

- We obtain the exact minimum number of differentially active S-boxes for KLEIN and PRINCE.
- We achieve the best differential and linear characteristics of KLEIN, PRINCE, FUTURE, and IVLBC in the literature.
- The solution times for the aforementioned MILP models are compared. Our proposed methods, namely Model 1 and Model 2, generally provide shorter solution times compared to the standard xor model.

The structure of the remainder of this thesis is as follows: Chapter 2 provides preliminaries, introducing differential and linear cryptanalysis and linear and mixed integer linear programming. Chapter 3 provides details on the construction of the differential and linear MILP models and introduces the proposed xor models. Chapter 4 presents the KLEIN cipher along with MILP models for determining the exact minimum number of differentially active S-boxes, as well as the best differential and linear characteristics. In Chapter 5, we analyze the PRINCE cipher using the MILP approach, presenting MILP formulations to determine the exact minimum number of differentially active S-boxes, along with the best differential and linear characteristics. Chapter 6 presents and models FUTURE using MILP to obtain the optimal differential and linear characteristics. Chapter 7 describes the MILP modeling of IVLBC, providing algorithmic details and the best differential and linear characteristics. Finally, we conclude the thesis in Chapter 8.

CHAPTER 2

PRELIMINARIES

In this chapter milestone cryptanalysis methods differential and linear cryptanalysis are briefly introduced. Linear Programming and Mixed-Integer Linear Programming are presented.

2.1 Differential Cryptanalysis

Biham and Shamir [3] pioneered the concept of differential cryptanalysis, a chosen-plaintext attack, which they employed in their analysis of the Data Encryption Standard (DES).

The fundamental premise of this attack is to establish a correlation between pairs of plaintext and ciphertext. Let P and P' denote two plaintexts encrypted as C and C' under the same key. The correlation between $(\Delta P, \Delta C)$, where $\Delta P = P \oplus P'$ and $\Delta C = C \oplus C'$, is computed for an r-round cipher, with the assumption that each round operates independently from each other.

In differential cryptanalysis, only nonlinear components, such as the S-box, influence the probability information of each round. Let Δx and Δy represent the input and output differences of an S-box. The probabilities of these differences are calculated using a Differential Distribution Table (DDT), in which we store the number of occurrences of each output per input $\Delta x, \Delta y$.

For the linear layer, positions of differences change with respect to permutation. Let Δy and Δz represent the input and output differences of the linear layer. In a round characteristic, the probabilities are obtained in the S-layer, and the positions are determined via the P-layer. Therefore, a complete round characteristic is represented as follows:

$$\Delta x \xrightarrow{S-layer} \Delta y \xrightarrow{P-layer} \Delta z$$

In order to construct complete characteristics, appropriate input and output differences are connected; in other words, the output differences from round $n-1$ are equal to the input differences of round n. Higher probabilities of round characteristics are selected to build complete characteristics, denoted as $(\Delta P, \Delta C)$.

Differential cryptanalysis marked a significant milestone in the field of cryptanalysis, prompting the development of ciphers designed to resist such attacks. Various variants have been proposed, including higher-order [20], truncated [18], impossible [6], and improbable [33] differential cryptanalysis.

2.2 Linear Cryptanalysis

Linear cryptanalysis, initially introduced by Matsui[23], constitutes a known-plaintext attack technique extensively applied in the analysis of cryptographic systems, including the Data Encryption Standard (DES).

The fundamental objective of linear cryptanalysis is to discover effective linear approximations between plaintext (denoted as P), ciphertext (C), and the encryption key (K). These approximations rely on the probability (or bias) of the linear relationship to recover the encryption key.

In the context of this technique, the relationship between plaintext, ciphertext, and the encryption key can be expressed as follows:

$$P[i_0, i_1, \cdots, i_a] \oplus C[j_0, j_1, \cdots, i_b] = K[k_0, k_1, \cdots, k_c]$$

Here, the variables i, j, and k represent specific bit positions, often referred to as input and output masks. To employ these linear approximations effectively in linear cryptanalysis, it is crucial for the probability to deviate from the expected value of $1/2$. Matsui introduced two algorithms to leverage these linear approximations, analogous to the analysis of each encryption round in differential cryptanalysis.

Similar to differential cryptanalysis, linear cryptanalysis independently analyzes each encryption round. Round characteristics are employed to construct complete characteristics, facilitating the cryptanalysis process.

Moreover, the investigation of the properties of S-boxes can reveal effective linear approximations. Linear Approximation Tables (LAT) of S-boxes offer valuable probability information regarding input and output masks that is calculated the number of occurrences of each output per input. In a round characteristic, probabilities are calculated within the S-layer, while positions are determined by the P-layer. Complete characteristics are obtained by linking input and output differences, similar to the approach used in differential cryptanalysis.

Linear cryptanalysis stands as a pivotal milestone in the realm of cryptography, enabling the analysis and evaluation of cryptographic systems. Variants of linear cryptanalysis have since emerged, including multiple approximations [17], multidimensional [13], and zero-correlation [4] linear cryptanalysis, which extend and refine the technique's capabilities.

2.3 Linear Programming

Linear Programming is a mathematical optimization technique that was discovered in the 1950s[9]. This versatile technique finds applications in various fields, including economics, manufacturing, mathematics, and engineering. Its primary objective is to derive solutions to objective functions while adhering to linear constraints.

The mathematical formalization of linear programming in standard form is provided in which x_i's are decision variables as follows [35]:

$$\begin{aligned} \text{Maximize (or Minimize)} \quad & c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{Subject to:} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \\ & \ldots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \\ & x_1, x_2, \ldots, x_n \geq 0. \end{aligned}$$

This representation features m constraints and n decision variables. In Linear Programming (LP), decision variables take real-number values. Certain efficient methods, such as interior point algorithms [25], exist to solve LP instances in polynomial time.

Mixed-Integer Linear Programming (MILP) constitutes a subfield of LP in which some decision variables are constrained to integer values. It is worth noting that MILP problems are generally classified as NP-hard, implying that there is no known polynomial-time algorithm for solving them. Nevertheless, specialized algorithms, including branch and bound, branch and cut, and branch and price [19], have been developed to address MILP challenges.

CHAPTER 3

MILP MODELLING OF DIFFERENTIAL AND LINEAR CRYPTANALYSIS

In a block cipher that uses matrix multiplication operations over $GF(2^n)$ for diffusion the multiplication of a vector by the matrix can be expressed in a set of xor operations. In the MILP modeling of such ciphers, the performance of the resulting MILP model can be significantly improved by reducing the complexity of the combined XOR operations within the model.

In this chapter, we proposed two novel xor models and implemented standard xor model to compare solution times of the MILP models. In the next chapters, these models are utilized to model the matrix multiplication operations over $GF(2^n)$. We investigate the construction of the MILP model for two purposes: finding the exact minimum number of differentially active S-boxes and identifying the best differential and linear characteristics using the MILP approach.

3.1 XOR models

In this study, we use the term "n-xor" to denote the xor operation involving $n + 1$ binary variables. As an example, $y = x_1 \oplus x_2 \oplus x_3$ is a 2-xor operation.

We investigate three models, namely the standard xor model, Model 1, and Model 2 to model the multiple xor operations that are used to represent the matrix multiplication in the analyzed block ciphers.

3.1.1 Standard Xor Model

In the standard xor model, multiple xors are divided into 1-xors that are modeled separately. The 1-xor operation $y = x_1 \oplus x_2$, where $y, x_1, x_2 \in \mathbb{F}_2$, is modeled with three variables and four constraints [26]:

$$\begin{aligned} x_1 - x_2 - y &\leq 0 & \qquad -x_1 + x_2 - y &\leq 0 \\ -x_1 - x_2 + y &\leq 0 & x_1 + x_2 + y &\leq 2 \end{aligned}$$

We can model the 2-xor operation $y = x_1 \oplus x_2 \oplus x_3$ from two separate 1-xor operations as, $d_1 = x_1 \oplus x_2$ and $y = d_1 \oplus x_3$ with five variables and eight constraints:

$$\begin{aligned} x_1 - x_2 - d_1 &\leq 0 & \qquad d_1 - x_3 - y &\leq 0 \\ -x_1 - x_2 + d_1 &\leq 0 & -d_1 + x_3 - y &\leq 0 \\ -x_1 + x_2 - d_1 &\leq 0 & -d_1 - x_3 + y &\leq 0 \\ x_1 + x_2 + d_1 &\leq 2 & d_1 + x_3 + y &\leq 2 \end{aligned}$$

where $d_1 \in \{0, 1\}$ is a dummy variable.

3.1.2 Model 1

In our method, we first calculate possible patterns for multiple xor operations. We then use Sasaki and Todo's approach [26] to represent these patterns with the minimum number of constraints.

An H-representation is a representation of a polyhedron that contains a set of given valid points. The H-representation of these patterns contains redundant inequalities, but with this approach, we can represent multiple xor operations with the minimum number of constraints. As an example, the 2-xor operation is calculated as follows:

Let $y = x_1 \oplus x_2 \oplus x_3$ in which $y, x_1, x_2, x_3 \in \mathbb{F}_2$. There are 8 possible xor results (valid points) after calculating H-representation, we obtain 16 inequalities. By applying

Sasaki and Todo's technique, we derive the following 8 inequalities:

$$-x_1 - x_2 + x_3 - y \leq 0 \qquad -x_1 - x_2 - x_3 + y \leq 0$$
$$x_1 - x_2 - x_3 - y \leq 0 \qquad -x_1 + x_2 - x_3 - y \leq 0$$
$$x_1 + x_2 - x_3 + y \leq 2 \qquad -x_1 + x_2 + x_3 + y \leq 2$$
$$x_1 - x_2 + x_3 + y \leq 2 \qquad x_1 + x_2 + x_3 - y \leq 2$$

With this approach, 2-xor is modeled without using dummy variables. In general, in order to model a given n-xor operation, we obtain the set of valid points of the xor operation in $\mathbb{F}_2^{n+2}$ and calculate its H-representation. Then, Sasaki and Todo's method [26] is applied to find the minimum set of inequalities to represent the xor operation [15].

3.1.3 Model 2

Fu et al. [10] implemented a method to model a 1-xor operation with a single constraint as follows:

$$a + b + c = 2d_1$$

where $a, b, c, d_1 \in \{0, 1\}$. We extend this approach to the n-xor case.

In Table 3.1, constraints are given to model XOR operations up to 5-xor.

Table 3.1: Constraints of n-XOR

n-XOR	XOR	Constraint
1	$a \oplus b = c$	$a + b + c = 2d_1$
2	$a \oplus b \oplus c = d$	$a + b + c + d = 4d_1 - 2d_2$
3	$a \oplus b \oplus c \oplus d = e$	$a + b + c + d + e = 4d_1 - 2d_2$
4	$a \oplus b \oplus c \oplus d \oplus e = f$	$a + b + c + d + e + f = 6d_1 - 4d_2 - 2d_3$
5	$a \oplus b \oplus c \oplus d \oplus e \oplus f = g$	$a + b + c + d + e + f + g = 6d_1 - 4d_2 - 2d_3$

6-xor ($a \oplus b \oplus c \oplus d \oplus e \oplus f \oplus g = h$) can be modeled via the following equality:

$$a + b + c + d + e + f + g + h = 8d_1 - 6d_2 - 4d_3 - 2d_4.$$

Also, 7-xor ($a \oplus b \oplus c \oplus d \oplus e \oplus f \oplus g \oplus h = i$) can be modeled as:

$$a + b + c + d + e + f + g + h + i = 8d_1 - 6d_2 - 4d_3 - 2d_4.$$

In general, for an even value of n, the n-xor operation $a_0 \oplus a_1 \oplus \cdots \oplus a_n = b$ is modeled as,

$$a_0 + a_1 + \cdots + a_n + b = (n+2)d_1 - \big(nd_2 + (n-2)d_3 \cdots + 2d_{(n/2)+1}\big),$$

and for an odd value of n:

$$a_0 + a_1 + \cdots + a_n + b = (n+1)d_1 - \big((n-1)d_2 + (n-3)d_3 + \cdots + 2d_{(n-1/2)+1}\big).$$

In Table 3.2, we compare the number of variables and constraints that are needed to represent the n-xor operation in three alternative models [16].

Table 3.2: Number of variables and constraints used to represent n-xor.

	Standard xor		Model 1		Model 2	
n-xor	# Variables	# Constraints	# Variables	# Constraints	# Variables	# Constraints
1	3	4	3	4	4	1
2	5	8	4	8	6	1
3	7	12	5	16	7	1
4	9	16	6	32	9	1
5	11	20	7	64	10	1
6	13	24	8	128	12	1
7	15	28	9	256	13	1

3.2 MILP Modelling

MILP modeling of a cipher begins with formulating the objective function according to the cryptanalysis method to be studied. MILP models are used to minimize or maximize an objective function under specified conditions by modeling each step of a cipher as a constraint. For instance, the objective function is chosen to minimize the summation of active S-boxes in order to find the minimum number of active S-boxes or is chosen to maximize the probability information to find the best characteristic.

The round operations of the cipher, such as S-box, permutation, xor, multiplication, and addition are modeled as constraints. The inputs and outputs of these components are defined as variables.

The probability information in the Difference Distribution Tables (DDT) or the Linear Approximation Tables (LAT) is encoded into constraints to be able to find the best differential or linear characteristics. Although the constraints developed for the differential and linear models are mostly similar, the constraints modeling the S-box and the matrix multiplication operations differ significantly between the two attack types.

3.2.1 S-box

Sun et al. [32] provided a method in which the S-box is modeled to find exact solutions.

Let a 4×4 bijective S-box have the input (x_0, x_1, x_2, x_3) and the output (y_0, y_1, y_2, y_3). The following inequalities of binary variables can be used to represent the activity of this S-box and $A = 1$ means that the S-box is active.

$$
\begin{aligned}
x_0 - A &\leq 0 \\
x_1 - A &\leq 0 \\
x_2 - A &\leq 0 \\
x_3 - A &\leq 0 \\
x_0 + x_1 + x_2 + x_3 - A &\geq 0 \\
4(x_0 + x_1 + x_2 + x_3) - (y_0 + y_1 + y_2 + y_3) &\geq 0 \\
4(y_0 + y_1 + y_2 + y_3) - (x_0 + x_1 + x_2 + x_3) &\geq 0
\end{aligned}
$$

We encode input and output in a binary vector, defined as:

$$\mathcal{Q} := (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3).$$

Furthermore, H-representation is a method for representing input vectors as a set of linear inequalities, which is an intersection of half-spaces. We calculate the H-representation of $\mathcal{E}$, denoted by $\mathcal{H}(\mathcal{Q})$, and obtain a set of linear inequalities. Via the

H-representation, we obtain a list of inequalities such as:

$$(\gamma_{0,0}, \gamma_{0,1}, \cdots, \gamma_{0,7}) \cdot \mathcal{Q} + \gamma_{0,8} \leq 0$$
$$\vdots$$
$$(\gamma_{t-1,0}, \gamma_{t-1,1}, \cdots, \gamma_{t-1,7}) \cdot \mathcal{Q} + \gamma_{t-1,8} \leq 0$$

where $\gamma_{i,j}$ are integer coefficients, $0 \leq j \leq 8$ and $0 \leq i < t$, where t denotes the total number of inequalities computed in H-representation. The H-representation of these valid points is calculated and 16 constraints are obtained, some of which are redundant.

Sun et al. [30] proposed a greedy approach to reduce the number of inequalities. Redundant equations are eliminated with Sasaki and Todo's method [26] that ensures the minimum number of inequalities for the representation of an S-box. Furthermore, if exact probability bounds are sought, the Difference Distribution Table (DDT) or the Linear Approximation Table (LAT) should be included in the model.

Suppose we want to model a 4×4 S-box with the probability of a difference,

$$p = Pr[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)],$$

and there are three distinct probabilities in its DDT such as 2^{-3}, 2^{-2}, and 1. The probability information is encoded in two bits as (π_0, π_1), denoting the binary encoding of $-\log_2 p$ as:

$$(\pi_0, \pi_1) = (0, 0) \implies p = 1$$
$$(\pi_0, \pi_1) = (0, 1) \implies p = 2^{-2}$$
$$(\pi_0, \pi_1) = (1, 1) \implies p = 2^{-3}$$

Then, we encode input, output, and probability information in a binary vector, defined as:

$$\mathcal{E} := (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, \pi_0, \pi_1).$$

Via the H-representation, we obtain a list of inequalities such as:

$$(\gamma_{0,0}, \gamma_{0,1}, \cdots, \gamma_{0,9}) \cdot \mathcal{E} + \gamma_{0,10} \leq 0$$
$$\vdots$$
$$(\gamma_{t-1,0}, \gamma_{t-1,1}, \cdots, \gamma_{t-1,9}) \cdot \mathcal{E} + \gamma_{t-1,10} \leq 0$$

where $\gamma_{i,j}$ are integer coefficients, $0 \leq j \leq 10$ and $0 \leq i < t$, where t denotes the total number of inequalities computed in H-representation. SageMath [34] is used to calculate the H-representation of the vectors.

For the linear case, the Linear Approximation Table (LAT) is considered and we apply the same procedure to represent the linear behavior of an S-box. Let b denote the probability of a linear bias: $b = Pr[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)]$, in which x_i and y_i are input and output vectors of an S-box. The exact probability values can be encoded by two bits (b_1, b_0) denoting the binary encoding of $-\log_2 b$ as:

$$\begin{aligned}(b_1, b_0) = (0, 0) &\implies b = 2^{-1}\\ (b_1, b_0) = (1, 0) &\implies b = 2^{-2}\\ (b_1, b_0) = (1, 1) &\implies b = 2^{-3}.\end{aligned}$$

Then the input, output, and probability entries in the LAT are encoded in binary vectors as:

$$v = (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, b_1, b_0).$$

3.2.2 Permutation

Let the input of the permutation Π be a_i and the output of the permutation be b_i for $0 \leq i < n$, where n is the block size of the permutation. In order to model this operation, binary variables b_i are defined to represent the output. Then, equations representing the permutation operation, $b_i = \Pi(a_i)$ for $0 \leq i < n$, are added to the MILP model as constraints.

3.2.3 MixColumn

Differential Case: In order to represent the MDS matrix, the primitive matrix representation provided by [29] is utilized for differential propagation. Let $M_{\mathcal{PR}}$ denote the $m \times m$ binary matrix which is the primitive representation of M over $GF(2)$, obtained by replacing the field elements in M by the $m \times m$ binary matrices. That is consider two field elements a and x in $GF(2^m)$. Multiplication of x by a defines a linear transformation of x. Hence, when x is represented as an m-bit vector over $GF(2)$, multiplication by a has an $m \times m$ matrix representation, which we denote by $\mathbf{a}$. Accordingly, when we need to represent the MDS operation in the cipher, which is multiplication by a matrix M with entries from $GF(2^m)$, as a linear transformation of the given input vector with entries from $GF(2)$, we replace each entry in M by its matrix representation and obtain the binary primitive representation of M , denoted by $M_{\mathcal{PR}}$.

For the state matrices Y and Z where $Z = MY$, let $Y_{\mathcal{B}}$ and $Z_{\mathcal{B}}$ denote the $n \times m$ binary matrices, where each column vector is obtained from the corresponding column vector of Y and Z by replacing each field element from $GF(2^k)$ by its binary representation over $GF(2)$. Hence, the MDS matrix multiplication over these binary vectors becomes,

$$Z_{\mathcal{B}} = M_{\mathcal{PR}} Y_{\mathcal{B}}.$$

The 1's in each row of $M_{\mathcal{PR}}$ indicate the elements to be XORed when a column vector is multiplied by $M_{\mathcal{PR}}$.

Linear Case: Let $M_{\mathcal{PR}}$ be the $m \times m$ binary matrix which is the primitive representation of M over $GF(2)$. Let $Y_{\mathcal{B}}$ and $Z_{\mathcal{B}}$ be the $m \times n$ binary matrices, where each column vector is obtained from the corresponding column vector of Y and Z by replacing each field element from $GF(2^m)$ by its binary representation over $GF(2)$. Hence, $Z_{\mathcal{B}} = M_{\mathcal{PR}} Y_{\mathcal{B}}$. We can transform a linear mask on each column of $Y_{\mathcal{B}}$ into a linear mask of the corresponding column of $Z_{\mathcal{B}}$ along the following lines:

Let y and z be column vectors such that $z = M_{\mathcal{PR}}\, y$, and β^T be the m-bit row vector (linear mask) indicating the active bits of y in a linear approximation. Then, the corresponding linear mask γ^T on z can be calculated as follows:

$$z = M_{\mathcal{PR}}\, y$$
$$M_{\mathcal{PR}}^{-1}\, z = y$$
$$\beta^T M_{\mathcal{PR}}^{-1}\, z = \beta^T y$$

Hence, $\gamma^T z = \beta^T y$ for,

$$\gamma^T = \beta^T M_{\mathcal{PR}}^{-1}.$$

3.2.4 Construction of the Objective Function

The objective function of a MILP model can be constructed either to minimize the number of active S-boxes or to maximize the probability of a characteristic. Models that involve probabilities are preferred whenever possible because they yield the exact best characteristic, but they also tend to be larger and much harder to solve.

In order to find the minimum number of differentially and linearly active s-boxes we minimize the summation of $\sum_i (A_i)$, for A_i denoting S-boxes in binary.

The objective function in differential cryptanalysis is to maximize the characteristic's overall probability $\prod_i p_i$, where p_i denotes the individual round probability. Therefore, the objective function for the differential MILP model becomes to minimize $\sum_i (\pi_{i,0} + 2\pi_{i,1})$, for $(\pi_{i,0}, \pi_{i,1})$ denoting $-\log_2 p_i$ in binary.

The objective function in linear cryptanalysis is to maximize the approximation's overall bias $\prod_i b_i$, where b_i denotes the individual round biases (in absolute value). For $(\pi_{i,0}, \pi_{i,1})$ denoting $-\log_2 b_i$ in binary, the objective function for the linear MILP model is to minimize $\sum_i (\pi_{i,0} + 2\pi_{i,1})$.

3.3 Experimental Setup

The experiments were performed on a computer with a 2.3 GHz Quad-Core Intel Core i5 processor and 8 GB of RAM, and the MILP models in this thesis were solved using the Gurobi optimizer [12] version 9.0.2. The H-representations were calculated using SageMath [34]. The reported timing results are CPU times in seconds.

The MILP models we constructed for differential and linear cryptanalysis are available at `https://github.com/murat-ilter`.

CHAPTER 4

MILP MODELING OF KLEIN

This chapter explains the MILP models we developed for linear and differential cryptanalysis of KLEIN.

We obtain the exact minimum number of differentially active S-boxes of KLEIN for each round. We were able to identify the best single-key linear and differential characteristics for up to 7 rounds of the cipher.

4.1 KLEIN Cipher

KLEIN [11] is a lightweight block cipher that was designed for embedded systems. There are three versions of this cipher with 64-bit, 80-bit, and 96-bit key sizes, and with 12, 16, and 20 rounds, respectively. All versions have a block size of 64 bits.

The cipher has a square SPN structure, similar to AES: The 64-bit round input is organized as a square 4×4 matrix of 4-bit nibbles, and goes through the round operations of SubNibbles (SN), RotateNibbles (RN), and MixNibbles (MN):

SubNibbles: Each nibble is substituted according to the 4×4 S-box of KLEIN given in Table 4.1:

Table 4.1: S-box of KLEIN.

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	7	4	A	9	1	F	B	0	C	3	2	6	8	E	D	5

RotateNibbles: RotateNibbles operation is given in Table 4.2::

Table 4.2: Permutation of KLEIN.

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3

where 0 denotes the most significant byte position.

MixNibbles: The block is multiplied by the MDS matrix M,

$$M = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

defined over the finite field $GF(2^8) = GF(2)/\langle x^8 + x^4 + x^3 + x + 1 \rangle$ for diffusion. The nibbles $c_0^i, c_1^i, \cdots, c_{15}^i$ are organized into two 4×1 byte vectors and multiplied by M:

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_0^i || c_1^i \\ c_2^i || c_3^i \\ c_4^i || c_5^i \\ c_6^i || c_7^i \end{pmatrix} = \begin{pmatrix} d_0^i || d_1^i \\ d_2^i || d_3^i \\ d_4^i || d_5^i \\ d_6^i || d_7^i \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} c_8^i || c_9^i \\ c_{10}^i || c_{11}^i \\ c_{12}^i || c_{13}^i \\ c_{14}^i || c_{15}^i \end{pmatrix} = \begin{pmatrix} d_8^i || d_9^i \\ d_{10}^i || d_{11}^i \\ d_{12}^i || d_{13}^i \\ d_{14}^i || d_{15}^i \end{pmatrix}$$

The inverse matrix,

$$M^{-1} = \begin{pmatrix} E & B & D & 9 \\ 9 & E & B & D \\ D & 9 & E & B \\ B & D & 9 & E \end{pmatrix}$$

with entries from $GF(2^8)$, is used for the decryption operation.

4.2 Differential MILP Model of KLEIN

The details of the MILP model for differential cryptanalysis of KLEIN is given in this section.

SubNibbles: In the DDT of KLEIN's S-box, the differential probabilities are 1, 2^{-2}, and 2^{-3}. Possible patterns with probability information are added to the MILP model, as described in Section 3.2. Then we computed the H-representation with SageMath, obtaining 2489 inequalities. Applying Sasaki and Todo's reduction method on the H-representation, we obtained 21 inequalities representing the DDT of KLEIN's S-box with the related probability information.

RotateNibbles: This operation is modeled inside the MixNibbles operation.

MixNibbles: The primitive representation of M is a binary matrix $M_{\mathcal{PR}}$ where the entries 1, 2, 3 in M are replaced by,

$$\mathbf{1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{2} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{3} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrices corresponding to the primitive representations of 1, 2, and 3 are substituted in the M matrix, and a new 32×32 binary matrix is obtained, which can be used to model a matrix multiplication as a set of xor operations.

For instance, in the first round output the following equations are obtained for $(d_0^1||d_1^1)$:

$$
\begin{aligned}
d_0^1[0] =& c_4^1[1] \oplus c_6^1[0] \oplus c_6^1[1] \oplus c_8^1[0] \oplus c_{10}^1[0] \\
d_0^1[1] =& c_4^1[2] \oplus c_6^1[1] \oplus c_6^1[2] \oplus c_8^1[1] \oplus c_{10}^1[1] \\
d_0^1[2] =& c_4^1[3] \oplus c_6^1[2] \oplus c_6^1[3] \oplus c_8^1[2] \oplus c_{10}^1[2] \\
d_0^1[3] =& c_4^1[0] \oplus c_5^1[0] \oplus c_6^1[0] \oplus c_6^1[3] \oplus c_7^1[0] \oplus c_8^1[3] \oplus c_{10}^1[3] \\
d_1^1[0] =& c_4^1[0] \oplus c_5^1[1] \oplus c_6^1[0] \oplus c_7^1[0] \oplus c_7^1[1] \oplus c_9^1[0] \oplus c_{11}^1[0] \\
d_1^1[1] =& c_5^1[2] \oplus c_7^1[1] \oplus c_7^1[2] \oplus c_9^1[1] \oplus c_{11}^1[1] \\
d_1^1[2] =& c_4^1[0] \oplus c_5^1[3] \oplus c_6^1[0] \oplus c_7^1[2] \oplus c_7^1[3] \oplus c_9^1[2] \oplus c_{11}^1[2] \\
d_1^1[3] =& c_4^1[0] \oplus c_6^1[0] \oplus c_7^1[3] \oplus c_9^1[3] \oplus c_{11}^1[3]
\end{aligned}
$$

These equations of multiple xors are written as inequalities and added to the MILP model as constraints. For instance, for the representations of $d_0^1[0]$ and $d_0^1[3]$, 4-xor and 6-xor models are used, respectively. In order to model this 32×32 matrix multiplication, it is enough to use 4-xor and 6-xor models. which are calculated according to the underlying finite field $GF(2^8) = GF(2)/\langle x^8 + x^4 + x^3 + x + 1\rangle$.

The 32×32 binary matrix $M_{\mathcal{PR}}$ is obtained by substituting $\mathbf{1}, \mathbf{2}$ and $\mathbf{3}$ in M.

KLEIN-64 is modeled using the standard xor model and Model 1 in order to obtain the exact minimum number of active S-boxes. The results are given in Table 4.3.

In order to find the best differential characteristic, the S-box differential values are represented with probability information. There exist three non-zero probabilities 1, 2^{-2}, and 2^{-3} in DDT. These probabilities are encoded with the corresponding possible patterns as described by Sun et al. [31]. The H-representation is calculated, and 2489 inequalities are obtained. Adopting the reduction method of Sasaki and Todo, 21 equations are shown to be enough for the representation of the S-box. The best single-key differential characteristic for 7 rounds with a probability of 2^{-59} is given in Table 4.4. [1] The best differential characteristics with three models are presented in Table 4.5.

[1] The lines with an (*) indicate that the search did not conclude within the given time limit and possibly better characteristics may exist.

Table 4.3: Minimum number of differentially active S-box of KLEIN-64.

		Standard xor model			Model 1		
Round	Act. S-box	# of Var.	# of Const.	Time (s.)	# of Var	# of Const.	Time (s.)
2	5	464	1748	1	224	4881	2
3	8	848	3412	80	368	9681	132
4	15	1232	5076	447	512	14484	202
5	18	1616	6740	877	656	19284	584
6	20	2000	8407	1989	800	24087	1760
7	24	2384	10071	3648	944	28887	3331
8	30	2768	11736	10285	1088	33688	5526
9	34	3152	13401	6129	1232	38489	7923
10	36	3536	15066	11687	1376	43290	20248
11	39	3920	16731	112950	1520	48091	39246
12	46	4304	18395	61070	1664	52892	110088

Table 4.4: The best 7-round differential characteristic of KLEIN-64.

Round	Diff.	Prob.
Input	0000 030E 000E 0000	1
1	0000 0B0E 0000 0000	2^{-6}
2	0B0F 0604 0000 0000	2^{-11}
3	000E 020E 010B 060D	2^{-21}
4	0101 0000 0000 0B0E	2^{-39}
5	0000 0000 0101 0000	2^{-49}
6	0006 0305 0000 0000	2^{-53}
7	0118 0519 0606 0A0C	2^{-59}

Table 4.5: Complexity of the alternative xor models for linear MILP solutions of KLEIN.

		Standard xor			Model 1			Model 2		
R	Prob.	#V.	#C.	T (s.)	# V.	# C.	T (s.)	# V.	# C.	T (s.)
2	2^{-10}	592	2113	14	352	5249	14	568	961	9
3	2^{-17}	1008	3777	30373	528	10049	15074	960	1473	2322
4	2^{-32}	1424	5444	136556	704	14852	50582	1352	1988	77279
5	2^{-42}	1840	7109	881567	880	19653	382301	1744	2501	297421
6 (*)	2^{-48}	2256	8769	>1000000	1056	24449	>1000000	2136	3013	>1000000
7 (*)	2^{-59}	2672	10439	>1000000	1232	29255	>1000000	2528	3527	>1000000

4.3 Linear MILP Model of KLEIN

The MILP model for linear cryptanalysis of KLEIN is constructed along the following lines, where the main differences from the differential model are objective function, representation of the S-box and MDS matrix multiplication operations:

SubNibbles: Three different bias values exist in the LAT of KLEIN: 2^{-1}, 2^{-2}, 2^{-3}. 1633 inequalities are acquired by means of computing the H-representation of possible patterns, which in turn can be reduced to 33 inequalities by Sasaki and Todo's reduction method.

RotateNibbles: This operation is modeled inside MixNibbles.

MixNibbles: $M_{\mathcal{PR}}$ is the primitive representation of M over $GF(2)$, which is a 32×32 binary matrix, as explained in Section 4.2. Let y and z be the 32×1 binary column vectors denoting the input and the output of a matrix multiplication operation in MixNibbles operation; i.e., $z = M_{\mathcal{PR}}y$.

The entries of M^{-1} are 9, B, D, E, which are replaced by $\mathbf{9}$, **B**,**D** and **E** in $M_{\mathcal{PR}}^{-1}$, according to the underlying finite field polynomial $GF(2)/\langle x^8 + x^4 + x^3 + x + 1\rangle$:

$$\mathbf{9} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As in Section 4.2, multiplication of a vector by the binary matrix $M_{\mathcal{PR}}^{-1}$ is modeled with multiple xor operations.

The solution complexity of the models, including the number of constraints, the number of variables, and the execution time (in CPU seconds), is given in Table4.6 for the linear models. [2] In linear cryptanalysis, Model 1 turned out to produce too many constraints to be handled by SageMath for the H-representation calculation and hence was excluded from the linear experiments.

Table 4.6: Complexity of the alternative xor models for linear MILP solutions of KLEIN.

		Standard xor			Model 2		
Round	Bias	#V.	#C.	T (s.)	# V.	# C.	T (s.)
2	2^{-6}	1168	4801	564	856	1345	67
3	2^{-9}	2160	8964	107040	1536	2052	17320
4	2^{-17}	3152	13124	>1000000	2216	2756	448893
5 (*)	2^{-24}	4144	17285	>1000000	2896	3461	>1000000
6 (*)	2^{-27}	5136	21445	>1000000	3576	4165	>1000000

The best linear characteristics we found for 6 rounds with the bias of 2^{-27} are given in Table 4.7.

[2] The lines with an (*) indicate that the search did not conclude within the given time limit and possibly better characteristics may exist.

Table 4.7: The best 6-round linear characteristic of KLEIN-64.

Round	Bias	Prob.
Input	0000 060A 0300 0000	1
1	0404 0000 0000 0000	2^{-4}
2	0000 0000 0201 0506	2^{-6}
3	0506 0501 0007 0707	2^{-12}
4	0D09 0000 0000 0400	2^{-21}
5	0000 0000 0700 0400	2^{-25}
6	EBA9 672D 8284 8687	2^{-27}

CHAPTER 5

MILP MODELING OF PRINCE

This chapter explains the MILP models we developed for linear and differential cryptanalysis of PRINCE. We find the exact minimum number of the differential active S-boxes of PRINCE for each round. Also, we discover the best linear and differential characteristics for up to 7 rounds of the cipher.

5.1 PRINCE Cipher

PRINCE [5] is a 64-bit block cipher with a 128-bit key and 12 rounds. The cipher has a square SPN structure, similar to AES: The 64-bit round input is organized as a square 4×4 matrix of 4-bit nibbles and goes through a series of rounds consisting of a substitution and a linear diffusion layer.

In the substitution layer, each nibble is substituted according to the 4×4 S-box given in Table 5.1:

Table 5.1: S-box of PRINCE.

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

The diffusion layer consists of a shift row and a matrix multiplication operation. The linear layer consists of a shift row (SR) operation and the matrix M multiplication. The shift row is identical to the one in AES but operates on 4-bit nibbles instead of bytes. The shift row operation changes the position of the nibbles. This operation is given in Table 5.2.

Table 5.2: Permutation of PRINCE.

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	5	A	F	4	9	E	3	8	D	2	7	C	1	6	B

The matrix multiplication operation is based on a 64×64 binary matrix M' constructed from a number of sub-matrices, as explained below:

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\hat{M}^{(0)} = \begin{pmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{pmatrix} \quad \hat{M}^{(1)} = \begin{pmatrix} M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{pmatrix}$$

M' is the 64×64 matrix where the diagonal blocks are $(\hat{M}^{(0)}, \hat{M}^{(1)}, \hat{M}^{(1)}, \hat{M}^{(0)})$ and the rest are 0s.

5.2 Differential MILP Model of PRINCE

S-box Layer: DDT of the S-box of PRINCE has 106 non-zero entries. H-representation of these possible patterns is calculated, and 300 inequalities are obtained. Applying Sasaki and Todo's reduction method, 22 inequalities are obtained to represent the S-box difference patterns of PRINCE.

Linear Layer: In the linear layer, there is a 64×64 binary matrix M' multiplication. There are three 1s in each row of matrix M'. Hence the equations of the matrix multiplications have the form:

$$d_0^1[0] = c_1^1[0] \oplus c_2^1[0] \oplus c_3^1[0].$$

Therefore, we need 2-xor models to represent the matrix multiplication M'. They are written as inequalities and added to the MILP model as constraints.

PRINCE is modeled using the standard xor model and Model 1. The results are compared in Table 5.3.

Table 5.3: Minimum number of differentially active S-box of PRINCE with standard xor model and Model 1.

		Standard Xor Model			Model 1		
R	Act. S-box	# of Var.	# of Const.	Time (s.)	# of Var.	# of Const.	Time (s.)
2	4	288	1121	1	224	1121	1
3	7	560	2161	19	432	2161	7
4	16	832	3204	20	640	3204	5
5	19	1104	4244	57	848	4244	45
6	20	1376	5284	599	1056	5284	208
7	23	1648	6262	437	1264	6262	404
8	32	1920	7303	1245	1472	7303	1425
9	35	2192	8343	1890	1680	8343	1688
10	36	2464	9384	5602	1888	9384	4981
11	39	2736	10425	19374	2096	10425	12272
12	48	3008	11466	26889	2304	11466	21780

In the design paper of PRINCE[5], the authors calculated the minimum number of differentially active S-boxes to be at least 48. By our MILP model, we showed that the actual number is exactly 48.

In order to find the best differential characteristic, the probability information of DDT is added to the representation of the S-box and the inverse S-box. There exist three non-zero probabilities, 1, 2^{-2}, and 2^{-3} in DDT. These probabilities are encoded with the corresponding possible differential patterns as given in Section 3.2. The H-representation is calculated, and 1975 constraints are obtained. Adopting the reduction method of Sasaki and Todo, 22 constraints are shown to be enough for the representation of the S-box and the inverse S-box. In Table 5.4, the best differential characteristics are presented for various numbers of rounds.

Table 5.4: Complexity of the alternative xor models for differential MILP solutions of PRINCE.

	Standard xor model			Model 1			Model 2		
Round	#V.	#C.	T (s.)	# V.	# C.	T (s.)	# V.	# C.	T (s.)
2	480	1475	3	416	1475	2	544	1027	1
3	784	2500	1302	656	2500	464	912	1604	206
4	1088	3524	159462	896	3524	15368	1280	2180	38705
5	1392	4548	177410	1136	4548	290543	1648	2756	141780
6	1696	5575	330389	1376	5575	235481	2016	3335	575157
7	1937	6536	431921	1552	6536	303585	2320	3848	365911

Previously, the best single-key differential characteristic on PRINCE in the literature was obtained for 6 rounds, with a probability of 2^{-62} [2]. Using the MILP model, we discovered a single-key differential characteristic for 7 rounds with a probability of 2^{-56} which is given in Table 5.5.

Table 5.5: Best 7-round Differential Characteristic of PRINCE.

Round	Diff.	Prob.
Input	0041 C800 0000 0000	1
1	1100 0000 0000 0110	2^{-8}
2	0000 0011 0110 0000	2^{-16}
3	0000 1100 1001 0000	2^{-24}
4	0110 0000 0000 0011	2^{-32}
5	0000 0088 0880 0000	2^{-40}
6	0000 0440 0044 0000	2^{-48}
7	9A3B 3B9A 9A2B 9A3B	2^{-56}

5.3 Linear MILP Model of PRINCE

The MILP model for linear cryptanalysis of PRINCE is constructed along the following lines:

S-box Layer: The LAT of PRINCE's S-box is modeled with 1202 inequalities in the H-representation. Sasaki and Todo's method is applied, and 33 constraints are enough to represent the LAT.

Linear Layer: Since PRINCE uses an involutory matrix, the constraints that are needed to model the inverse of M' are identical to those used to model M' in the differential model.

We utilized the alternative xor models described in Section 3.1 to model the matrix multiplication operation in PRINCE and compared their efficiency. The solution complexity of the models, including the number of variables, the number of constraints, and the execution time (in CPU seconds), is presented in Table 5.6 for the linear models.

Table 5.6: Complexity of the alternative xor models for linear MILP solutions of PRINCE.

	Standard xor			Model 1			Model 2		
Round	#V.	#C.	T (s.)	# V.	# C.	T (s.)	# V.	# C.	T (s.)
2	480	1859	3	416	1859	2	544	1411	1
3	784	3076	831	656	3076	324	912	2180	73
4	1088	4293	27592	896	4293	24513	1280	2949	91409
5	1392	5510	21610	1136	5510	68815	1648	3718	14601
6	1696	6727	23807	1376	6727	79587	2016	4487	25981
7	1936	7880	156500	1552	7880	47481	2320	5192	74070

Using the MILP model, we discovered a single-key linear characteristic for 7 rounds with a bias of 2^{-29} which is given in Table 5.7.

Table 5.7: Best 7-round Linear Characteristic of PRINCE.

Round	Linear Mask	Bias.
Input	0440 4004 0000 0000	1
1	2002 0020 0000 0003	2^{-5}
2	2400 0000 0000 0240	2^{-9}
3	2002 0000 0000 2200	2^{-13}
4	0000 0000 0220 2200	2^{-17}
5	0000 0000 4200 2004	2^{-21}
6	0000 0000 2002 0220	2^{-25}
7	4044 0044 4044 0000	2^{-29}

CHAPTER 6

MILP MODELING OF FUTURE

This chapter explains the MILP models we developed to find the best linear and differential characteristics of FUTURE. We were able to identify single-key linear and differential characteristics for up to 5 rounds of the cipher.

6.1 FUTURE Cipher

FUTURE is an AES-like block cipher, where the operations are carried out on nibbles rather than bytes. It has a 10-round lightweight structure, designed for low latency and low hardware cost. The S-box and the MDS matrix are designed especially to be efficient in hardware. FUTURE block size is 64 bits, and the key length is 128 bits.

SubCell: The 4×4 S-box of FUTURE which is a composition of 4 different lightweight S-boxes is given in Table 6.1.

Table 6.1: S-box of FUTURE

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	1	3	0	2	7	E	4	D	9	A	C	6	F	5	8	B

ShiftRow The ith row of the state matrix ($0 \leq i \leq 3$) is shifted to the right, depending on the value of i:

$$\begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix} \leftarrow \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_{13} & s_1 & s_5 & s_9 \\ s_{10} & s_{14} & s_2 & s_6 \\ s_7 & s_{11} & s_{15} & s_3 \end{pmatrix}$$

MixColumn The finite field multiplication of FUTURE is done over $GF(2^4) = GF(2)/\langle x^4 + x + 1 \rangle$. The state matrix entries are considered elements in $GF(2^4)$ and multiplied with the MDS matrix M, as $X \leftarrow MX$:

$$M = \begin{pmatrix} 8 & 9 & 1 & 8 \\ 3 & 2 & 9 & 9 \\ 2 & 3 & 8 & 9 \\ 9 & 9 & 8 & 1 \end{pmatrix}$$

AddRoundKey: The 64-bit round key is XORed to the state of the cipher.

The Round Function: The basic round operations of FUTURE are SubCell, MixColumn, ShiftRow, and AddRoundKey. The MixColumn operation is omitted in the final round. The state of the cipher is denoted by a 4×4 matrix X where each entry is a nibble; i.e., $s_i \in \{0,1\}^4$ for $0 \leq i \leq 15$:

$$X = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}$$

6.2 Differential MILP Model of FUTURE

The round function elements of FUTURE, namely the SubCell, MixColumn, and ShiftRow operations, are modeled for differential cryptanalysis using the techniques described below:

SubCell: The DDT is calculated for the S-box of FUTURE, which contains three non-zero values; 2, 4, and 16. As described in Section 3.2.3, we encoded each input, output, and probability information as a vector, and computed the H-representation using SageMath. The solution returned 333 inequalities including redundant ones. We utilized Sasaki and Todo's approach and obtained 18 inequalities to represent the S-box's differential behavior.

MixColumn: In order to represent the MDS matrix, the primitive matrix representation provided by [29] is utilized for differential propagation. FUTURE's MDS matrix

M contains the field elements $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, $\mathbf{8}$, $\mathbf{9}$ from $GF(2^4)$. Field multiplication by these scalars in $GF(2^4)$ is a linear transformation over $GF(2)$, represented via the following matrices:

$$\mathbf{1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{2} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{3} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{8} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \mathbf{9} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Let $M_{\mathcal{PR}}$ denote the 16×16 binary matrix which is the primitive representation of M over $GF(2)$, obtained by replacing the field elements in M by the 4×4 binary matrices given above as explained in Section 3.2.3. The 1's in each row of $M_{\mathcal{PR}}$ indicate the elements to be XORed when a column vector is multiplied by $M_{\mathcal{PR}}$.

To model the differential propagation over each MDS matrix multiplication, we need 64 new constraints and 204 new binary d_i dummy variables.

ShiftRow: The binary variables resulting from the MixColumn operation are permuted through the ShiftRow operation. Then, 64 new binary variables are introduced and assigned to these results.

AddRoundKey: Since we model single-key differential cryptanalysis, there is no need to model the XOR operation with the round key.

Search Strategy: The number of variables and constraints used in the MILP model increases as more rounds are added to the model, and the solution time increases exponentially as a result. Zhou et al. [37], in their MILP analysis of the GIFT cipher, added extra constraints to the model, to limit the number of active S-boxes in each round and hence to restrict the solution space. We adopted a similar approach to obtain differential characteristics of FUTURE. For instance, the 4-round differential characteristic is obtained by adding the following four constraints:

$$A_0^0 + A_1^0 + \cdots A_{15}^0 = 4$$
$$A_0^1 + A_1^1 + \cdots A_{15}^1 = 1$$
$$A_0^2 + A_1^2 + \cdots A_{15}^2 = 4$$
$$A_0^3 + A_1^3 + \cdots A_{15}^3 = 16$$

where A_j^i stands for the jth S-box in the ith round. These extra constraints are used to determine the number of active S-boxes in each round, such as 4-1-4-16 in this example search strategy.

In Table 6.2, the best differential probabilities are given with respect to the search strategies we tried for up to five rounds.

Table 6.2: The search strategies tried and the maximum differential probabilities obtained for FUTURE up to 5 rounds.

# of rounds	Extra Constraint	Max. Diff. Prob.	# of Var.	# of Cons.
2	1-4	2^{-10}	620	930
3	4-1-4	2^{-18}	1064	1458
4	4-1-4-16	2^{-51}	1508	1986
	1-4-16-4	2^{-55}		
	16-4-1-4	2^{-50}		
	4-16-4-1	2^{-53}		
5	4-1-4-16-4	2^{-63}	1952	2518
	1-4-16-4-1	2^{-58}		
	2-16-4-1-2	2^{-61}		
	2-4-16-4-1	2^{-58}		
	1-4-16-4-2	2^{-61}		

A 5-round characteristic with 2^{-58} probability has been found through our searches. Remarkably, one of these characteristics involves 27 active S-boxes, which is not the minimum number of active S-boxes for 5 rounds.

Designers of FUTURE provided a 4-round differential characteristic with a probability of 2^{-62}. We were able to obtain the probability 2^{-58} for a 5-round characteristic. The details of the 5-round characteristic are given in Table 6.3.

Table 6.3: Differential characteristic of FUTURE for 5 round

Round	Difference	Diff. Prob.
Input	0704 0000 0000 0000	1
1	4000 0700 0050 0007	2^{-4}
2	6161 1C16 4482 3262	2^{-13}
3	0000 0000 0000 6122	2^{-48}
4	0000 0000 0002 0000	2^{-56}
5	0090 0001 8000 0900	2^{-58}

6.3 Linear MILP Model of FUTURE

In this section, we describe the details of the MILP model constructed for linear cryptanalysis of FUTURE and how it is implemented in practice. We focus on how a linear approximation of the S-box can be transformed into a linear approximation of the round function, propagating through the MDS matrix multiplication.

SubCell: We calculated the LAT for FUTURE's S-box, and, as described in Section 3.2, we encoded each input, output, and bias (in absolute value) information as a vector. Then we computed the H-representation. The solution returned 505 inequalities including redundant ones. We utilized Sasaki and Todo's approach and obtained 18 inequalities to represent the S-box's linear behavior.

MixColumn: Let $M_{\mathcal{PR}}$ be the 16×16 binary matrix which is the primitive representation of M over $GF(2)$, as explained in Section 3.2.3, and let $Y_{\mathcal{B}}$ and $Z_{\mathcal{B}}$ be the 16×4 binary matrices, where each column vector is obtained from the corresponding column vector of Y and Z by replacing each field element from $GF(2^4)$ by its binary representation over $GF(2)$.

We need 64 new constraints and 200 new binary d_i dummy variables are needed to model linear propagation over each MDS matrix multiplication,

ShiftRow: The binary variables resulting from the MixColumn operation are permuted through the ShiftRow operation. 64 new binary variables are defined and assigned to these results as introduced in Section 3.2.2.

AddRoundKey: There is no need to model the XOR operation with the round key since linear cryptanalysis is conducted.

Search Strategy: As explained in Section 6.2, the number of variables and constraints used in the MILP model increases as more rounds are added to the model, and the solution time increases exponentially as a result. To tackle this problem and to keep the MILP search within practical limits, we add extra constraints that indicate the number of active S-boxes in each round. The search strategies we used in our search of linear approximations of FUTURE are listed in Table 6.4.

The linear approximation biases (in absolute values) up to five rounds are given in Table 6.4.

Table 6.4: The search strategies tried and the maximum linear biases obtained for FUTURE up to 5 rounds.

# of rounds	Extra Constraint	Max. Linear Bias	# of Var.	# of Cons.
2	1-4	2^{-6}	616	930
3	4-1-4	2^{-10}	1056	1458
4	16-4-1-4	2^{-26}	1496	1986
5	1-4-16-4-1	2^{-32}	1936	2518
	1-4-16-4-2	2^{-31}		
	2-4-16-4-1	2^{-32}		

A 5-round approximation with a bias of 2^{-31} has been found through our searches. The details of the 5-round characteristic are given in Table 6.5.

Table 6.5: Linear characteristic of FUTURE for 5-round

Round	Input Mask	Linear Bias
Input	0000 0000 0090 0000	1
1	0080 0001 1000 0900	2^{-2}
2	1EF4 79B4 338A FF41	2^{-6}
3	0000 0000 8D73 0000	2^{-25}
4	0000 0000 D000 0F00	2^{-29}
5	0150 00E7 D007 8500	2^{-31}

We compare the solution times of differential and linear characteristics of FUTURE modeled with the n-XOR method and the method proposed by Ilter and Selcuk [16] in Table 6.6 and Table 6.7.

As shown in Table 6.6 and in Table 6.7, the Model 2 uses fewer constraints to model xor operation, leading to shortening solution time.

Table 6.6: Timing comparison of XOR methods for differential characteristics of FUTURE

Round	Ext. Cons.	Model 1			Model 2		
		# of Var.	# of Cons.	Time (s.)	# of Var.	# of Cons.	Time (s.)
2	-	416	4961	4	620	929	2
3	4-1-4	656	10545	30	1064	1457	2
4	16-4-1-4	896	15621	445	1508	1986	193
4	4-1-4-16	896	15621	478	1508	1986	54

Table 6.7: Timing comparison of XOR methods for linear characteristics of FUTURE

R	Ext. Cons.	Model 1			Model 2		
		# of Var.	# of Cons.	T (s.)	# of Var.	# of Cons.	T(s.)
2	-	416	5217	61	616	929	11
3	4-1-4	656	10036	10	1056	1460	1
4	16-4-1-4	896	14853	579	1496	1989	13
4	4-1-4-16	896	14853	260	1496	1989	27

CHAPTER 7

MILP MODELING OF IVLBC

This chapter explains the MILP models we developed for linear and differential cryptanalysis of IVLBC. We were able to identify the best single-key linear and differential characteristics for up to 7 rounds of the cipher.

7.1 IVLBC

IVLBC [14] is an SPN type of block cipher with 28 rounds. The block size is 64-bit and it supports 80-bit and 128-bit keys. The round operations are Add-RoundKey, Sub-Cells, Permute-Nibbles, and Mix-Columns. These are designed as involutive, therefore decryption is the same as encryption.

Sub-Cells: IVLBC uses 4×4 S-box which is given in Table 7.1:

Table 7.1: S-box of IVLBC.

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	F	E	5	D	3	6	C	B	9	A	8	7	4	2	1

Permute-Nibbles: IVLBC uses nibble-based involutive permutation that is given in Table 7.2:

Table 7.2: Permutation of IVLBC

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	0	7	A	D	4	B	E	1	8	F	2	5	C	3	6	9

Mix-Columns: Involutive almost MDS matrix M is defined in $GF(2^4)$. The state vector E is multiplied with the matrix M :

$$E = \begin{pmatrix} E_0 & E_4 & E_8 & E_{12} \\ E_1 & E_5 & E_9 & E_{13} \\ E_2 & E_6 & E_{10} & E_{14} \\ E_3 & E_7 & E_{11} & E_{15} \end{pmatrix} M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

The result is denoted as L in which $L = ME$

Add-RoundKey 64-bit round keys are obtained from the master key and denoted as RK_i for round i. Round keys are xored with the state.

Key Generation: Since we conduct differential and linear cryptanalysis, the Key generation is not given. For further details reader may refer to IVLBC design paper[14].

Round Function: IVLBC is designed for 28 rounds. Encryption of IVLBC is given in Algorithm 1.

Algorithm 1 Encryption of IVLBC

Input: Plaintext, Round Keys RK_i

Output: Ciphertext

for i from 1 to 28 **do**

```
    Add-RoundKey(RK_i, State)
    Sub-Cells(State)
    Permute-Cells(State)
    Mix-Columns(State)
```

end for

```
AddRoundKey(RK_29, State)
```

7.2 Differential MILP Model of IVLBC

Sub-Cells: The DDT is calculated for the S-box of FUTURE, which contains three non-zero values; 2, 4, and 16. As described in Section 3.2, we encoded each input, output, and probability information as a vector, and computed the H-representation using SageMath. Sasaki and Todo's method [26] is used to eliminate these redundant equations. 20 equations are needed to represent the differential behavior of IVLBC's

S-box with probability information.

Permute-Nibbles: IVLBC uses nibble permutation. We introduce 64 new binary variables z_i in order to represent the permutation $\mathcal{P}$. Permutation operation is modeled as $z_i = y_{P(i)}$. 64 equations and 64 new variables are needed to model the Permute-Nibbles operation for each round.

Mix-Columns: The primitive representation of the matrix M is a binary matrix $M_{\mathcal{PR}}$ where the entries 0 and 1 in M are replaced by,

$$1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} 0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

which are calculated according to the underlying finite field $GF(2^4)$ as given in Section 3.2.3. Then, the Mix-Columns operation can be represented as a 2-xor operation. 64 equations and 128 dummy variables d_i are needed to model the Mix-Columns operation for each round.

The best differential characteristics we found are given in Table 7.3. The tables list the probability (or, linear bias) of the optimal characteristic for a given number of rounds. The tables also list the total number of variables and constraints involved, indicating the complexity of each MILP model. We were able to go up to 7 rounds for both attack types.

The correctness of the obtained probabilities has been verified by statistical sampling for smaller numbers of rounds.

Table 7.3: The best differential characteristic of IVLBC up to 7 rounds.

Rounds	Diff. Prob	#Var.	#Const.
1	2^{-2}	432	561
2	2^{-8}	800	1121
3	2^{-14}	1168	1681
4	2^{-32}	1536	2241
5	2^{-34}	1904	2801
6	2^{-40}	2272	3361
7	2^{-46}	2640	3921

We discovered a differential characteristic for 7 rounds of IVLBC with a probability of 2^{-46}, which is given in Table 7.4.

Table 7.4: The best 7-round differential characteristic of IVLBC.

Round	Difference	Differential Probability
Input	0001 0010 0000 1000	1
1	0000 0000 0000 0002	2^{-6}
2	0000 0000 1011 0000	2^{-8}
3	2202 2022 0222 0000	2^{-14}
4	0001 0010 0000 1000	2^{-32}
5	0000 0000 0000 0002	2^{-38}
6	0000 0000 1011 0000	2^{-40}
7	2202 2022 0222 0000	2^{-46}

7.3 Linear MILP Model of IVLBC

Sub-Cells: LAT contains elements 2, 4, and 8 (in absolute values). Redundant equations are eliminated with Sasaki and Todo's method [26] and 16 equations are obtained to represent the linear behavior of IVLBC's S-box.

Permute-Nibbles: Permute-Nibbles is modeled the same way in the differential case.

Mix-Columns: Since IVLBC uses an involutional M matrix, Mix-Columns is modeled the same way in the differential case.

The best linear characteristics we found are given in Table 7.5.

Table 7.5: The best linear characteristic of IVLBC up to 7 rounds.

#rounds	Linear Bias	#Var.	#Const.
1	2^{-2}	432	497
2	2^{-5}	800	993
3	2^{-8}	1168	1489
4	2^{-15}	1536	1985
5	2^{-18}	1904	2481
6	2^{-21}	2272	2977
7	2^{-24}	2640	3473

We discovered a linear characteristic for 7 rounds of IVLBC with a bias of 2^{-24}, which is given in Table 7.6.

Table 7.6: The best 7-round Linear characteristic of IVLBC.

Round	Linear Mask	Linear Bias
Input	0000 0A00 3000 0008	1
1	0000 0000 00E0 0000	2^{-4}
2	1101 0000 0000 0000	2^{-5}
3	0222 2220 0000 2022	2^{-8}
4	0000 0100 1000 0001	2^{-15}
5	0000 0000 0030 0000	2^{-20}
6	2202 0000 0000 0000	2^{-21}
7	0111 3330 0000 9099	2^{-24}

CHAPTER 8

CONCLUSION

MILP approach has many application areas in cryptanalysis. Notably, two milestone cryptanalysis methods, differential and linear, can be utilized with MILP to discover cipher resistance against these attacks. In this thesis, we address two main problems; finding the exact minimum number of differentially active S-boxes and identifying the best characteristics using the MILP approach. The task of obtaining solutions to determine the best characteristics through MILP is more challenging than finding the exact minimum number of active S-boxes. Both of these problems require the use of efficient MILP models.

We introduce two alternative MILP modeling methods for representing equations including multiple xor operations. Model 1 employs fewer variables, while Model 2 works with fewer constraints. In general, Model 1 and Model 2 provide shorter solution times with respect to the standard xor model. These developed xor models are quite general and can be applied to other cryptanalysis methods. We apply these novel models to describe matrix multiplication over $GF(2^n)$, with the standard xor model serving as the baseline for comparisons.

Utilizing these three models, we formulate MILP models for analyzing the KLEIN, PRINCE, FUTURE, and IVLBC ciphers. The MILP models developed in this study enable us to precisely determine the minimum number of active S-boxes and identify the best characteristics for different round numbers. Our results are as follows:

- For KLEIN, the exact minimum number of differential active S-boxes is 46 for 12 rounds, the probability of the best single-key differential characteristics is

2^{-59} for 7 rounds, and the bias of the best single-key linear characteristic is 2^{-27} for 7 rounds.

- For PRINCE, the exact minimum number of differential active S-boxes is 48 for 12 rounds, the probability of the best single-key differential characteristics is 2^{-56} for 7 rounds, and the bias of the best single-key linear characteristic is 2^{-29} for 7 rounds.
- For FUTURE, the probability of a single-key differential characteristic is 2^{-58} for 5 rounds, and the bias of the single-key linear characteristic is 2^{-31} for 5 rounds.
- For IVLBC, the probability of the best single-key differential characteristics is 2^{-46} for 7 rounds, and the bias of the best single-key linear characteristic is 2^{-24} for 7 rounds.

The accomplished results improve the best single-key differential and linear characteristics of these ciphers to the extent of our knowledge.

As a future work, the proposed xor models in this thesis can have broad applicability beyond differential and linear cryptanalysis, enhancing their utility in various cryptanalysis methods.

REFERENCES

[1] A. Abdelkhalek, Y. Sasaki, Y. Todo, M. Tolba, and A. M. Youssef, MILP modeling for (large) s-boxes to optimize probability of differential characteristics, IACR Transactions on Symmetric Cryptology, pp. 99–129, 2017.

[2] R. Ankele and S. Kölbl, Mind the gap-a closer look at the security of block ciphers against differential cryptanalysis, in *International Conference on Selected Areas in Cryptography*, pp. 163–190, Springer, 2018.

[3] E. Biham and A. Shamir, Differential cryptanalysis of DES-like cryptosystems, Journal of CRYPTOLOGY, 4, pp. 3–72, 1991.

[4] A. Bogdanov and V. Rijmen, Linear hulls with correlation zero and linear cryptanalysis of block ciphers, Designs, codes and cryptography, 70, pp. 369–383, 2014.

[5] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, et al., PRINCE–a low-latency block cipher for pervasive computing applications, in *International conference on the theory and application of cryptology and information security*, pp. 208–225, Springer, 2012.

[6] J. Borst, L. R. Knudsen, and V. Rijmen, Two attacks on reduced IDEA, in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1–13, Springer, 1997.

[7] C. Boura and D. Coggia, Efficient MILP modelings for sboxes and linear layers of SPN ciphers, IACR Transactions on Symmetric Cryptology, pp. 327–361, 2020.

[8] J. Daemen and V. Rijmen, Rijndael for AES., in *AES Candidate Conference*, pp. 343–348, 2000.

[9] G. B. Dantzig, Linear programming, Operations research, 50(1), pp. 42–47, 2002.

[10] K. Fu, M. Wang, Y. Guo, S. Sun, and L. Hu, MILP-based automatic search algorithms for differential and linear trails for SPECK, in *International Conference on Fast Software Encryption*, pp. 268–288, Springer, 2016.

[11] Z. Gong, S. Nikova, and Y. W. Law, KLEIN: a new family of lightweight block ciphers, in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pp. 1–18, Springer, 2011.

[12] I. Gurobi Optimization, Gurobi optimizer reference manual, URL http://www. gurobi. com, 2018.

[13] M. Hermelin, J. Y. Cho, and K. Nyberg, Multidimensional linear cryptanalysis of reduced round SERPENT, in *Australasian Conference on Information Security and Privacy*, pp. 203–215, Springer, 2008.

[14] X. Huang, L. Li, and J. Yang, IVLBC: An involutive lightweight block cipher for internet of things, IEEE Systems Journal, 2022.

[15] M. B. Ilter and A. A. Selçuk, A new MILP model for matrix multiplications with applications to KLEIN and PRINCE., in *SECRYPT*, pp. 420–427, 2021.

[16] M. B. İlter and A. A. Selçuk, MILP-aided cryptanalysis of the FUTURE block cipher, in *International Conference on Information Technology and Communications Security*, pp. 153–167, Springer, 2022.

[17] B. S. Kaliski and M. J. Robshaw, Linear cryptanalysis using multiple approximations, in *Advances in Cryptology—CRYPTO'94: 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings 14*, pp. 26–39, Springer, 1994.

[18] L. R. Knudsen, Truncated and higher order differentials, in *Fast Software Encryption: Second International Workshop Leuven, Belgium, December 14–16, 1994 Proceedings 2*, pp. 196–211, Springer, 1995.

[19] P. H. Kumar and R. Mageshvaran, Methods and solvers used for solving mixed integer linear programming and mixed nonlinear programming problems: A review, Int. J. Sci. Technol. Res, 9(1), pp. 1872–1882, 2020.

[20] X. Lai, Higher order derivatives and differential cryptanalysis, Communications and Cryptography: Two Sides of One Tapestry, pp. 227–233, 1994.

[21] L. Li, W. Wu, Y. Zheng, and L. Zhang, The relationship between the construction and solution of the milp models and applications, Cryptology ePrint Archive, 2019.

[22] Z. Li, W. Bi, X. Dong, and X. Wang, Improved conditional cube attacks on keccak keyed modes with MILP method, in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 99–127, Springer, 2017.

[23] M. Matsui, Linear cryptanalysis method for DES cipher, in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 386–397, Springer, 1993.

[24] N. Mouha, Q. Wang, D. Gu, and B. Preneel, Differential and linear cryptanalysis using mixed-integer linear programming, in *International Conference on Information Security and Cryptology*, pp. 57–76, Springer, 2011.

[25] F. A. Potra and S. J. Wright, Interior-point methods, Journal of computational and applied mathematics, 124(1-2), pp. 281–302, 2000.

[26] Y. Sasaki and Y. Todo, New algorithm for modeling S-box in MILP based differential and division trail search, in *International Conference for Information Technology and Communications*, pp. 150–165, Springer, 2017.

[27] Y. Sasaki and Y. Todo, New impossible differential search tool from design and cryptanalysis aspects, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 185–215, Springer, 2017.

[28] L. Sun, W. Wang, R. Liu, and M. Wang, MILP-aided bit-based division property for arx-based block cipher, Cryptology ePrint Archive, 2016.

[29] L. Sun, W. Wang, and M. Q. Wang, MILP-aided bit-based division property for primitives with non-bit-permutation linear layers, IET Information Security, 14(1), pp. 12–20, 2019.

[30] S. Sun, L. Hu, L. Song, Y. Xie, and P. Wang, Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks, in *International Conference on Information Security and Cryptology*, pp. 39–51, Springer, 2013.

[31] S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, L. Song, and K. Fu, Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties, IACR Cryptology ePrint Archive, 747, p. 2014, 2014.

[32] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers, in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 158–178, Springer, 2014.

[33] C. Tezcan, The improbable differential attack: Cryptanalysis of reduced round CLEFIA, in *Progress in Cryptology-INDOCRYPT 2010: 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings 11*, pp. 197–209, Springer, 2010.

[34] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 9.2)*, 2020, `https://www.sagemath.org`.

[35] R. J. Vanderbei et al., *Linear programming*, Springer, 2020.

[36] J. Yin, C. Ma, L. Lyu, J. Song, G. Zeng, C. Ma, and F. Wei, Improved cryptanalysis of an ISO standard lightweight block cipher with refined MILP modelling, in *International Conference on Information Security and Cryptology*, pp. 404–426, Springer, 2017.

[37] B. Zhu, X. Dong, and H. Yu, MILP-based differential attack on round-reduced GIFT, in *Cryptographers' Track at the RSA Conference*, pp. 372–390, Springer, 2019.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: İlter, Murat Burhan
Nationality: Turkish

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.,Cryptography	Middle East Technical University	2016
B.S., Mathematics	Middle East Technical University	2014
High School	Ankara Anatolian High School	2009

PUBLICATIONS

International Conference Publications

- Murat Burhan Ilter, Ali Aydin Selçuk: MILP-Aided Cryptanalysis of the FUTURE Block Cipher. SecITC 2022: 153-167
- Murat Burhan Ilter, Nese Koçak, Erkan Uslu, Oguz Yayla, Nergiz Yuca: On the Number of Arithmetic Operations in NTT-based Polynomial Multiplication in Kyber and Dilithium Cryptosystems. SIN 2021: 1-7
- Murat Burhan Ilter, Ali Aydin Selçuk: A New MILP Model for Matrix Multiplications with Applications to KLEIN and PRINCE. SECRYPT 2021: 420-427
- Berkin Aksoy, Yusuf Alper Bilgin, Murat Cenk, Murat Burhan Ilter, Nese Koçak, Yunus Emre Yılmaz. Analyzing NIST 2nd-round Lattice-based Post-

quantum KEM Algorithms, Information Security and Cryptology Conference, Ankara, Turkey, 2019

International Journal Publications

- Murat Burhan Ilter, Murat Cenk: Efficient Big Integer Multiplication in Cryptography, International Journal of Information Security Science, c. 6, sayı. 4, ss. 70-78, Ara. 2017