Turkish Journal of Electrical Engineering and Computer Sciences

Volume 32 | Number 1 Article 11

2-7-2024

# MILP modeling of matrix multiplication: cryptanalysis of KLEIN and PRINCE

Murat Burhan İLTER
mbilter@aselsan.com.tr

Ali Aydın SELÇUK
aliaydinselcuk@gmail.com

Recommended Citation
İLTER, Murat Burhan and SELÇUK, Ali Aydın (2024) "MILP modeling of matrix multiplication: cryptanalysis of KLEIN and PRINCE," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 32: No. 1, Article 11. https://doi.org/10.55730/1300-0632.4062
Available at: https://journals.tubitak.gov.tr/elektrik/vol32/iss1/11

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2024) 32: 183 – 197
© TÜBİTAK
doi:10.55730/1300-0632.4062

# MILP modeling of matrix multiplication: cryptanalysis of KLEIN and PRINCE

**Murat Burhan İLTER**[1,*], **Ali Aydın SELÇUK**[2]
[1]Aselsan Inc., Ankara, Turkiye
[2]Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkiye

**Received:** 18.09.2023 • **Accepted/Published Online:** 06.11.2023 • **Final Version:** 07.02.2024

**Abstract:** Mixed-integer linear programming (MILP) techniques are widely used in cryptanalysis, aiding in the discovery of optimal linear and differential characteristics. This paper delves into the analysis of block ciphers KLEIN and PRINCE using MILP, specifically calculating the best linear and differential characteristics for reduced-round versions. Both ciphers employ matrix multiplication in their diffusion layers, which we model using multiple XOR operations. To this end, we propose two novel MILP models for multiple XOR operations, which use fewer variables and constraints, proving to be more efficient than standard methods for XOR modeling.

For differential cryptanalysis, we identify characteristics with a probability of $2^{-59}$ for 7 rounds of KLEIN and a probability of $2^{-56}$ for 7 rounds of PRINCE. In linear cryptanalysis, we identify characteristics with a bias of $2^{-27}$ for 6 rounds of KLEIN and a bias of $2^{-29}$ for 7 rounds of PRINCE. These results establish the best single-key differential and linear distinguishers for these ciphers in the literature.

**Key words:** MILP, cryptanalysis, differential cryptanalysis, linear cryptanalysis, optimization

## 1. Introduction

Mixed-integer linear programming (MILP) techniques have found extensive application in cryptanalysis [1–4] since their introduction by Mouha et al. [5]. Although automated search techniques for optimal distinguishers existed well before the introduction of MILP to the area, they were specific to attacks and ciphers and required much-concentrated effort to develop. By contrast, MILP models for ciphers can be developed from existing building blocks, such as models for S-boxes and XOR operations, and can be deployed to search for optimal distinguishers of a given cipher in an effective manner.

MILP modeling commences by formulating an objective function based on the chosen cryptanalysis method. For example, to find the best linear characteristic, the objective function is designed to maximize linear biases. Next, the round operations of the cipher, such as S-box, permutation, XOR, multiplication, and addition, are modeled as constraints, with the inputs and outputs of these components defined as variables.

The efficacy of an MILP model is highly dependent on its complexity, including the number of constraints and variables involved. The quest for more efficient models capable of analyzing a higher number of cipher rounds has been a focal point in MILP-based studies in the literature.

[*] Correspondence: ilter.muratb@gmail.com

Several notable studies have addressed this challenge. Sasaki and Todo [6] introduced a method to represent an S-box with fewer constraints, Fu et al.[2] presented a methodology using a single constraint to model XOR operations, and Yin et al.[7] proposed XOR operation models with fewer variables.

In the context of SPN block ciphers utilizing (MDS) matrix multiplication operations over $GF(2^n)$ for diffusion, such as AES, expressing 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.

This paper explores alternative methods to model combined XOR operations, presenting two new MILP models. We apply these methods to the KLEIN [8] and PRINCE [9] block ciphers, modeling their differential and linear characteristics. Each attack is modeled in three alternative ways, and their solution times are compared. Ultimately, we derive the best single-key differential and linear characteristics for KLEIN and PRINCE available in the literature.

The paper is organized as follows: Related work is reviewed in Section 2. Section 3 details the MILP models used for XOR. Section 4 analyzes the differential and linear propagation of ciphers. Sections 5 and 6 provide details related to constructing the linear and differential MILP models of KLEIN and PRINCE, respectively. The paper concludes in Section 7.

## 2. Related work

The application of MILP techniques in block cipher cryptanalysis has evolved through several notable works. Mouha et al. [5] were the first to employ MILP to find a lower bound on the minimum number of active S-boxes in a differential and linear attack on AES and Enocoro ciphers.

Sun et al. [10] used MILP to find the minimum number of differentially and linearly active S-boxes for attacks on bit-oriented ciphers. They provided new related key and single-key characteristics for PRESENT-80.

Sun et al. [11] modeled SIMON, Serpent, LBlock, and DESL with a new S-box modeling approach. They studied the exact representation of an S-box via H-representation and logical condition modeling.

Sun et al. [12] proposed a method to find the best differential and linear characteristics, rather than lower bounds, using exact models for S-box probabilities. They studied PRESENT-128, DESL, LBlock, and SIMON48 ciphers and obtained improved results for single-key and related-key cryptanalysis.

Sasaki and Todo [6] introduced a way to represent an S-box exactly, stating that the representation of an S-box can be done with the minimum number of equations via the MILP approach.

Yin et al.[7] presented a model of XOR operations using fewer variables, focusing on the linear and differential cryptanalysis of the HIGHT algorithm. Fu et al.[2] analyzed the SPECK cipher with an efficient method of modeling the XOR operation.

## 3. Modeling the $n$-XOR Operation

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

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

### 3.1. 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 [6]:

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

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:

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

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

### 3.2. Model 1

In our method, we first calculate possible patterns for multiple XOR operations. We then use Sasaki and Todo's approach [6] to represent these patterns with the minimum number of constraints. 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:

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

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 [6] is applied to find the minimum set of inequalities to represent the XOR operation.

### 3.3. Model 2

Fu et al. [2] provided a method to model $y = x_1 \oplus x_2$ using a single constraint and a dummy variable $d_1$:

$$y + x_1 + x_2 = 2d_1,$$

where $y, x_1, x_2, d_1 \in \mathbb{F}_2$.

We generalize this approach to model $n$-XOR operation $y = x_0 \oplus x_1 \oplus \cdots \oplus x_n$ as,

$$\begin{cases} x_0 + x_1 + \cdots + x_n + y = (n+2)d_1 - \big(nd_2 + (n-2)d_3 \cdots + 2d_{(n/2)+1}\big), & n \text{ is even} \\ x_0 + x_1 + \cdots + x_n + y = (n+1)d_1 - \big((n-1)d_2 + (n-3)d_3 + \cdots + 2d_{(n-1)/2+1}\big), & n \text{ is odd.} \end{cases}$$

In Table 1, we compare the number of variables and constraints that are needed to represent the $n$-XOR operation in three alternative models.

**Table 1**. 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 |

## 4. Modeling differential and linear propagation

We provide an overview of the components of the MILP models developed in this section. Linear constraints with binary variables are used to represent matrix multiplication, permutation, and S-box operations over a finite field. The probability information in the difference distribution tables (DDT) or the linear approximation tables (LAT) is encoded into constraints in order 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 substitution and the matrix multiplication operations differ significantly between the two attack types. The three alternative XOR models explained in Section 3 are utilized to model the matrix multiplication operations over $GF(2^m)$. Other operations such as S-box substitutions and bit permutations are modeled using methods from the literature.

### 4.1. S-box

We used the method of Sun et al. [11] to represent the S-boxes: Binary variable $A$ represents the S-box's activity (where $A = 1$ indicates that it is active), and the activity of input and output bits, for a $4 \times 4$ S-box, are denoted by binary vectors $(x_1, x_2, x_3, x_4)$ and $(y_1, y_2, y_3, y_4)$, respectively, with the following constraints:

$$\begin{aligned} x_1 - A &\leq 0 \\ x_2 - A &\leq 0 \\ x_3 - A &\leq 0 \\ x_4 - A &\leq 0 \\ x_1 + x_2 + x_3 + x_4 - A &\geq 0 \\ 4x_1 + 4x_2 + 4x_3 + 4x_4 - y_1 - y_2 - y_3 - y_4 &\geq 0 \\ 4y_1 + 4y_2 + 4y_3 + 4y_4 - x_1 - x_2 - x_3 - x_4 &\geq 0 \end{aligned}$$

The differential behavior of an S-box is modeled using the method of Sun et al. [12]. Let, a $4 \times 4$ S-box has probability values in the DDT, $p = Pr[(x_0, x_1, x_2, x_3) \rightarrow (y_0, y_1, y_2, y_3)]$, which are exact powers of 2. Then the exact probability values can be encoded by two bits $(\pi_1, \pi_0)$ denoting the binary encoding of $-\log_2 p$ as:

$$\begin{aligned} (\pi_1, \pi_0) = (0,0) &\implies p = 1 \\ (\pi_1, \pi_0) = (0,1) &\implies p = 2^{-1} \\ (\pi_1, \pi_0) = (1,0) &\implies p = 2^{-2} \\ (\pi_1, \pi_0) = (1,1) &\implies p = 2^{-3}. \end{aligned}$$

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

$$v = (x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3, \pi_1, \pi_0).$$

For the set of vectors created, SageMath [13] is used to calculate its H-representation, which is a set of hyperplane equations whose intersection contains the given set of vectors. Possibly, some of the inequalities produced by the H-representation can be redundant. In order to remove these redundant inequalities, an MILP instance is built and solved, which yields a minimized set of constraints that represent the S-box along with its LAT or DDT. The details of the H-representation construction process can be found in [11] and [6].

### 4.2. Permutation

To represent the permutation $P$ in differential and linear cryptanalysis, new binary variables $y_i$'s are introduced. The input of the permutation is modeled by $x_i$ and the output of the permutation is modeled by $y_i = P(x_i)$ [10].

### 4.3. Matrix multiplication

We model the matrix multiplications by multiple XOR operations, according to the primitive representation of the matrix, using the method proposed by Sun et al. [14].

### 4.4. Objective function

The objective function is designed to find the characteristic with the maximum differential probability or linear bias. The related probability information is taken from the S-boxes and encoded as described in Section 4.1.

The differential probability to be maximized is taken as the product of differential probabilities, $\prod_i p_i$, over the active S-boxes. Hence, the objective function can be formulated as minimizing $\sum_i(\pi_{i,0} + 2\pi_{i,1})$, where $(\pi_{i,1}, \pi_{i,0})$ represents $-\log_2 p_i$ in binary, as described in Section 4.1.

The linear bias to be maximized is taken as the product of linear biases $b_i$ according to the Piling-up Lemma, $2^{n-1}\prod_i b_i$, for $n$ active S-boxes. Hence, the objective function can be formulated as minimizing $\sum_i(\pi_{i,0} + 2\pi_{i,1})$, where $(\pi_{i,1}, \pi_{i,0})$ represents $-\log_2 b_i$ in binary.

### 4.5. 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 were solved using the Gurobi optimizer [15] version 9.0.2. The H-representations were calculated using SageMath [13]. The reported timing results are CPU times in seconds.

The MILP models we constructed for KLEIN and PRINCE are available at `https://github.com/murat-ilter/Klein-Prince`.

## 5. MILP analysis of KLEIN

This section explains the MILP models we developed for linear and differential cryptanalysis of KLEIN. Using these models, we managed to find linear and differential characteristics for up to 6 and 7 rounds of the cipher, respectively.

### 5.1. KLEIN cipher

KLEIN [8] 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 $4 \times 4$ 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 $4 \times 4$ 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:** The nibbles are rotated according to the following permutation:

| 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 $4 \times 1$ 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.

### 5.2. Differential model

The MILP model for differential cryptanalysis of KLEIN is constructed along the following lines:

**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 4.1. 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[1] of $M$ is a binary matrix $M_{\mathcal{PR}}$ where the entries 1, 2, 3 in $M$ are replaced by

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

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

[1] 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 $M_{\mathcal{PR}}$.

We can represent the MDS matrix multiplication operation in MixNibbles by multiplication of a 32-bit input binary vector by $M_{\mathcal{PR}}$, which in turn can be modeled by multiple XOR operations as described in Section 3.

### 5.3. Linear model

The MILP model for linear cryptanalysis of KLEIN is constructed along the following lines, where the main difference from the differential model is in the 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 $32 \times 32$ binary matrix, as explained in Section 5.2. Let $y$ and $z$ be the $32 \times 1$ binary column vectors denoting the input and the output of a matrix multiplication operation in MixNibbles operation; i.e., $z = M_{\mathcal{PR}} y$.

Let $\beta^T$ be the 32-bit linear mask (row vector) indicating the active bits of $y$ in a linear approximation. We need to transform this linear mask of $y$ into a linear mask $\gamma^T$ for $z$, which can be calculated as:

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

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

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

The entries of $M^{-1}$ are 9, B, D, E, which are replaced by $\mathbf{9}, \mathbf{B}, \mathbf{D}$, and $\mathbf{E}$ 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 5.2, multiplication of a vector by the binary matrix $M_{\mathcal{PR}}^{-1}$is modeled with multiple XOR operations.

### 5.4. Results

We solved the MILP models for KLEIN using the Gurobi optimizer [15]. The best 2-round differential characteristic we found is 0000 00E0 0000 0000 $\rightarrow$ E03B 00DB 4000 E0E0 which can be decomposed as: 0000 00E0 0000 0000 $\xrightarrow{SN}$ 0000 0030 0000 0000 $\xrightarrow{RN,MN}$ 5060 3030 0000 0000 $\xrightarrow{SN}$ 2060 E0E0 0000 0000 $\xrightarrow{RN,MN}$ E03B 00DB 4000 E0E0 with probability $2^{-10}$, where $SN$, $RN$, and $MN$ represent the SubNibbles, RotateNibbles, and MixNibbles operations, respectively.

The best 3-round differential characteristic we found is, 0000 E000 10E0 0000 $\rightarrow$ 5090 9000 6030 3050 which can be decomposed as: 0000 E000 10E0 0000 $\xrightarrow{SN}$ 0000 3000 3030 0000 $\xrightarrow{RN,MN}$ 6050 0000 0000 0000 $\xrightarrow{SN}$ 4020 0000 0000 0000 $\xrightarrow{RN,MN}$ 0000 0000 60E0 E000 $\xrightarrow{SN}$ 0000 0000 6030 3000 $\xrightarrow{RN,MN}$ 5090 9000 6030 3050 with probability $2^{-17}$. [2] The best differential characteristics we found for 4–7 rounds are given in Table 2.

**Table 2**. The best differential characteristics of KLEIN we obtained for 4–7 rounds. The input difference for round $i$ is denoted by $\Delta X_{i-1}$.

| # rounds | 4 | 5 | 6 | 7 |
|---|---|---|---|---|
| $\Delta X_0$ | 0E0E 0000 0000 0D00 | 0000 0100 010E 0000 | 10E0 0000 0000 D000 | 0000 030E 000E 0000 |
| $\Delta X_1$ | 0000 0000 0000 0D0B | 0605 0000 0000 0000 | 0000 0000 0000 D0B0 | 0000 0B0E 0000 0000 |
| $\Delta X_2$ | 0000 0000 0E0E 0400 | 0000 0000 060E 0E00 | 0000 0000 E0E0 4000 | 0B0F 0604 0000 0000 |
| $\Delta X_3$ | 0006 0305 0603 0305 | 070F 0D02 0603 0305 | 00E0 7090 6030 3050 | 000E 020E 010B 060D |
| $\Delta X_4$ | 0613 0A1D 0203 090D | 0000 0001 0000 0100 | 0000 0010 90D0 0000 | 0101 0000 0000 0B0E |
| $\Delta X_5$ | | 0506 0303 0603 0305 | 00D0 B000 0000 0000 | 0000 0000 0101 0000 |
| $\Delta X_6$ | | | A050 50F0 2020 6040 | 0006 0305 0000 0000 |
| $\Delta X_7$ | | | | 0118 0519 0606 0A0C |
| probability | $2^{-32}$ | $2^{-42}$ | $2^{-48}$ | $2^{-59}$ |

The best 2-round linear characteristic we found is 0000 A000 0000 0000 $\rightarrow$ 2C64 9C34 CAA6 027E which can be decomposed as: 0000 A000 0000 0000 $\xrightarrow{SN}$ 0000 8000 0000 0000 $\xrightarrow{RN,MN}$ 70D0 B090 0000 0000 $\xrightarrow{SN}$ 4050 C020 0000 0000 $\xrightarrow{RN,MN}$ 2C64 9C34 CAA6 027E with bias $2^{-6}$.

The best 3-round linear characteristic we found is, 0004 0000 0000 0808 $\rightarrow$ A662 25E7 8785 8781 which can be decomposed as: 0004 0000 0000 0808 $\xrightarrow{SN}$ 0007 0000 0000 0A0A $\xrightarrow{RN,MN}$ 0000 0000 090E 0000 $\xrightarrow{SN}$

[2] Correctness of the models and of the differential and linear probabilities found have been verified for smaller numbers of rounds by statistical sampling.

0000 0000 0201 0000 $\xrightarrow{RN,MN}$ 0303 0003 0000 0000 $\xrightarrow{SN}$ 0206 0006 0000 0000 $\xrightarrow{RN,MN}$ A662 25E7 8785 8781 with bias $2^{-9}$. The best linear characteristics we found for 4–6 rounds are given in Table 3.

**Table 3**. The best linear characteristic of KLEIN we obtained for 4–6 rounds. The input linear mask for round $i$ is denoted by $\alpha_{i-1}$.

| # rounds | 4 | 5 | 6 |
|---|---|---|---|
| $\alpha_0$ | 0000 0808 0008 0000 | 0000 2010 8000 0000 | 0000 060A 0300 0000 |
| $\alpha_1$ | 090E 0000 0000 0000 | E0E0 0000 0000 0000 | 0404 0000 0000 0000 |
| $\alpha_2$ | 0000 0000 0303 0003 | 0000 0000 80C0 E040 | 0000 0000 0201 0506 |
| $\alpha_3$ | 0404 0004 0F05 080C | 1030 7070 7030 5070 | 0506 0501 0007 0707 |
| $\alpha_4$ | E0A8 622C AC68 2FED | 0000 8000 8000 A020 | 0D09 0000 0000 0400 |
| $\alpha_5$ |  | 1692 2E7A 2E9A D612 | 0000 0000 0700 0400 |
| $\alpha_6$ |  |  | EBA9 672D 8284 8687 |
| bias | $2^{-17}$ | $2^{-24}$ | $2^{-27}$ |

To the best of our knowledge, these results provide the first single-key differential and linear characteristics of KLEIN in the literature.

We employed the alternative XOR models described in Section 3 to model the matrix multiplication operation in KLEIN and compared their efficiency. The solution complexity of the models, including the number of constraints, the number of variables, and the execution time (in CPU seconds), are given in Tables 4 and 5 for the differential and linear models, respectively.[3] The alternative XOR models employed in MILP yielded the same probabilities throughout the experiments. 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**. Complexity of the alternative XOR models for differential MILP solutions of KLEIN.

|  | Standard XOR |  |  | Model 1 |  |  | Model 2 |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Round | #V. | #C. | T (s.) | # V. | # C. | T (s.) | # V. | # C. | T (s.) |
| 2 | 592 | 2113 | 14 | 352 | 5249 | 14 | 568 | 961 | 9 |
| 3 | 1008 | 3777 | 30,373 | 528 | 10,049 | 15,074 | 960 | 1473 | 2322 |
| 4 | 1424 | 5444 | 136,556 | 704 | 14,852 | 50,582 | 1352 | 1988 | 77,279 |
| 5 | 1840 | 7109 | 881,567 | 880 | 19,653 | 382,301 | 1744 | 2501 | 297,421 |
| 6 (*) | 2256 | 8769 | >1,000,000 | 1056 | 24,449 | >1,000,000 | 2136 | 3013 | >1,000,000 |
| 7 (*) | 2672 | 10,439 | >1,000,000 | 1232 | 29,255 | >1,000,000 | 2528 | 3527 | >1,000,000 |

**Table 5**. Complexity of the alternative XOR models for linear MILP solutions of KLEIN.

|  | Standard XOR |  |  | Model 2 |  |  |
|---|---|---|---|---|---|---|
| Round | #V. | #C. | T (s.) | # V. | # C. | T (s.) |
| 2 | 1168 | 4801 | 564 | 856 | 1345 | 67 |
| 3 | 2160 | 8964 | 107,040 | 1536 | 2052 | 17,320 |
| 4 | 3152 | 13,124 | >1,000,000 | 2216 | 2756 | 448,893 |
| 5 (*) | 4144 | 17,285 | >1,000,000 | 2896 | 3461 | >1,000,000 |
| 6 (*) | 5136 | 21,445 | >1,000,000 | 3576 | 4165 | >1,000,000 |

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

## 6. MILP model for PRINCE

This section explains the MILP models we developed for the linear and differential cryptanalysis of PRINCE. We were able to identify the best single-key linear and differential characteristics for up to 7 rounds of the cipher, after which point the probabilities (or, biases) of characteristics become too small for an effective attack.

### 6.1. PRINCE cipher

PRINCE [9] 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 $4 \times 4$ 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 $4 \times 4$ S-box:

| 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 shift row is identical to the one in AES but operates on 4-bit nibbles instead of bytes. The matrix multiplication operation is based on a $64 \times 64$ binary matrix $M'$ constructed from a number of submatrices, as explained below:

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

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

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

### 6.2. Differential model

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

**S-box Layer:** In the DDT of PRINCE's S-box, there are three nonzero probabilities: $1$, $2^{-2}$, and $2^{-3}$. We encoded these probabilities with the corresponding differential patterns, as described in Section 4.1. Next, we computed the H-representation with SageMath, obtaining 1975 inequalities. Applying Sasaki and Todo's reduction method on it, we obtained 22 inequalities representing the DDT of PRINCE's S-box with the related probability information.

**Linear Layer:** Each row of the 64×64 matrix $M'$ contains exactly three 1s. Therefore, the multiplication of an input vector by each row of the matrix consists of an XOR of three input bits and can be modeled as a 2-XOR operation. The equations to model these 2-XOR operations are added as constraints to the MILP model, as described in Section 3. Shift row operation is carried out as given in Section 4.2.

### 6.3. Linear Model

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.

### 6.4. Results

We solved the MILP models for PRINCE using the Gurobi optimizer [15]. The best 2-round differential characteristic we found is 000C 0000 0000 0000 → 0880 0080 8808 8088 which can be decomposed as: 000C 0000 0000 0000 $\xrightarrow{Sl}$ 0001 0000 0000 0000 $\xrightarrow{Ll}$ 0000 0001 0010 0100 $\xrightarrow{Sl}$ 0000 0008 0080 0800 $\xrightarrow{Ll}$ 0880 0080 8808 8088 with probability $2^{-8}$, where $Sl$ and $Ll$ represent the S-box layer and Linear layer, respectively.

The best 3-round differential characteristic we found is 0000 0000 0000 1C01 → 8001 0118 1100 0801 which can be decomposed as: 0000 0000 0000 1C01 $\xrightarrow{Sl}$ 0000 0000 0000 1101 $\xrightarrow{Ll}$ 0000 0000 0100 0000 $\xrightarrow{Sl}$ 0000 0000 0100 0000 $\xrightarrow{Ll}$ 0010 0000 1000 0001 $\xrightarrow{Sl}$ 0080 0000 1000 0001 $\xrightarrow{Ll}$ 8001 0118 1100 0801 with probability $2^{-14}$. The best differential characteristics for 4–7 rounds are given in Table 6.

**Table 6**. The best differential characteristics of PRINCE we obtained for 4–7 rounds. The input difference for round $i$ is denoted by $\Delta X_{i-1}$.

| # rounds | 4 | 5 | 6 | 7 |
|---|---|---|---|---|
| $\Delta X_0$ | 0000 1101 0000 0000 | 0000 0011 0C10 0000 | 001C 1100 0000 0000 | 0041 C800 0000 0000 |
| $\Delta X_1$ | 0000 1000 0000 0000 | 0000 1100 1001 0000 | 1100 0000 0000 0110 | 1100 0000 0000 0110 |
| $\Delta X_2$ | 0800 8000 0000 0080 | 0110 0000 0000 0011 | 0000 0011 0110 0000 | 0000 0011 0110 0000 |
| $\Delta X_3$ | 0400 0044 0444 4440 | 0000 0088 0880 0000 | 0000 1100 1001 0000 | 0000 1100 1001 0000 |
| $\Delta X_4$ | 3012 1203 0023 1120 | 0440 0000 0000 0044 | 0110 0000 0000 0011 | 0110 0000 0000 0011 |
| $\Delta X_5$ |  | 2002 0000 0110 3300 | 0000 0080 0810 0100 | 0000 0088 0880 0000 |
| $\Delta X_6$ |  |  | 0000 4404 4551 1101 | 0000 0440 0044 0000 |
| $\Delta X_7$ |  |  |  | 9A3B 3B9A 9A2B 9A3B |
| probability | $2^{-32}$ | $2^{-40}$ | $2^{-48}$ | $2^{-56}$ |

The best 2-round linear characteristic we found is 0C00 0000 0000 0000 → 1011 0100 0110 1101 which can be decomposed as: 0C00 0000 0000 0000 $\xrightarrow{Sl}$ 0800 0000 0000 0000 $\xrightarrow{Ll}$ 8000 0000 0080 0800 $\xrightarrow{Sl}$ 1000 0000 0010 0100 $\xrightarrow{Ll}$ 1011 0100 0110 1101 with bias $2^{-5}$.

The best 3-round linear characteristic we found is, 1044 0000 0000 0000 → 4440 4404 4004 0400 which can be decomposed as: 1044 0000 0000 0000 $\xrightarrow{Sl}$ 2022 0000 0000 0000 $\xrightarrow{Ll}$ 0000 0000 0000 0200 $\xrightarrow{Sl}$ 0000 0000 0000 0400 $\xrightarrow{Ll}$ 0004 0040 0400 0000 $\xrightarrow{Sl}$ 0004 0040 0400 0000 $\xrightarrow{Ll}$ 4440 4404 4004 0400 with bias $2^{-8}$. The best linear characteristics for 4–7 rounds are expressed in Table 7.

The best previously known single-key differential characteristic on PRINCE was for 6 rounds of the cipher with a probability of $2^{-62}$ [16]. We improved the analysis to 7 rounds with a probability of $2^{-56}$ using the MILP approach. As for linear cryptanalysis, our results provide the first known single-key linear distinguishers. The minimum number of active S-boxes increases dramatically after the 8th round and these attacks become

**Table 7**. The best linear characteristic of PRINCE we obtained for 4–7 rounds. The input linear mask for round $i$ is denoted by $\alpha_{i-1}$.

| # rounds | 4 | 5 | 6 | 7 |
|---|---|---|---|---|
| $\alpha_0$ | 0000 0088 0000 2004 | 0220 2002 0000 0000 | 0000 0000 0204 0202 | 0440 4004 0000 0000 |
| $\alpha_1$ | 0000 1040 0401 0000 | 4400 4004 0000 0000 | 0000 0440 0000 4004 | 2002 0020 0000 0003 |
| $\alpha_2$ | 0220 0000 2002 0000 | 4200 0000 0000 0420 | 0000 2040 0402 0000 | 2400 0000 0000 0240 |
| $\alpha_3$ | 4000 0404 4000 0000 | 4004 0000 0000 4400 | 0000 4400 0000 0044 | 2002 0000 0000 2200 |
| $\alpha_4$ | 4440 0404 0044 0004 | 0000 0002 0420 4000 | 0204 2040 0000 0000 | 0000 0000 0220 2200 |
| $\alpha_5$ |  | 0044 4040 0004 4044 | 4000 4000 0040 0040 | 0000 0000 4200 2004 |
| $\alpha_6$ |  |  | 4044 0444 4404 4440 | 0000 0000 2002 0220 |
| $\alpha_7$ |  |  |  | 4044 0044 4044 0000 |
| bias | $2^{-17}$ | $2^{-21}$ | $2^{-25}$ | $2^{-29}$ |

infeasible [17].

We utilized the alternative XOR models described in Section 3 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 Tables 8 and 9 for the differential and linear models, respectively.

**Table 8**. Complexity of the alternative XOR models for differential 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 | 1475 | 3 | 416 | 1475 | 2 | 544 | 1027 | 1 |
| 3 | 784 | 2500 | 1302 | 656 | 2500 | 464 | 912 | 1604 | 206 |
| 4 | 1088 | 3524 | 159,462 | 896 | 3524 | 15,368 | 1280 | 2180 | 38,705 |
| 5 | 1392 | 4548 | 177,410 | 1136 | 4548 | 290,543 | 1648 | 2756 | 141,780 |
| 6 | 1696 | 5575 | 330,389 | 1376 | 5575 | 235,481 | 2016 | 3335 | 575,157 |
| 7 | 1937 | 6536 | 431,921 | 1552 | 6536 | 303,585 | 2320 | 3848 | 365,911 |

**Table 9**. 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 | 27,592 | 896 | 4293 | 24,513 | 1280 | 2949 | 91,409 |
| 5 | 1392 | 5510 | 21,610 | 1136 | 5510 | 68,815 | 1648 | 3718 | 14,601 |
| 6 | 1696 | 6727 | 23,807 | 1376 | 6727 | 79,587 | 2016 | 4487 | 25,981 |
| 7 | 1936 | 7880 | 156,500 | 1552 | 7880 | 47,481 | 2320 | 5192 | 74,070 |

## 7. Conclusions

In this paper, we proposed two alternative MILP modeling methods to model equations of multiple XOR operations. Model 1 works with fewer variables, and Model 2 works with fewer constraints. We used these new $n$-XOR models to model matrix multiplication over $GF(2^m)$. The standard MILP model of XOR operations is also used as the base case for comparisons.

Using these three models, we constructed MILP models for the PRINCE and KLEIN ciphers. MILP models in this study allowed us to determine the best single-key differential and linear characteristics for various round numbers. The best single-key differential characteristics of probability $2^{-59}$ and $2^{-56}$ were found for 7 rounds of KLEIN and PRINCE, respectively; and the best single-key linear characteristics for 6 rounds of KLEIN and 7 rounds of PRINCE were found with biases $2^{-27}$ and $2^{-29}$, respectively.

The proposed models are quite general and can be applied to other ciphers that use matrix multiplication operations over finite fields $GF(2^m)$ in the diffusion layer. Using these models, improved results on differential and linear properties of similar ciphers can be obtained.

## Acknowledgment

A preliminary version of this work, with a much limited and preliminary set of experimental results for differential cryptanalysis, and without any study of linear cryptanalysis, was presented at the SECRPYT'21 conference [17].

## References

[1] Zhu B, Dong X, Yu H. MILP-Based differential attack on round-reduced GIFT. In Topics in Cryptology–CT-RSA; San Francisco, CA, USA; 2019; pp. 372-390.

[2] Fu K, Wang M, Guo Y, Sun S, Hu L. MILP-based automatic search algorithms for differential and linear trails for SPECK. In Fast Software Encryption: 23rd International Conference, FSE 2016; Bochum, Germany; 2016; pp. 268-288.

[3] Sasaki Y, Todo Y. New impossible differential search tool from design and cryptanalysis aspects revealing structural properties of several ciphers. In Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques; Paris, France; pp. 185-215.

[4] Li Z, Bi W, Dong X, Wang X. Improved conditional cube attacks on Keccak keyed modes with MILP method. In Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security; Hong Kong, China; pp. 99-127.

[5] Mouha N, Wang Q, Gu D, Preneel B. Differential and linear cryptanalysis using mixed-integer linear programming. Information Security and Cryptology: 7th International Conference, Inscrypt 2011; Beijing, China; 2011. pp. 57-76.

[6] Sasaki Y, Todo Y. New Algorithm for Modeling S-box in MILP Based Differential and Division Trail Search. In Innovative Security Solutions for Information Technology and Communications: 10th International Conference, SecITC 2017; Bucharest, Romania; pp. 150-165.

[7] Yin J, Ma C, Lyu L, Song J, Zeng G et al. Improved cryptanalysis of an ISO standard lightweight block cipher with refined MILP modelling. In Information Security and Cryptology: 13th International Conference, Inscrypt 2017; Xi'an, China; pp. 404-426.

[8] Gong Z, Nikova S, Law YW. KLEIN: A new family of lightweight block ciphers. In RFID. Security and Privacy: 7th International Workshop, RFIDSec 2011; Amherst, USA; pp. 1-18.

[9] Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knezevic M et al. PRINCE - A low-latency block cipher for pervasive computing applications. In Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security; Beijing, China; pp. 208-225.

[10] Sun S, Hu L, Song L, Xie Y, Wang P. Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks. In Information Security and Cryptology: 9th International Conference, Inscrypt 2013; Guangzhou, China; pp. 39-51.

[11] Sun S, Hu L, Wang P, Qiao K, Ma X et al. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In Advances in Cryptology–ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security; Kaoshiung, Taiwan, ROC; pp. 158-178.

[12] Sun S, Hu L, Wang M, Wang P, Qiao K et al. Towards Finding the Best Characteristics of Some Bit-oriented Block Ciphers and Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties. Cryptology ePrint Archive, Report 747. 2014: pp. 1–31.

[13] Stein WA et al. Sage Mathematics Software (Version 9.1), The Sage Development Team. 2021.

[14] Sun L, Wang W, Wang MQ. MILP-aided bit-based division property for primitives with non-bit-permutation linear layers. IET Information Security. 2020;14 (1): 12–20.

[15] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2021.

[16] Ankele R, Kölbl S. Mind the Gap - A Closer Look at the Security of Block Ciphers against Differential Cryptanalysis. In Selected Areas in Cryptography–SAC 2018: 25th International Conference; Calgary, AB, Canada; pp. 163-190.

[17] İlter MB, Selçuk A. A new MILP model for matrix multiplications with applications to KLEIN and PRINCE. In Proceedings of the 18th International Conference on Security and Cryptography-SECRYPT21. pp. 420-427.