# MILP-aided Cryptanalysis of the FUTURE Block Cipher

Murat Burhan İlter[1,2] and Ali Aydın Selçuk[3]

[1] Inst. of Applied Mathematics, Middle East Technical University, Ankara, Turkey
[2] Aselsan Inc., Ankara, Turkey
ilter.muratb@gmail.com
[3] Dept. of Computer Eng., TOBB Univ. of Economics and Tech., Ankara, Turkey
aselcuk@etu.edu.tr

**Abstract.** FUTURE is a recently proposed, lightweight block cipher. It has an AES-like, SP-based, 10-round encryption function, where, unlike most other lightweight constructions, the diffusion layer is based on an MDS matrix. Despite its relative complexity, it has a remarkable hardware performance due to careful design decisions.

In this paper, we conducted a MILP-based analysis of the cipher, where we incorporated exact probabilities rather than just the number of active S-boxes into the model. Through the MILP analysis, we were able to find differential and linear distinguishers for up to 5 rounds of FUTURE, extending the known distinguishers of the cipher by one round.

**Keywords:** FUTURE · MILP · differential cryptanalysis · linear cryptanalysis

## 1 Introduction

FUTURE is a new 64-bit lightweight block cipher, recently proposed by Gupta et al. [5]. It is a 10-round, AES-like cipher that operates on 4-bit nibbles rather than bytes. FUTURE is interesting as being one of the few lightweight cipher designs where the diffusion layer is based on an MDS matrix. It is also remarkable for the lightweight construction of its MDS matrix and the S-box: Designers of FUTURE obtained the MDS matrix to have a minimal cost by multiplying four sparse matrices, and obtained the S-box by the composition of four low-hardware-cost S-boxes. The authors benchmarked hardware implementations on FPGA and ASIC and compared FUTURE to several well-known lightweight ciphers in the literature with respect to size, critical path, and throughput. FUTURE ended up giving the best results among the compared algorithms in many respects [5].

Mixed integer linear programming (MILP) is a well-known optimization method to find the optimal solution of a linear objective function, subject to a given set of linear constraints. It has found widespread application in security analysis of ciphers and hash functions over the past decade [8,12,14]. By encoding the internal structure of a cipher as a set of linear constraints, and the characteristic to be found as the objective function, a search for optimal characteristics can be carried out using general tools, such as the Gurobi optimizer [6]. MILP analyses have been particularly effective for lightweight ciphers where the models are more tractable, and the exact optimal characteristics can be found [8,12,14,13,10,9]. For general, non-lightweight ciphers such as AES, MILP has been used to prove differential and linear lower bounds [8,12].

A preliminary MILP analysis of FUTURE was given in the design paper [5]. The authors solved MILP models to find the minimum number of active S-boxes in a characteristic. They concluded that 4-round differential and linear distinguishers were possible, but five or more rounds of FUTURE should be safe from such distinguishers.

In this paper, we conduct a more detailed MILP analysis of FUTURE, where we incorporate the exact differential and linear probabilities of the cipher into the MILP model. We work with exact probabilities rather than the number of active S-boxes, with an increased complexity of the model. After applying several techniques to increase the effectiveness of the MILP search, our analysis obtains 5-round differential and linear distinguishers for the cipher.

The organization of the rest of this paper is as follows: Application of MILP techniques in cryptography is surveyed in Section 2. FUTURE is described in Section 3. The construction details of our MILP models are described in Section 4. The MILP models for differential and linear cryptanalysis are given in Section 5 and Section 6, respectively. The paper is concluded in Section 7.

## 2 Related Work

Mouha et al. [8] proposed using MILP techniques to find lower bounds on the number of active S-boxes in cryptanalysis of word-oriented ciphers. They investigated linear and differential cryptanalysis of the AES and Enocoro ciphers by this technique and obtained the desired lower bounds.

Sun et al. [12] improved Mouha et al.'s technique to find the exact minimum number of active S-boxes for bit-oriented block ciphers. They modeled PRESENT-80 by MILP for single-key and related-key differential analysis.

Sun et al. [14] gave the first MILP-based analysis that used H-representation and logical condition modeling to obtain an exact representation of an S-box. They analyzed the ciphers SIMON, Serpent, LBlock, and DESL, and obtained some significant results of differential cryptanalysis and related key attacks on these ciphers.

Sun et al. [13] improved this technique further to incorporate the probability (or, bias) information into the MILP model and to find the optimal characteristic with the highest probability (or, bias). In this work, the probability information of possible linear and differential patterns was encoded within an S-box representation. They studied SIMON48, LBlock, DESL, and PRESENT-128 ciphers and improved results for linear, differential, and related-key cryptanalysis.

Sasaki and Todo [9] further improved the technique of [13] by adding a MILP-based optimization phase to the algorithm to obtain a minimized representation of S-boxes with smallest number of constraints.

MILP modeling has more recently been applied to different cryptanalysis methods, such as the cube attack [4], and impossible differential cryptanalysis [9].

Different types of ciphers, besides bit-oriented, lightweight ciphers, have also been analyzed by MILP: Sun et al. [10] applied the technique to analyze ARX-based ciphers. Sun et al. [11] showed how to model differential propagation over an MDS matrix multiplication by MILP. Abdelkhalek et al. [1] and Boura and Coggia [2] modeled ciphers with 8 × 8 S-boxes by MILP.

Efficiency improvements on various components of MILP models have also been studied in the literature. Fu et al. [3] provided a way to reduce the number of constraints needed to model an XOR operation. Yin et al .[16] and Ilter and Selcuk [7] proposed more efficient ways to model multiple combined XOR operations.

# 3 FUTURE

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. The FUTURE block size is 64 bits, and the key length is 128 bits.

**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 $0 \leq i \leq 15$:

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

The round function is presented in Figure 1.

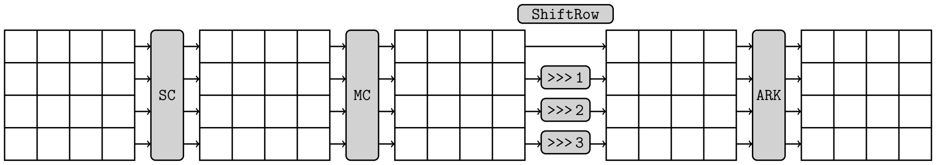

Fig. 1: Round function of FUTURE

**SubCell** The 4 × 4 S-box of FUTURE which is a composition of 4 different lightweight S-boxes is given in Table 1.

Table 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 |

**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 $X \leftarrow MX$:

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

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

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

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

## 4 Construction of MILP Models

The MILP approach has been widely used in cryptanalysis since Mouha et al. [8] introduced the technique. The main idea is to find the optimal solution of an objective function (e.g., the minimum number of active S-boxes or the maximum differential probability) with respect to certain constraints, according to the MILP model of a given cipher. The technique was first used to find the minimum number of active S-boxes in a characteristic [8,12]. It was later refined by Sun et al. [14] to find the optimal characteristic with the maximum differential probability or the maximum linear bias. In this paper, our objective function will be to maximize the differential probability (or linear bias) of a characteristic.

We need to model cipher components as constraints to construct a MILP model to analyze differential and linear characteristics. Therefore, the S-box, permutation, and matrix multiplication over a finite field are represented by linear inequalities with binary variables. This section provides an overview of the MILP modeling of block cipher components, such as the nibble-oriented S-box, MDS matrix multiplication, and permutation.

The number of variables and constraints in a MILP model affects its solution time dramatically. Hence, efficient cipher component modeling is essential to obtain a shorter solution time. With this aim in mind, we modeled the XOR operations by generalizing the idea of Fu et al. [3].

Gurobi optimizer [6] v.9.5.2 is used to solve the MILP models, and SageMath [15] is used to calculate the H-representations. The experiments are carried out on a 2.3 GHz Quad-Core Intel Core i5 processor with 8 GB RAM.

### 4.1 S-box

Lower bounds for the minimum number of active S-boxes can be obtained via using the branch number of S-boxes, as Mouha et al. [8] showed. Sun et al. [14] provided a method in which S-box is modeled to find exact solutions.

Let a $4 \times 4$ 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.

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

Furthermore, if exact probability bounds are sought, the Difference Distribution Table (DDT) or the Linear Approximation Table (LAT) should be included in the model. Sun et al. [14] proposed a greedy approach to model the DDT (LAT), which was later improved by Sasaki and Todo [9]. Our model is based on Sasaki and Todo's approach:

Suppose we want to model a $4 \times 4$ 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 $(\pi_0, \pi_1)$, denoting the binary encoding of $-\log_2 p$ as:

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

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

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

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

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

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.

Some of the inequalities calculated in H-representation may possibly be redundant. In order to eliminate the redundant inequalities, a MILP instance is built and solved. The solution provides a minimized set of constraints that represents the S-box with its DDT (or, LAT). Further details of the H-representation construction can be found in [14] and [9].

### 4.2 Permutation

Let the input of the permutation $\Pi$ 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.

### 4.3 MDS Matrix Multiplication

Mouha et al. [8] modeled matrix multiplication with the branch number of the linear transformation. The solution obtained by this method yields lower bounds on the number of active S-boxes.

MDS matrix multiplication can be carried out by shift and XOR operations over the base field. Sun et al. [11] provided a method to model matrix multiplication with binary XOR operation. This representation can be used to model differential propagation. In order to model linear propagation, we need a different representation which is discussed in Section 6.1.

### 4.4 XOR Operation

There are several different ways to model a binary XOR operation by MILP in the literature. Mouha et al. [8] provided a method that requires 4 constraints and 3 variables to model a 1-XOR operation, i.e., $c = a \oplus b$, where $a, b, c, d_1 \in \{0, 1\}$, as follows:

$$\begin{aligned} a+b+c &\geq 2d_1 \\ d_1 &\geq a \\ d_1 &\geq b \\ d_1 &\geq c \end{aligned}$$

The operation $d = a \oplus b \oplus c$, where $a, b, c, d \in \{0, 1\}$, is called a 2-XOR operation. It can be modeled via Mouha's approach with 8 constraints and 5 variables. Alternatively, Yin et al. [16] provided a method to model 2-XOR operation, which requires the following 8 constraints and 4 variables:

$$\begin{aligned} a+b-c+d &\geq 0 \\ a+b+c-d &\geq 0 \\ -a+b+c+d &\geq 0 \\ a-b+c+d &\geq 0 \\ -a-b+c-d &\geq -2 \\ a-b-c-d &\geq -2 \\ -a+b-c-d &\geq -2 \\ -a-b-c+d &\geq -2 \end{aligned}$$

Dummy variables are not used in this approach.

Fu et al. [3] implemented a method to model a 1-XOR operation with a single constraint as follows:

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

where $a, b, c, d_1 \in \{0, 1\}$. In this work, we extend this approach to the $n$-XOR case. In Table 2, constraints are given to model XOR operations up to 5-XOR.

Table 2: 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 $n$:

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

### 4.5 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. The MILP analysis in the original FUTURE paper [5] focused on the number of active S-boxes. We chose to work with the exact probabilities instead.

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 $\sum_i(\pi_{i,0} + 2\pi_{i,1})$.

## 5 Differential Cryptanalysis of FUTURE

In this section, we describe the details of the MILP model constructed for differential cryptanalysis of FUTURE and how it is implemented in practice. [4]

### 5.1 Differential MILP Model Construction

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 4.1, we encoded each input, output, and probability information as a vector, and computed the H-representation using SAGE. 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.

[4] https://github.com/murat-ilter/future-bc

**MixColumn** In order to represent the MDS matrix, the primitive matrix representation provided by [11] is utilized for differential propagation. FUTURE's MDS matrix $M$ contains the field elements **1**, **2**, **3**, **8**, **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:

$$\mathbf{1} = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix} \quad \mathbf{2} = \begin{pmatrix} 0&1&0&0\\ 0&0&1&0\\ 1&0&0&1\\ 1&0&0&0 \end{pmatrix} \quad \mathbf{3} = \begin{pmatrix} 1&1&0&0\\ 0&1&1&0\\ 1&0&1&1\\ 1&0&0&1 \end{pmatrix} \quad \mathbf{8} = \begin{pmatrix} 1&0&0&1\\ 1&1&0&0\\ 0&1&1&0\\ 0&0&1&0 \end{pmatrix} \quad \mathbf{9} = \begin{pmatrix} 0&0&0&1\\ 1&0&0&0\\ 0&1&0&0\\ 0&0&1&1 \end{pmatrix}$$

Let $M_{\mathcal{PR}}$ denote the $16 \times 16$ binary matrix which is the primitive representation of $M$ over $GF(2)$, obtained by replacing the field elements in $M$ by the $4 \times 4$ binary matrices given above. For the state matrices $Y$ and $Z$ where $Z = MY$, let $Y_{\mathcal{B}}$ and $Z_{\mathcal{B}}$ denote the $16 \times 4$ 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 $GF(2)$. 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 $M_{\mathcal{PR}}$.

For instance, consider the 16-bit column vectors $y = (y_0, y_1, \cdots, y_{15})^T$ and $z = (z_0, z_1, \cdots, z_{15})^T$, where $z = M_{\mathcal{PR}}\, y$. The following 16 equations are needed to represent the differential propagation:

$$\begin{aligned}
y_0 + y_3 + y_7 + y_8 + y_{12} + y_{15} + z_0 - 6d_0 + 4d_1 + 2d_2 &= 0\\
y_0 + y_1 + y_4 + y_9 + y_{12} + y_{13} + z_1 - 6d_3 + 4d_4 + 2d_5 &= 0\\
y_1 + y_2 + y_5 + y_{10} + y_{13} + y_{14} + z_2 - 6d_6 + 4d_7 + 2d_8 &= 0\\
y_2 + y_6 + y_7 + y_{11} + y_{14} + z_3 - 6d_9 + 4d_{10} + 2d_{11} &= 0\\
y_0 + y_1 + y_5 + y_{11} + y_{15} + z_4 - 6d_{12} + 4d_{13} + 2d_{14} &= 0\\
y_1 + y_2 + y_6 + y_8 + y_{12} + z_5 - 6d_{15} + 4d_{16} + 2d_{17} &= 0\\
y_0 + y_2 + y_3 + y_4 + y_7 + y_9 + y_{13} + z_6 - 8d_{18} + 6d_{19} + 4d_{20} + 2d_{21} &= 0\\
y_0 + y_3 + y_4 + y_{10} + y_{11} + y_{14} + y_{15} + z_7 - 8d_{22} + 6d_{23} + 4d_{24} + 2d_{25} &= 0\\
y_1 + y_4 + y_5 + y_8 + y_{11} + y_{15} + z_8 - 6d_{26} + 4d_{27} + 2d_{28} &= 0\\
y_2 + y_5 + y_6 + y_8 + y_9 + y_{12} + z_9 - 6d_{29} + 4d_{30} + 2d_{31} &= 0\\
y_0 + y_3 + y_4 + y_6 + y_7 + y_9 + y_{10} + y_{13} + z_{10} - 8d_{32} + 6d_{33} + 4d_{34} + 2d_{35} &= 0\\
y_0 + y_4 + y_7 + y_{10} + y_{14} + y_{15} + z_{11} - 6d_{36} + 4d_{37} + 2d_{38} &= 0\\
y_3 + y_7 + y_8 + y_{11} + y_{12} + z_{12} - 6d_{39} + 4d_{40} + 2d_{41} &= 0\\
y_0 + y_4 + y_8 + y_9 + y_{13} + z_{13} - 6d_{42} + 4d_{43} + 2d_{44} &= 0\\
y_1 + y_5 + y_9 + y_{10} + y_{14} + z_{14} - 6d_{45} + 4d_{46} + 2d_{47} &= 0\\
y_2 + y_3 + y_6 + y_7 + y_{10} + y_{15} + z_{15} - 6d_{48} + 4d_{49} + 2d_{50} &= 0
\end{aligned}$$

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 a single-key differential cryptanalysis, there is no need to model the XOR operation with the round key.

### 5.2 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. [17], 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 3-round differential characteristic is obtained by adding the following three constraints:

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

where $A_j^i$ stands for the $j$th S-box in the $i$th round. These extra constraints are used to determine the number of active S-boxes in each round, such as 4-1-4 in this example search strategy. In Table 3, the best differential probabilities are given with respect to the search strategies we tried.

### 5.3 Results

The differential characteristic probabilities up to five rounds are given in Table 3.

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 is given in Table 4.

Table 3: 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 | $2^{-61}$ | | |

Table 4: 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 Linear Cryptanalysis 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.

### 6.1 Linear MILP Model Construction

**SubCell** We calculated the LAT for FUTURE's S-box, and, as described in Section 4.1, we encoded each input, output, and bias (in absolute value) information as a vector. Then we computed the H-representation using SAGE. 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 $16 \times 16$ binary matrix which is the primitive representation of $M$ over $GF(2)$, as explained in Section 5.1, and let $Y_{\mathcal{B}}$ and $Z_{\mathcal{B}}$ be the $16 \times 4$ 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 $GF(2)$. 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 $\beta^T$ be the 16-bit row vector (linear mask) indicating the active bits of $y$ in a linear approximation. Then, the corresponding linear mask $\gamma^T$ on $z$ can be calculated as follows:

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

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

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

**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 4.2.

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

### 6.2 Search Strategy

As explained in Section 5.2, 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 5.

### 6.3 Results

The linear approximation biases (in absolute values) up to five rounds are given in Table 5. A 5-round approximation with a bias of $2^{-31}$ has been found through our searches. The details of the 5-round characteristic is given in Table 6.

## 7 Conclusion

FUTURE is a new, promising lightweight cipher designed for low latency and low hardware cost, based on an AES-like structure. In this paper, we conducted a MILP-based analysis of the cipher to find single-key differential and linear distinguishers. We incorporated the DDT and LAT probabilities into the model and obtained some previously unknown characteristics up to five rounds.

Table 5: 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 | $2^{-32}$ |  |  |

Table 6: 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 | $2^{-31}$ |

As an additional contribution, we showed an efficient way to model an $n$-XOR operation with one constraint. The proposed method can be used to improve the MILP models of various cryptanalysis methods in the literature.

The 5-round distinguishers we discovered improve the known distinguishers of FUTURE by one round. Nevertheless, they cannot be extended to the full version of the cipher, and hence do not pose an immediate threat to its security. FUTURE still enjoys a reasonable security margin.

## References

1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: Milp modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Transactions on Symmetric Cryptology pp. 99–129 (2017)
2. Boura, C., Coggia, D.: Efficient milp modelings for sboxes and linear layers of spn ciphers. IACR Transactions on Symmetric Cryptology pp. 327–361 (2020)
3. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: Milp-based automatic search algorithms for differential and linear trails for speck. In: International Conference on Fast Software Encryption. pp. 268–288. Springer (2016)
4. Funabiki, Y., Todo, Y., Isobe, T., Morii, M.: Several milp-aided attacks against snow 2.0. In: International Conference on Cryptology and Network Security. pp. 394–413. Springer (2018)
5. Gupta, K.C., Pandey, S.K., Samanta, S.: Future: A lightweight block cipher using an optimal diffusion matrix. In: International Conference on Cryptology in Africa. pp. 28–52. Springer (2022)
6. Gurobi Optimization, I.: Gurobi optimizer reference manual. URL http://www. gurobi. com (2018)
7. Ilter, M.B., Selçuk, A.A.: A new milp model for matrix multiplications with applications to klein and prince. In: SECRYPT. pp. 420–427 (2021)

8. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: International Conference on Information Security and Cryptology. pp. 57–76. Springer (2011)
9. Sasaki, Y., Todo, Y.: 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)
10. Sun, L., Wang, W., Liu, R., Wang, M.: Milp-aided bit-based division property for arx-based block cipher. Cryptology ePrint Archive (2016)
11. Sun, L., Wang, W., Wang, M.Q.: Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. IET Information Security **14**(1), 12–20 (2020)
12. 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: International Conference on Information Security and Cryptology. pp. 39–51. Springer (2013)
13. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: 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**, 2014 (2014)
14. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: 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)
15. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 9.2) (2020), `https://www.sagemath.org`
16. Yin, J., Ma, C., Lyu, L., Song, J., Zeng, G., Ma, C., Wei, F.: 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)
17. Zhu, B., Dong, X., Yu, H.: MILP-based differential attack on round-reduced GIFT. In: Cryptographers' Track at the RSA Conference. pp. 372–390. Springer (2019)