# Related-Key Cryptanalysis of FUTURE The Full Round Distinguishing Attack

Amit Jana, Smita Das, Ayantika Chatterjee, and Debdeep Mukhopadhyay

Indian Institute of Technology, Kharagpur {janaamit001,smita1995star,cayantika,debdeep.mukhopadhyay}@gmail.com

Abstract. In Africacrypt 2022, Gupta *et al.* introduced a 64-bit lightweight MDS matrix-based SPN-like block cipher designed to encrypt data in a single clock cycle with minimal implementation cost, particularly when unrolled. While various attack models were discussed, the security of the cipher in the related-key setting was not addressed. In this work, we bridge this gap by conducting a security analysis of the cipher under related-key attacks using MILP (Mixed Integer Linear Programming)-based techniques. Our model enables a related-key distinguishing attack on 8 rounds of FUTURE, requiring 2<sup>64</sup> plaintexts, 2<sup>63</sup> XOR operations, and negligible memory. Additionally, we present a 10-round boomerang distinguisher with a probability of 2<sup>-45</sup>, leading to a distinguishing attack with 2<sup>46</sup> plaintexts, 2<sup>46</sup> XOR operations, and negligible memory. This result demonstrates a full break of the cipher's 64-bit security in the related-key setting.

Keywords: Related key cryptanalysis · Boomerang attack · FUTURE.

## 1 Introduction

In recent years, the demand for cryptographic solutions optimized for resourceconstrained environments-such as RFID tags, sensor networks, and contactless smart cards-has led to the development of lightweight cryptographic primitives. Unlike traditional cryptographic methods like AES [21], SHA-256 [33], and RSA [35], which are designed for systems with substantial processing power and memory, lightweight cryptography prioritizes efficiency across various metrics including hardware cost, power utilization, and latency. Block ciphers, which can be thought of a pseudo-random permutations to transform plaintext into ciphertext blocks of fixed lengths, are mainly categorized into Feistel structures and substitution-permutation networks (SPNs). Feistel structures, used in ciphers like TWINE [45] and Piccolo [39], are cost-effective but require more rounds to ensure security, while SPNs offer robust security but can be more resource-intensive. The field of lightweight cryptography has expanded significantly, with ciphers such as PRESENT [16], KATAN [19], SIMON & SPECK [6], PRINCE [17], MANTIS [7], LED [24], MIDORI [4], and GIFT [5] being optimized for parameters like code size, latency, and energy consumption. Moreover, tweakable block ciphers like SKINNY [7], CRAFT [8], and QARMA [3], enhance encryption modes and authentication. Additionally, CRAFT addresses challenges such as resistance to Differential Fault Analysis (DFA) attacks.

Several lightweight block ciphers, including LED, MIDORI, and SKINNY, build on the fundamental structure of the AES round function, modifying its components to enhance performance. AES employs MDS (Maximum Distance Separable) matrices in its round function to achieve strong diffusion, which is essential for robust security against various cryptographic threats. However, incorporating MDS matrices into lightweight block ciphers poses a challenge due to their high implementation cost. This often necessitates additional rounds in these ciphers to maintain security against attacks such as differential and linear attacks. As a result, many lightweight block ciphers opt for lighter components, such as near-MDS matrices. This approach helps manage implementation costs while still aiming to achieve effective diffusion, even though MDS matrices offer superior diffusion benefits.

Mixed Integer Linear Programming (MILP) is a well-established optimization technique used to find the optimal solution for a linear objective function subject to a set of linear constraints. In 2011, Mouha et al. introduced an automated differential path search method utilizing MILP [31], which helps generate lower bounds for the number of active S-boxes. At that time, the method could not account for the differential properties of the S-box, limiting its application to bit-oriented ciphers like PRESENT and LS-designs [30]. This limitation was later addressed by Sunet al. [44, 43], who developed two distinct methods to model the differential propagation of S-boxes using systems of inequalities. The first approach uses logical conditions to represent differential properties through linear inequalities. The second approach employs a geometric method to capture all possible input-output difference transitions through an n-bit S-box, computing the H-representation (convex hull) of this set using the SageMath inequality generator function and simplifying constraints with a greedy approach. Sasaki and Todo [37] further advanced this technique by incorporating a MILP-based optimization phase to achieve a more compact representation of S-boxes with fewer constraints. Additionally, Boura et al. [18] enhanced the method by reducing the number of constraints needed to capture the differential properties of an S-box by adding related constraints from the set of constraints generated by the SageMath inequality generator function.

In 2022, Gupta *et al.* introduced a new 64-bit lightweight block cipher known as FUTURE [25], which stands out for its exceptionally low implementation cost compared to other block ciphers, particularly when implemented in an unrolled fashion. Notably, FUTURE is one of the few lightweight ciphers where all the round components are new, and it employs an MDS matrix for its diffusion layer. The internal functions of the cipher are designed for high hardware efficiency, with the MDS matrix and S-box being specifically optimized to minimize hardware costs. The S-box used is reported to match the cryptographic quality of those in SKINNY and Piccolo. Hardware benchmarks on FPGA and ASIC platforms demonstrated that FUTURE outperforms several well-known lightweight ciphers in terms of size, critical path, and throughput, achieving superior results across multiple metrics.

Researchers have explored various attack methods on the FUTURE cipher in single-key settings. In [26], a bit-based Mixed integer linear programming (MILP) approach was used to identify both differential and linear distinguishers, revealing distinguishers up to five rounds with probabilities of  $2^{-58}$  and  $2^{-62}$ , respectively. In [38], a meet-in-the-middle (MITM) technique combined with MILP demonstrated a key recovery attack with data, time, and memory complexities of  $2^{64}$ ,  $2^{126}$ , and  $2^{34}$ , respectively. Lin *et al.* [29] also employed a MILP-aided MITM attack, achieving complexities of  $2^{64}$  for data,  $2^{124}$  for time, and  $2^{48}$  for memory complexities. Additionally, Roy *et al.* [36] conducted an attack based on biclique structures with data, time, and memory complexities of  $2^{48}$ ,  $2^{125.54}$ , and  $2^{32}$ , respectively. Furthermore, Mondal *et al.* [30] applied Yoyo techniques in the secret-key settings to distinguish up to five and six rounds with data complexities of  $2^{9.83}$  and  $2^{58.83}$ , respectively.

Despite the theoretical attacks on FUTURE in single-key settings, there has been no related-key cryptanalysis attempted on this cipher, and the design proposal did not include any related-key analysis. This paper addresses this gap by providing a detailed examination of related-key cryptanalysis. We develop both word and bit-oriented MILP models to identify improved differential characteristics. Although [26] outlines a bit-oriented MILP model for searching differential characteristics in single-key settings for the FUTURE cipher, the description is incomplete for related-key models. In this work, we provide a comprehensive description of how to build both word and bit-oriented MILP framework for the FUTURE cipher, which can also be useful for constructing MILP frameworks for other SPN-like ciphers. To search for differential characteristics, our approach first employs the word-oriented model and then utilizes a bit-based model based on the optimal input-output difference patterns obtained from the word-oriented model. When searching for related-key differences, the basic word-oriented model is insufficient because it fails to capture optimal difference patterns due to the potential cancellation of intermediate word differences when both the state and key words are active. To address this, we introduce a new non-linear constraint to improve the identification of better difference patterns for optimal differential characteristics. We apply this enhanced search technique to the FUTURE cipher, covering up to 7 rounds.

Our Contributions. Our contributions are three-fold, as follows:

- We propose an extensive bit-based related-key MILP model for the FU-TURE cipher, which can be helpful for building MILP models for any MDS matrix-based SPN ciphers. We revisit Boura *et al.* [18]'s work due to insuffcient information in their proposed algorithm to generate optimal number of constraints to capture the behavior of DDT. From our understanding, we provide a revised algorithm that produces the same results as Boura *et al.* [18, Algorithm 1], but with a larger set of final constraints. Additionally, we provide a detailed explanation of how to construct a primitive representation of MDS (or near-MDS) matrices using a companion matrix approach, which is compatible with the cipher's structure.

- Utilizing this technique, we demonstrate an 8-round related-key differential characteristic for FUTURE with a probability of 2<sup>-63.4</sup>, which leads to a distinguisher with 2<sup>64</sup> data, 2<sup>63</sup> time, and negligible memory complexities.
- Additionally, we develop a full-round related-key boomerang distinguisher with practical complexities, indicating a full-round break of the cipher. A detailed comparison of the previous attack methods and their complexities is provided in Table 1.

Attool	Tumor	#Pounda	Sotting	Droh		Complexity		Reference	
Attack	Types	#Rounds	Settings	1 100.	Data	Time	Memory	Reference	
Differential	-	5	Single-key	2-58	-	-	-	[26]	
MITM	Key Recovery	10	Single-key	-	264	2126	2 <sup>34</sup>	[38]	
MITM	Key Recovery	10	Single-key	-	264	2124	248	[29]	
Biclique	Key Recovery	10	Single-key	-	≤ 2 <sup>48</sup>	2 <sup>125.53</sup>	2 <sup>32</sup>	[36]	
Varia	Distinguisher	6	Single-key	-	2 <sup>58.83</sup>	-	-	[90]	
1090	Distinguisher	8	Known-key	-	215	-	-	ျခပျ	
Differential	Distinguisher	8	Related-key	2-63.4	264	2 <sup>63</sup> XOR	Negligible	This Work	
Boomerang	Distinguisher	10	Related-key	2-45	2 <sup>46</sup>	2 <sup>46</sup> XOR	Negligible	This Work	

Table 1: A Comparison of Different Attacks on FUTURE

**Outline of the Paper.** The paper is structured as follows: Section 2 provides an overview of the FUTURE cipher. In Section 3, we present a brief introduction to related-key differential and boomerang cryptanalysis. Section 4 explains the bit-oriented MILP model used for the analysis. In Section 5, we apply this model to construct related-key differential and boomerang distinguishers for the FUTURE cipher. Finally, Section 8 concludes the paper with remarks and suggestions for future work.

# 2 Description of FUTURE

FUTURE is an SPN-based 64-bit lightweight block cipher designed to have applications on low hardware cost and latency. It has a key size of 128-bit.

**The Round Function.** The round structure of the FUTURE cipher consists of four operations: SubCell, MixColumn, ShiftRow, and AddRoundKey, as illustrated in Figure 2. Notably, the MixColumn operation is omitted in the final round. The cipher processes a 64-bit input state S arranged as a  $4 \times 4$  matrix, where each cell is a nibble (i.e.,  $s_i \in \{0, 1\}^4$  for  $0 \le i \le 15$ ), as shown in the Figure 1a. Furthermore, the round structure is depicted in the following Figure 2.

**SubCell.** The nonlinear transformation in the round function is defined by the SubCell operation, which applies a 4-bit S-box to each cell of the state matrix. This transformation is depicted in Figure 1b.



Fig. 1: The State Representation and S-box Table of FUTURE Cipher



Fig. 2: Round Function

**MixColumn.** The linear operation is represented by the finite field matrix multiplication involving the MDS (maximum distance separable) matrix ( $\mu$ ) and the state matrix, where the matrix elements are in  $GF(2^4)$ . The MDS matrix is illustrated in Figure 3a. Matrix and vector multiplications are performed in the field  $\mathbb{F}_{2^4}$ , defined by the primitive polynomial  $x^4 + x + 1$ .

	ך8 9 1 8 ך	5 <sub>0</sub> <i>s</i> 4	<b>S</b> 8	<i>s</i> <sub>12</sub>		[ <i>s</i> 0	<b>S</b> 4	<b>S</b> 8	s <sub>12</sub> <sup>-</sup>
	3299	<i>s</i> <sub>1</sub> <i>s</i> <sub>5</sub>	<b>S</b> 9	<b>s</b> 13		<i>s</i> <sub>13</sub>	$s_1$	<b>S</b> 5	<b>S</b> 9
$\mu =$	2389	S2 S6	<i>s</i> <sub>10</sub>	<b>S</b> 14	<b>→</b>	S10	<b>S</b> 14	<b>s</b> 2	<b>S</b> 6
	[9981]	<i>s</i> <sub>3</sub> <i>s</i> <sub>7</sub>	<i>s</i> <sub>11</sub>	<i>S</i> 15_		[ <i>s</i> 7	<i>s</i> <sub>11</sub>	<i>S</i> 15	<b>S</b> 3_

(a) The MixColumn Matrix (b) The ShiftRow Operation

Fig. 3: The MixColumn Matrix and ShiftRow Operation of FUTURE Cipher

**ShiftRow.** Each row (row(i), i = 0, 1, 2, 3) of the state matrix is rotated to the right by i positions following the MixColumn operation. This process is illustrated in Figure 3b.

**AddRoundKey.** The 64-bit round keys (sub-keys)  $SK_i$ , i = 0, 1, ..., 10 are XORed to the state S in each round. Additionally, the final round sub-key  $SK_{10}$  is XORed with the state before producing the ciphertext.

**Key Schedule.** In FUTURE encryption, a 128-bit secret key K is divided into two halves  $k_0$  and  $k_1$  for generating round and whitening keys.  $k_0$  acts as the whitening key and generates each round sub-keys  $SK_i$ , i = 0, 1, ..., 10 depends



Fig. 4: FUTURE Encryption Scheme

on whether i is even or odd. If i is even,  $k_0$  is left-rotated by  $5 \cdot \frac{i}{2}$  bits; if i is odd,  $k_1$  undergoes the same left rotation. Left rotation involves circularly shifting bits. Additionally, except for the  $5^{th}$  and  $10^{th}$  rounds, a single '1' bit is XORed into specific positions within 4-bit cells during each encryption round, with these operations defined by round constants.

## 3 Related Key Cryptanalysis

A related-key attack [9] involves analyzing a cipher using multiple keys with known mathematical relationships between them. The attacker has access to encryption or decryption functions with these keys and aims to determine the actual secret keys. The simplest form uses a constant ( $\Delta$ ) XOR relation between keys, such as  $K_2 = K_1 \oplus \Delta$ . Related-key attacks offer more freedom compared to other attacks but can be more challenging to implement. Resistance to such attacks is crucial, as exemplified by the design goals of AES cipher. This paper employs differential attacks and boomerang attacks in a related-key context.

**Related-Key Differential Cryptanalysis.** Differential cryptanalysis is an effective method for analyzing and attacking symmetric-key ciphers by examining the differences between pairs of plaintexts and ciphertexts, known as "differential". Introduced by Biham and Shamir [13] in 1990, this technique seeks to identify specific patterns in these differences, termed "differential characteristics", that are unique to the encryption algorithm. By studying these patterns,

cryptanalysts can infer the internal state of the cipher and, with sufficient data, uncover the secret key. This approach can lead to distinguishing attacks and key-recovery attacks.

In related-key attack settings, an attacker can establish or enforce a relationship between multiple keys and has access to the corresponding encryption and decryption functions for all these keys. Consider a tuple  $(\Delta_{in}, \Delta_K, \Delta_{out})$  as an *n*round related-key differential for a keyed round function  $f_{\mathcal{K}}$ , where  $f_{\mathcal{K}}^{\iota}$  represents the output after the *i*-th round for i = 0, 1, ..., n-1. This differential is valid if, for some plaintext P and key K, the equation  $f_K^{n-1}(P) \oplus f_{K \oplus \Delta_K}^{n-1}(P \oplus \Delta_{in}) = \Delta_{out}$ holds. Let  $S_{P,K}^{i}$  denote the internal state of the round function at round i with inputs P and K. The tuple  $(\Delta_{in}, \Delta_K, \Delta S_0, \dots, \Delta S_{n-1} = \Delta_{out})$  is an *n*-round related-key differential characteristic if  $(\Delta_{in}, \Delta_{out}, \Delta_K)$  is an *n*-round relatedkey differential and for all  $i, S_{P,K}^i \oplus S_{P\oplus\Delta_{\mathrm{in}},K\oplus\Delta_K}^i = \Delta S_i$ . Let  $p = \Pr[(\Delta_{\mathrm{in}}, \Delta_K) \to \Delta_{\mathrm{out}}]$  represent the probability that the related-key

differential  $(\Delta_{in}, \Delta_K, \Delta_{out})$  holds. This implies that if  $\frac{1}{\rho}$  number of plaintexts Pand keys K are selected uniformly at random, the equation  $f_K(P) \oplus f_{K \oplus \Delta_K}(P \oplus A_K)$  $\Delta_{\rm in}$ ) =  $\Delta_{\rm out}$  will be satisfied at least once.

Related-Key Boomerang Attack. The boomerang attack, introduced by Wagner in [47], is a differential cryptanalysis method that combines two highprobability differentials to enhance the chances of breaking a cipher. This is described in Figure 5a. For a block cipher  $E = E_1 \circ E_0$ , with differentials  $\Delta_0 \xrightarrow{\rho} \Delta_1$ for  $E_0$  and  $\nabla_0 \xrightarrow{q} \nabla_1$  for  $E_1$ , the attack checks if differential relationships hold, with an expected probability of success given by:

$$\Pr(E^{-1}(E(x) \oplus \nabla_1) \oplus E^{-1}(E(x \oplus \Delta_0) \oplus \nabla_1) = \Delta_0] = p^2 \cdot q^2.$$

The procedure for mounting the distinguisher in adaptive settings is as follows:

- 1. Request the ciphertexts  $C_0 = E(P_0)$  and  $C_1 = E(P_1)$ , where  $P_1 = P_0 \oplus \Delta_0$ . 2. Request the plaintexts  $P_2 = E^{-1}(C_2)$  and  $P_3 = E^{-1}(C_3)$ , where  $C_2 = E^{-1}(C_3)$ .  $C_0 \oplus \nabla_1$  and  $C_3 = C_1 \oplus \nabla_1$ .
- 3. Verify if  $P_2 \oplus P_3 = \Delta_0$ .

To amplify this attack, the amplified boomerang attack [27] was proposed which works in a non-adaptive (chosen-plaintext attack) scenario. In this attack, the expected probability to get a right quartet will be  $p^2 \cdot q^2 \cdot 2^{-n}$ . Furthermore, in [10, 11], they have pointed out that any value of  $\Delta_1$  and  $\nabla_0$  can be considered as long as  $\Delta_1 \neq \nabla_0$ . As a result, the probability of the right quartet is increased to  $2^{-n} \cdot \hat{p}^2 \cdot \hat{q}^2$ , where  $\hat{p} = \sqrt{\sum_i \Pr^2(\Delta_0 \to \Delta_1^i)}$  and  $\hat{q} = \sqrt{\sum_i \Pr^2(\nabla_0^j \to \nabla_1)}$ . Note that this amplification can be done in the adaptive setting to increase the

probability to  $\hat{p}^2 \cdot \hat{q}^2$  from  $p^2 \cdot q^2$ . The sandwich attack [22] further refines this approach by decomposing the cipher into three parts and using the Boomerang Connectivity Table [20] (BCT) to systematically analyze the connections between



boomerang (b) Related key Boomere

Fig. 5: The Boomerang Framework

input and output differences, improving the probability approximation of the distinguisher.

The related-key boomerang attack [12], depicted in Figure 5b, utilizes both key and plaintext differences. It assumes that the upper sub-cipher  $E_0$  follows a differential characteristic  $\Delta_0 \xrightarrow{p} \Delta_1$  under a key difference  $\alpha = K_0 \oplus K_1 = K_2 \oplus K_3$ , while the lower sub-cipher  $E_1$  has a differential characteristic  $\nabla_0 \xrightarrow{q} \nabla_1$  under a key difference  $\beta = K_0 \oplus K_2 = K_1 \oplus K_3$ . A related-key distinguisher is built using four different unknown keys:  $K_0$ ,  $K_1 = K_0 \oplus \alpha$ ,  $K_2 = K_0 \oplus \beta$ , and  $K_3 = K_1 \oplus \beta$ . The related-key boomerang distinguisher in the adaptive scenario is executed as follows:

- 1. Request the ciphertext pairs  $(C_0, C_1)$ , where  $C_0 = E_{K_0}(P_0)$  and  $C_1 = E_{K_1}(P_1)$ , with  $P_0 \oplus P_1 = \Delta_0, K_0 \oplus K_1 = \alpha$ .
- 2. Request the plaintexts pairs  $(P_2, P_3)$ , where  $P_2 = E_{K_2}^{-1}(C_2)$  and  $P_3 = E_{K_3}^{-1}(C_3)$ , with  $K_2 \oplus K_3 = \alpha$ ,  $C_2 = C_0 \oplus \nabla_1$  and  $C_3 = C_1 \oplus \nabla_1$ . 3. Verify if  $P_2 \oplus P_3 = \Delta_0$ .

## 4 Mixed-Integer Linear Programming models

Mixed-integer linear programming (MILP) has been successfully utilized to develop automated search algorithms for differential and linear cryptanalysis. Two primary modeling approaches exist for implementing ciphers: the word-oriented model and the bit-oriented model. In the word-oriented model, the cipher state is treated as a sequence of words, with each word represented as a binary variable. In contrast, the bit-oriented model represents each bit of the cipher state as a binary variable, ensuring the generation of the most optimal and valid differential characteristics without any inconsistencies in the trail. The MILP constraints introduced in Mouha *et al.*'s method are insufficient to fully capture the differential propagation behavior in linear diffusion layers built from non-MDS codes. In [42], the authors first proposed a bit-oriented model specifically for SPN ciphers that utilize bit permutation-based linear layers.

In this section, we model the FUTURE cipher components as constraints to construct a bit-based MILP model for analyzing differential characteristics. To build this model, the S-box, permutation, and matrix multiplication over a finite field are represented by linear inequalities with binary variables. In [26], the authors present a bit-based MILP model for the FUTURE cipher aimed at searching for single-key differential and linear characteristics. However, the details provided are incomplete, particularly regarding the generation of linear inequalities for the S-box and the conversion of the MDS matrix to a binary matrix using the companion matrix representation. By employing linear inequalities, one can construct a comprehensive bit-based MILP model that automatically identifies differential characteristics.

#### 4.1 Constraints for SubCell Operation

For differential cryptanalysis using bit-based MILP, the goal is to generate a minimal number of constraints involving input and output bits of an S-box to capture the actual behavior of the differences according to the difference distribution table (DDT). Let us assume that,  $(x_0, \ldots, x_{n-1})$  and  $(y_0, \ldots, y_{n-1})$ represent the input and output bit differences of an  $n \times n$  S-box respectively. The problem corresponds to modeling the fact that  $(x_0, \ldots, x_{n-1}) \rightarrow (y_0, \ldots, y_{n-1})$ is a possible difference transition in a DDT. In this regard, two different approaches were proposed in 2014 by Sun et al. [44, 43]. The first is a geometrical one and consists of computing the H-representation of the convex hull of the set of possible transitions. The second one is based on logical condition modeling. The first approach is to use the Sagemath inequality generator by taking all the valid difference transition points from the DDT and it generates the number of linear inequalities which satisfies all the valid difference transition points. However, the number of inequalities using Sagemath is typically quite high with many redundant inequalities. The authors of [44] applied a greedy algorithm to reduce the number of constraints. In this approach, the algorithm adds to the solution set the best possible inequality which can remove the highest number of impossible difference transition points among those that have not been removed yet. Later, Sasaki and Todo in [37] proposed a new reduction algorithm to further reduce the number of constraints compared to the greedy approach. They proposed to model the problem of minimizing the set of inequalities that remove all the impossible difference transition points as a MILP problem itself and solve it by some solver. More precisely, their method consists of assigning a binary variable  $z_i$  to each inequality in which  $z_i = 1$  denotes that inequality  $\boldsymbol{i}$  is included in the system. Then for each impossible difference transition point j, add the corresponding constraints in the list  $\mathcal{L}_j$  which leads to the inequality as  $\sum_{i \in \mathcal{L}_i} z_i \ge 1$ . Finally, the MILP solver is used for minimizing  $\sum_i z_i$  giving a solution to optimize the constraints to capture the DDT of an S-box. However,

**Algorithm 1** Revised Boura *et al.*'s Approach to Compute a Set of Inequalities from DDT of an S-box.

1: procedure COMPUTECONSTRAINTS( $\mathcal{VDP}, k(\geq 2)$ ) 2:  $\mathcal{H}_{set} \leftarrow \text{inequality}_{generator}(\mathcal{VDP})$ 3:  $\mathcal{C}_{set} \leftarrow \mathcal{H}_{set}$ 4:  $\mathcal{D}_{set} \leftarrow \{\}$ for all  $\alpha \in \mathcal{VDP}$  do 5: $\mathcal{H}_{set}^{\alpha} = \{ C \in \mathcal{H}_{set} | C(\alpha) = 0 \}$ 6: for all  $\mathcal{H}_{set'}^{\alpha} \alpha \in \mathcal{VDP}$  do 7:if  $k \ge |\mathcal{H}_{set}^{\alpha}|$  then 8: for all  $\{C_1, \dots, C_k\} \subseteq \mathcal{H}_{set}^{\alpha}$  do  $C_{new} = C_1 + \dots + C_k$ 9: 10: Add Cnew into Dset 11: for all constraints  $C_i \in \mathcal{H}_{set}$ ,  $1 \le i \le |\mathcal{H}_{set}|$  do 12:Construct the set  $S_i = \{\beta \in \mathcal{IDP} | C_i(\beta) < 0\}$ 13: $\mathcal{L} = [S_1, \cdots, S_{|\mathcal{H}_{set}|}]$  $\blacktriangleright$  A list of sets 14:for all constraints  $C_j \in \mathcal{D}_{set}, 1 \leq j \leq |\mathcal{D}_{set}|$  do 15:Construct the set  $S = \{\beta \in \mathcal{IDP} | C_i(\beta) < 0\}$ 16:17:if  $\not \exists$  any  $S_i$  from  $\mathcal{L}$  such that  $S \subseteq S_i$  then 18: Add  $\boldsymbol{S}$  in the list  $\boldsymbol{\mathcal{L}}$ 19:Add  $C_i$  in  $C_{set}$ 20: Apply Sasaki et al.'s [37] approach from  $C_{set}$  to finally get the optimal number of constraints to capture the DDT of S-box

there are other works [46, 28, 34] that further reduce the number of constraints to capture DDT of S-box by proposing new approaches to generate additional inequalities, surpassing the Sagemath inequality generator or using Boura et al.'s approach. In this work, we follow the method outlined by Boura *et al.* [18] to generate the constraints that capture the DDT of a FUTURE S-box. The DDT of the FUTURE S-box is provided in Table 3a (in Appendix A). The algorithm presented by Boura et al. [18, Algorithm 1] for deriving a set of inequalities from the DDT of an S-box lacks certain details. Specifically, in step 7, the authors state that if  $C_{new}$  removes a new set of impossible transitions,  $C_{new}$  should be added as new constraints to capture the DDT of the S-box. However, the precise meaning of  $C_{new}$  is unclear. It is not explicitly explained whether the set S of impossible transitions from  $C_{new}$  should contain elements distinct from the elements of  $S_i$ , for all i, where each  $S_i$  is a set of impossible transitions for every constraint in  $\mathcal{C}_{set}$ . To clarify this, we revisited their approach and outlined the complete steps necessary to generate the S-box constraints based on our understanding.

Suppose, a difference transition  $x \to y, x, y \in \mathbb{F}_2^4$  through the S-box can be seen as a vector of  $\mathbb{F}_2^{2n}$ , involving 2n binary variables represented as  $(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1})$  or as  $(x_0, \ldots, x_{2n-1})$ . Let  $\mathcal{VDP}$  and  $\mathcal{IDP}$  denote a set of valid and impossible difference transition points according to the DDT of FUTURE S-box. For example,  $0x12 \in \mathcal{VDP}$  and  $0x11 \in \mathcal{IDP}$  are the valid and invalid difference points according to the DDT in Table 3a. Given  $\mathcal{VDP}$  to the inequality\_generator() function in the sage.geometry.polyhedron class of Sagemath returns a list of inequalities as the H-set representation of the convex hull of all possible transitions in a DDT. For FUTURE S-box, we get returns 214 inequalities. We denote this list of inequalities as  $\mathcal{H}_{set}$ . This set  $\mathcal{H}_{set}$  has the following properties: (1.) each  $\alpha \in \mathcal{VDP}$  must satisfy all the constraints in  $\mathcal{H}_{set}$ and (2.) each  $\beta \in \mathcal{IDP}$  will not be satisfied by at least one of the constraints in  $\mathcal{H}_{set}$ . The interesting point here is that the addition of any number of constraints always maintains the above two properties. Although, adding the constraints by choosing all the subsets (of cardinality k) of constraints from  $\mathcal{H}_{set}$  and then append it to  $\mathcal{H}_{set}$  can increase the list  $\mathcal{H}_{set}$  remarkably high. Instead, for each  $\alpha \in \mathcal{VDP}$ , the authors choose the constraints from  $\mathcal{H}_{set}$  which satisfies by the point  $\alpha$  and store them in another list  $\mathcal{H}_{set}^{\alpha}$ . Then, for each set  $\mathcal{H}_{set}^{\alpha}$ , choose  $\binom{|\mathcal{H}_{aet}^{\sigma}|}{\nu}$  constraints, denoted  $C_1, \ldots, C_k$ , and add  $C_{new} = C_1 + \cdots + C_k$  to a set  $\mathcal{D}$ set. A list of sets,  $\mathcal{L}$ , is constructed, where each entry contains the impossible transition points for each constraint in  $\mathcal{D}_{set}$ . Finally, the constraints from  $\mathcal{D}_{set}$ are filtered and added to  $\mathcal{L}$ , ensuring that the set of impossible transition points S is not a subset of any set already in  $\mathcal{L}$ . After this filtration, Sasaki and Todo's method is applied to the selected constraints to generate the optimal constraints for capturing the DDT of the S-box. These steps are detailed in Algorithm 1.

Using this algorithm with k = 2, we obtain 971 constraints for  $C_{set}$  and ultimately reduce this to 17 constraints using Sasaki *et al.*'s approach. In comparison, according to [18, Algorithm 1], the authors generated approximately 500 constraints for the PRESENT S-box with k = 2, which were reduced to 17 constraints using Sasaki *et al.*'s method. However, by applying the revised approach outlined in Algorithm 1, we generated 1138 constraints for  $C_{new}$  for the PRESENT S-box, which were similarly reduced to 17 constraints using Sasaki *et al.*'s method. Using the revised Algorithm 1, we obtained 17 constraints for both the FUTURE and PRESENT S-boxes to capture the DDT. These constraints are depicted in Figure 16 and Figure 17 (in Appendix A), respectively.

#### 4.2 Constraints for MixColumn Operation

For word-based MILP modeling, Mouha *et al.* [31] modeled the MixColumn matrix multiplication using its branch number, i.e., a lower bounds on the number of active S-boxes. Whereas for a bit-based model, an MDS (or near MDS) matrix  $\mu$  must be converted to a binary matrix over the base field  $\mathbb{F}_2$ , which is called the primitive representation of M. In [40], the authors give a short description of the primitive representation of  $\mu$  using a companion matrix. However, in [41], Sun *et al.* provided a method to obtain a primitive representation using linear maps with matrix representation. In this work, we thoroughly explore how to efficiently compute a primitive representation of  $\mu$  using a companion matrix that is compatible with the cipher's bit format, whether it follows a least significant bit (LSB) to the most significant bit (MSB) order or an MSB to LSB order. Finally, to model matrix multiplication in MILP, we might need several binary XOR operations.

For 1-XOR operation,  $c = a \oplus b$ , a, b,  $c \in \{0, 1\}$ , Mouha *et al.* [31] modeled it using 4 constraints and 3 variables as  $a+b+c \ge 2d_1$ ,  $d_1 \ge a$ , b, c, where  $d_1$ is a dummy variable. Similarly, for  $d = a \oplus b \oplus c$ , a, b, c,  $d \in \{0, 1\}$ , known as a 2-XOR operation, the approach requires 8 constraints and 5 variables. Yin *et al.* [48] showed a method to model it using 8 constraints and 4 variables. However, Fu *et al.* [23] efficiently model the 1-XOR operation using only one constraint  $a+b+c+d = 2d_1$ , a, b, c, d,  $d_1 \in \{0, 1\}$ . Based on this approach, the authors in [26] extend this approach to model the *n*-XOR operations  $a_0 \oplus \ldots \oplus a_{n-1} = b$ , as follows.

$$a_0 + \ldots + a_{n-1} + b = \begin{cases} (n+2)d_1 - (nd_2 + (n-2)d_3 + \ldots + 2d_{\frac{n}{2}+1}) & \text{if } n \text{ is even} \\ (n+1)d_1 - ((n-1)d_2 + (n-3)d_3 + \ldots + 2d_{\frac{n-1}{2}+1}) & \text{if } n \text{ is odd} \end{cases}$$

In our model, we adopt this approach for *n*-XOR operations for MixColumn matrix multiplication. In FUTURE cipher, the multiplication by  $4 \times 4$  MDS matrix  $\mu$  is performed over  $GF(2^4)$ , defined by the primitive polynomial  $x^4 + x + 1$ . Let,  $\alpha$  be a primitive element, serving as a root of the polynomial  $x^4 + x + 1$ . The MDS matrix mu includes field elements 1, 2, 3, 8, and 9 from  $GF(2^4)$ . To model the multiplication by  $\mu$  for bit-oriented MILP, we need to convert the  $4 \times 4$  MDS matrix  $\mu$  over  $GF(2^4)$  into a primitive representation of  $\mu$ , i.e., a  $16 \times 16$  binary matrix over the base field  $\mathbb{F}_2$ . Using linear maps with matrix representation, the authors [26] express the corresponding  $4 \times 4$  binary matrices of these field elements in Figure 6.

	۲1000Ţ		ך0 1 0 0 ס			ך11007		[1001 <sup>-</sup>		ך1 1 0 0 ך	Ĺ
-	0100		0010			0110	0	1100		0110	Ĺ
1 =	0010/	2 =	1001	'	3 =	1011	, 8=	0110	, 9=	1011	Ĺ
	0001		1000			1001		0010		1001	Ĺ

Fig. 6:  $4 \times 4$  binary matrix representation of the field elements in  $\mu$ 

Note that, the primitive representation of  $\mu$  by replacing the corresponding field elements 1, 2, 3, 8, and 9 is compatible with cipher representation from MSB to LSB. However, this primitive representation would not be compatible with the cipher representation from LSB to MSB. Here we will describe how to construct the primitive representation of  $\mu$  using a companion matrix to model the cipher which would be compatible in both ways. We know that  $2 = 0010 \in GF(2^4)$  is the root  $\alpha$  of the primitive polynomial  $x^4 + x + 1$  over  $Gf(2^4)$ . Let us assume that, the state of the cipher represented from MSB to LSB, i.e.,  $S = s_{63}||s_{62}|| \dots ||s_0|$ . In this case, the companion matrix representation of  $\alpha$  of the monic primitive polynomial  $c_0 + c_1x + c_2x^2 + c_3x^3 + x^4$ ,  $c_i \in \mathbb{F}_2$  can

be written as

$$2 = \alpha = \begin{bmatrix} c_3 \ 1 \ 0 \ 0 \\ c_2 \ 0 \ 1 \ 0 \\ c_1 \ 0 \ 0 \ 1 \end{bmatrix} = \begin{bmatrix} 0 \ 1 \ 0 \ 0 \\ 0 \ 0 \ 1 \ 0 \\ 1 \ 0 \ 0 \ 1 \end{bmatrix} \text{ with } 1 = \alpha^0 = \begin{bmatrix} 1 \ 0 \ 0 \ 0 \\ 0 \ 1 \ 0 \ 0 \\ 0 \ 0 \ 1 \ 0 \\ 0 \ 0 \ 1 \ 0 \end{bmatrix}.$$

The other field elements of  $\mu$  can be computed as

$$3 = \alpha + 1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, 8 = \alpha^3 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, 9 = \alpha^3 + 1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Thus the  $16 \times 16$  binary matrix  $M_1$ , representing the primitive form of  $\mu$ , corresponds to the cipher's bit representation from MSB to LSB over  $\mathbb{F}_2$  is given in Figure 13 (in Appendix A). On the other hand, if the cipher is represented from LSB to MSB, i.e.,  $S = s_0 ||s_1|| \dots ||s_{63}$ , then the companion matrix representation of  $\alpha$  of the monic primitive polynomial  $c_0 + c_1 x + c_2 x^2 + c_3 x^3 + x^4$ ,  $c_i \in \mathbb{F}_2$  can be written as

$$2 = \alpha = \begin{bmatrix} 0 & 0 & 0 & c_0 \\ 1 & 0 & c_1 \\ 0 & 1 & 0 & c_2 \\ 0 & 0 & 1 & c_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ with } 1 = \alpha^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Similarly, the other field elements of  $\mu$  can be computed as

$$3 = \alpha + 1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \ 8 = \alpha^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \ 9 = \alpha^3 + 1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Thus the  $16 \times 16$  binary matrix  $M_2$ , serving as the primitive representation of  $\mu$ , corresponds to the bit order from LSB to MSB over  $\mathbb{F}_2$  is given in Figure 13 (in Appendix A). Apart from these two companion matrix representations, using any other form of companion matrix in the model either by transposing it or by reordering the rows/columns of the above two matrices would not be compatible with the cipher representation. This is because altering the companion matrix used to construct the primitive representation M would change the bit sequences. As a result, multiplying M (a  $16 \times 16$  matrix) by the state (a  $16 \times 4$  matrix) would not produce a correct state consistent with the cipher's structure.

Finally, the  $4 \times 4$  state matrix of FUTURE cipher can be further deduced to  $16 \times 4$  binary matrix. Let, the 16-bit column vectors as  $y = (y_0, y_1, \dots, y_{15})^T$  and  $t = (t_0, t_1, \dots, t_{15})^T$ , where  $t = M \cdot y$ . The 16 constraints corresponding to one column transformation of the state after the MixColumn operation are given in Figure 14 (see Appendix A). Therefore, for all four columns of the state, a total of  $16 \cdot 4 = 64$  constraints are required to represent the differential propagation through the MixColumn operation.

#### 4.3 Constraints for ShiftRow Operation

The ShiftRow operation performs a row-wise shift at the nibble level, which can be represented as a bit-wise permutation  $\pi : \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$ . To model this operation, the binary variables resulting from the MixColumn step are permuted by ShiftRow. After that, 64 new binary variables are introduced and assigned to these permuted values. If  $x_i$  and  $y_i$  represent the input and output binary variables respectively, the constraint  $y_i = \pi(x_i)$  is added to the MILP model. To reduce the number of constraints, the output binary variables can be directly permuted according to the ShiftRow bit-wise permutation  $\pi$  while modeling the MixColumn operation.

#### 4.4 Constraints for AddRoundKey Operation

The AddRoundKey operation directly XORs the state bits with the round keys. In the bit-oriented related-key model, the state difference is XORed directly with the sub-key difference. To model the XOR operation between the key and state differences ( $c = a \oplus b$ ), we use the following constraints without introducing dummy variables:  $c \ge a - b$ ,  $c \ge b - a$ ,  $c \le a + b$ ,  $c \le 2 - a - b$ .

#### 4.5 Construction of the Objective Function

The objective function of an MILP model can be designed to minimize the number of active S-boxes. In a bit-oriented MILP model, there will be no inconsistencies in the propagation of bit differences through rounds, provided the S-box constraints accurately represent its DDT. To account for an active S-box in a bit-based model, we introduce a dummy variable along with four additional constraints for each S-box. Let the input bit differences of an S-box be represented by  $(\delta x_3, \delta x_2, \delta x_1, \delta x_0)$  and define a new binary dummy variable,  $d_0$ . This dummy variable  $d_0$  will determine whether the S-box is active or inactive

This dummy variable  $d_0$  will determine whether the S-box is active or inactive based on the following constraints:  $\sum_{i=0}^{n-1} \delta x_i \ge d_0, d_0 \ge \delta x_i, i = 0, 1, 2, 3$ . The

objective function is then to minimize the sum of the dummy variables  $d_i$  for each S-box position in the rounds. To calculate the probability of the differential trail produced by the model, the probability of each active S-box from the DDT must be checked, and the overall probability of the differential characteristic is obtained by multiplying these values. For a clearer understanding of the MILP model applied to the FUTURE cipher, we provide our MILP model implementation in [2].

### 5 Results

This section presents an analysis of the differential characteristics of FUTURE in the related key attack setting. The differential characteristics are determined using the methodology in Section 4.

Algorithm 2 Distinguishing Attack against FUTURE Reduced to 8 Rounds

1:	procedure DISTINGUISHER(( $\Delta P$ = 0x0000800011800008, $\Delta S$	$K_0 =$
	$0 \times 0000800011800008,  \Delta S K_1 = 0 \times 00000200002$	00002)
	$\xrightarrow{\text{8 round differential}} \Delta C = 0 \times 01442 \alpha 0000899108 )$	
2:	Randomly choose a key $K = SK_0    SK_1 \stackrel{\$}{\leftarrow} \{0, 1\}^{128}$ .	
3:	Form another key $K' = K \oplus \Delta SK_0    \Delta SK_1$ . $\triangleright$ Steps 2 and 3 are chosen	ı by the
	oracle.	
4:	Choose $2^{63}$ distinct plaintexts $P_i$ , $i = 1, 2, \dots, 2^{63}$	
5:	for $i = 1$ to $2^{63}$ do	
6:	Query $P_i$ to the encryption oracle under the key $K$ and obt	ain the
	corresponding ciphertext $C_i = E_K(P_i)$ .	
7:	Query $P'_i = P_i \oplus \Delta P$ to the encryption oracle under the key $K'$ and	l obtain
	the corresponding ciphertext $C'_i = E_{\mathcal{K}'}(\mathcal{P}'_i)$ .	
8:	if $C_i \oplus C'_i == \Delta C$ then	
9:	Return 1	rounds.
10:	Return 0 ► The oracle is a random perm	utation.

#### 5.1 Related-Key Differential Distinguishers

To search for the differential characteristics of FUTURE in the related-key setting, we constructed an MILP model using the Gurobi Python API [1]. The necessary constraints for building the model across rounds are outlined in Section 4. A summary of the related key characteristics for different rounds, along with their probabilities, is presented in Table 2. This model enables us to search for related-key differential characteristics up to 7 rounds. However, due to the large number of constraints and variables, the model struggles to complete the search for 6 and 7 rounds. For the 7-round case, we identified several differential characteristics with 22 active S-boxes and a probability of approximately  $2^{-48}$ . The 7-round differential characteristic is shown in Figure 15 in Appendix ??. To confirm the individual probabilities for each S-box, the DDT, and inverse DDT are provided in Table 3a and Table 3b, respectively. Additionally, we identified three distinct clustering effects for the 7-round differential characteristic (see Table 2) from 50 different solutions generated by the model, where the characteristics share the same input and output. This clustering further increases the probability of the differential characteristic to  $3 \cdot 2^{-48} \approx 2^{-46.4}$ . Furthermore, for 8 rounds, the solver could not reach a near-optimal solution due to the large number of constraints and variables. Therefore, we extended the 7-round differential characteristic by adding an additional round. Using the MILP model, we verified that seven S-boxes are active in the final round, with a probability of  $2^{-17}$ . Consequently, the overall probability for the 8-round differential characteristic becomes  $2^{-46.4} \cdot 2^{-17} = 2^{-63.4}$ . This can be directly leveraged to mount an attack on the security notion of indistinguishability against FU-TURE reduced to 8 rounds. The attack procedure is detailed in Algorithm 2. In this distinguisher, the attacker requires  $2^{63}$  plaintext pairs, effectively exhausting the entire plaintext space. The offline time complexity amounts to  $2^{63}$ 

XOR operations. The attack does not necessitate storing intermediate values, except for one ciphertext when  $C_i \oplus C'_i = \Delta C$  is satisfied. Therefore, the memory complexity is minimal, or effectively negligible.

*Experimental Verification.* As previously mentioned, the bit-oriented MILP model guarantees no inconsistencies in the solutions it returns. In our experiments, we successfully verified differential characteristics with a probability greater than  $2^{-32}$ . The implementation used to verify these characteristics is available in [2].



Fig. 7: Eight Round Related Key Differential Characteristic of FUTURE Cipher

### 5.2 Related Key Boomerang Distinguisher

In this section, we construct boomerang distinguishers for FUTURE over different rounds. Using our automated search model, we identify two distinct relatedkey differential characteristics for five rounds each, corresponding to the upper and lower halves of the boomerang. These characteristics have probabilities of  $2^{-14}$  and  $2^{-16}$ , respectively. For clarity, let  $\Delta_0 \xrightarrow{Upper Trail} \Delta_1$  and  $\nabla_0 \xrightarrow{Lower Trail} \nabla_1$  denote the differential characteristics for the upper and lower five rounds of the full boomerang, respectively. Additionally, let  $\alpha$  and  $\beta$  represent the differences in the round keys of the upper and lower trails. The full round boomerang structure is illustrated in Figure 8. Thus, the distinguishing probability for this boomerang is given by  $(2^{-14})^2 \cdot (2^{-16})^2 = 2^{-60}$ . , which can be utilized to perform a distinguishing attack on the full-round FUTURE cipher under adaptively chosen plaintext and ciphertext (ACPC) settings. The detailed attack procedure is presented in Algorithm 3. In this distinguisher, the

#Rounds	#Active	Differe	ential	Probability			
#Hounds	S-box	Input Differences	Output Difference	Tiobability			
		$\Delta P = 0 \times 2300\ 0010\ 0001\ 0000$					
4	2	$\Delta K_0 = 0 \times 2300\ 0010\ 0001\ 0000$	$\Delta C = 0 \times 0000 \ 4000 \ 0440 \ 008c$	2-5			
		$\Delta K_1 = 0 \times 0004 \ 0000 \ 0000 \ 0000$					
		$\Delta P = 0 \times 1201 \ 01 c 0 \ 0000 \ 0000$					
5	6	$\Delta K_0 = 0 \times 0000 \ 01 c 0 \ 0000 \ 0000$	$\Delta C = 0 \times 0008 \ 0000 \ 8000 \ 0802$	2 <sup>-14</sup>			
		$\Delta K_1 = 0 \times 0000 \ 0200 \ 0020 \ 0002$					
		$\Delta P = 0 x c 840\ 0000\ 0000\ 0005$					
6	11	11	11	11	$\Delta K_0 = 0 x c 840\ 0000\ 0000\ 0005$	$\Delta C = 0 \times 0200\ 0010\ 0001\ d420$	2-27
		$\Delta K_1 = 0 \times 0000\ 0000\ 0000\ 2480$					
		$\Delta P = 0 \times 0000\ 8000\ 1180\ 0008$					
7	22	$\Delta K_0 = 0 \times 0000 \ 8000 \ 1180 \ 0008$	$\Delta C = 0 \times 0000 \ 0007 \ 1002 \ 0000$	2 <sup>-48</sup>			
		$\Delta K_1 = 0 \times 0000\ 0000\ 000b\ 0000$					
		$\Delta P = 0 \times 0000\ 8000\ 1180\ 0008$					
8	22	$\Delta K_0 = 0x0000\ 8000\ 1180\ 0008$	$\Delta C = 0 \times 0144 \ 2a 00 \ 0089 \ 9108$	2-63.4			
	22	$\Delta K_1 = 0 \times 0000\ 0000\ 000b\ 0000$					

 Table 2: Related Key Differentials for Different Rounds of FUTURE using Bit 

 Oriented MILP Model

attacker needs  $2^{60}$  plaintext pairs, which corresponds to  $2^{61}$  plaintexts in total. The offline time complexity is  $2 \cdot 2^{60} = 2^{61}$  XOR operations. The attack does not require storing intermediate values, except for one plaintext  $P_i$  when  $P_2^i \oplus P_3^i = \Delta_0$  is satisfied. As a result, the memory complexity is negligible.

Checking Incompatibilities in the Boomerang. In boomerang-style attacks, selecting compatible differential characteristics for  $E_0$  and  $E_1$  is crucial, as independent choices can lead to incompatibility and reduce the probability of generating a right quartet to zero. Murphy [32] highlighted that dependencies between characteristics can benefit attackers. Biryukov *et al.* introduced the middle-round S-box trick [14], and later Biryukov and Khovratovich [15] proposed techniques like the ladder and S-box switch to improve probabilities. These ideas were formalized by Dunkelman et al. as the sandwich attack [22], which divides the cipher into three parts, enhancing the overall probability. Further, to evaluate the middle part efficiently and systematically, the authors [20] introduced a boomerang connectivity table (BCT) for a single round.

Suppose that the middle layer at the fourth round of the given boomerang (Figure 8) is composed of 16 S-box layers independently. For more clarity, we only chose one s-box layer which is depicted in Figure 9. According to Figure 9, the BCT [20] is defined in the following way.

# $\operatorname{BCT}(\Delta_i, \nabla_o) = \{ x \in \{0, 1\}^4 : S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_0) \oplus \nabla_o) = = \Delta_0 \}.$

This BCT provides a unified representation of existing observations on checking the inconsistency as well as the dependencies to further increase the probability of the boomerang for a single round.

**Incompatibility.** Incompatibility occurs when, as shown in Figure 9, the boomerang connection table (BCT) entry  $BCT(\Delta_i, \nabla_o) = 0$ , meaning the



Fig. 8: Full Round Boomerang Distinguisher



Fig. 9: Single S-box Layer in the Middle of the Boomerang

boomerang cannot be formed. If BCT $(\Delta_i, \nabla_o) \neq 0$ , the differential characteristics are compatible to form the quartet for the boomerang.

**Ladder Switch.** The ladder switch, introduced in [15], occurs when  $\Delta_i \neq 0$  and  $\nabla_o = 0$ , resulting in BCT $(\Delta_i, \nabla_o) = 2^4$ , i.e.,  $\Pr[\Delta'_i = \Delta_i] = 1$ . Geometrically, when  $\nabla_o = 0$ , the upper planes coincide, and the input pairs  $(x_3, x_4)$  on the opposite plane are directly replaced by  $(x_1, x_2)$ . In a similar fashion, if  $\Delta_i = 0$  and  $\nabla_o \neq 0$ , the lower planes coincide, and the input pairs  $(y_1, y_3)$  on the opposite plane are directly replaced by  $(y_2, y_4)$ .

#### Algorithm 3 Boomerang Distinguishing Attack against the FUTURE Cipher

1:	procedure	DISTINGUISHER(	(	$\Delta_0, \Delta$	Δ1,	$\nabla_0$ ,	$\nabla_1$ ,	α,	δ)	
----	-----------	----------------	---	--------------------	-----	--------------	--------------	----	----	--

- Randomly choose a key K<sub>0</sub> 
   <sup>\$</sup> {0,1}<sup>128</sup>. 
   Steps 2 and 3 are chosen by the oracle.

   Form another keys K<sub>1</sub> = K<sub>0</sub> ⊕ α, K<sub>2</sub> = K<sub>0</sub> ⊕ β, and K<sub>3</sub> = K<sub>0</sub> ⊕ α ⊕ β.
   Choose 2<sup>60</sup> distinct plaintext pairs as (P<sup>i</sup><sub>0</sub>, P<sup>i</sup><sub>1</sub> ⊕ Δ<sub>0</sub>), i = 1, 2, ..., 2<sup>60</sup>
- 5: for i = 1 to  $2^{60}$  do

6: Query  $P_0^i$  to the encryption oracle under the key  $K_0$  and obtain the corresponding ciphertext  $C_0^i = E_{K_0}(P_0^i)$ .

- 7: Query  $P_1^i = P_0^i \oplus \Delta_0$  to the encryption oracle under the key  $K_1$  and obtain the corresponding ciphertext  $C_1^i = E_{K_1}(P_1^i)$ .
- 8: Compute  $C_2^i = C_0^i \oplus \nabla_1$  and  $C_3^i = C_1^i \oplus \overline{\nabla_1}$ .
- 9: Query  $C_2^i$  to the decryption oracle under the key  $K_2$  and obtain the corresponding plaintext  $P_2^i = E_{K_2}(C_2^i)$ .
- 10: Query  $C_3^i$  to the decryption oracle under the key  $K_3$  and obtain the corresponding plaintext  $P_3^i = E_{K_3}(C_3^i)$ .

11:	if $P_2^l \oplus P_3^l == \Delta_0$ then	
12:	Return 1	$\blacktriangleright$ The oracle is the

13: Return 0 ► The oracle is a random permutation.

FUTURE cipher.

**S-box Switch.** The S-box switch, introduced in [15], occurs when  $\text{DDT}(\Delta_i, \Delta_o) \neq 0$  and  $\Delta_o = \nabla_o$ , resulting in  $\text{BCT}(\Delta_i, \nabla_o) = \text{DDT}(\Delta_i, \Delta_o)$ , i.e.,  $\Pr[\Delta'_i = \Delta_i] = \frac{\text{DDT}(\Delta_i, \Delta_o)}{2^4}$ . Geometrically, when  $\Delta_o = \nabla_o$ , the upper planes interchange their input pairs, i.e., the input pairs  $(\mathbf{x}_3, \mathbf{x}_4)$  on the opposite plane are directly replaced by  $(\mathbf{x}_2, \mathbf{x}_1)$ , consistent with  $\text{DDT}(\Delta_i, \Delta_o)$ .

Based on the switch techniques and the BCT, we verified the compatibility of the full round boomerang distinguisher shown in Figure 8. In this distinguisher, the S-box layer in the fifth round (Round 4) is chosen as the middle layer. We examine the state difference at the Round 4 S-box layer for the upper differential trail and the state difference at the Round 5 S-box layer for the lower trail. This setup is illustrated in Figure 10. As shown, only the third S-box is active in the upper trail, while all S-box nibbles are active in the lower trail. Consequently, all nibbles except the third in the middle layer fall under the ladder switch category, resulting in a probability of 1. For the third nibble position,  $\Delta_i = 0 \times 07$  and  $\nabla_o = 0 \times 0c$ , where we confirmed that BCT $(0 \times 07, 0 \times 0c) = 2$ , validating the compatibility of our differential characteristics to form the full-round boomerang distinguisher.

**Refinements to the Boomerang Distinguisher.** In the previous paragraph, we demonstrated the compatibility of the two differential characteristics necessary to form a full-round boomerang using middle-round switch effects. Now,

we will delve into a more detailed analysis of how these switching effects can be leveraged to significantly enhance the boomerang probability. As shown in Figure 10, there are three active S-boxes at positions 12, 14, and 15 in the lower trail during round 5. Tracing this lower trail backward, the first column  $((0, 0, 0, 8)^T)$  contains a single active nibble,  $0 \times 08$ , at the third position following the inverse ShiftRow operation. This nibble difference,  $0 \times 08$ , arises from the difference  $0 \times 09$  after the inverse S-box operation in round 5. Notably, the other two active S-boxes in round 5 do not impact the first column after the inverse ShiftRow and MixColumn operations, as depicted in Figure 11.



Fig. 10: Middle Round Switching Effects

According to the Figure 11, if we chose all possible differences  $\delta$  from  $0 \times 9$ through S-box inverse, i.e.,  $0x9 \xrightarrow{inverse DDT} \{0x1, 0x7, 0x8, 0xa, 0xc, 0xe\}$ (see Table 3b), we get different  $\delta_3(=\eta_3) \in \{0xc, 0x8, 0xd, 0xf, 0xa, 0x9\}$ through inverse ShiftRow and MixColumn operations. Finally, we checked that if  $\delta_3(=\eta_3) \in \{0xc, 0xa, 0x9\}$ , then BCT( $\zeta(=0x7), \eta \neq 0$ . This demonstrates that the two differential characteristics are compatible for forming the quartet in the boomerang if the output differences are  $\{0x8, 0xc, 0xe\}$  from the input difference  $0 \times 09$  through the inverse S-box operation at round 5 (at position 15 in the lower half). This increases the probability from  $2^{-3}$  (=  $\Pr[0x9 \xrightarrow{S^{-1}} 0x8]$  to  $2^{-1}$ . Furthermore, any possible output differences from the input differences 0x05 and 0x02 (i.e.,  $Pr[0x5 \xrightarrow{S^{-1}} *] = 1$ ,  $Pr[0x2 \xrightarrow{S^{-1}} *]$ \*] = 1) do not affect the first column after the ShiftRow and MixColumn inverse operations, increasing the probability from  $2^{-5}$  to 1. Similarly, the output difference corresponding to the active S-box at round 4 for the upper half can be arbitrary, i.e.,  $0x7 \xrightarrow{\text{DDT}} *$ . This also increases the probability from  $2^{-2}$  to 1. As a result, for the one lower half, the probability improves by a factor of  $2^7$ . Thus, for the two parallel lower halves, the probability improves by a factor of  $(2^7)^2 = 2^{14}$ . For upper halves, the probability improves by a factor of  $(2^2)^2 = 2^4$ . Additionally, we account for the probability that BCT $(7, \delta_3) \neq 0$  for the middle-round switch at the round 4 S-box operation.

Since  $\delta_3 \in 0xc$ , 0xa, 0x9, the probability of BCT $(7, \delta_3) \neq 0$  is lower bounded by the minimum of their respective probabilities, i.e.,

$$\Pr[\operatorname{BCT}(7, \delta_3) \neq 0] \ge \frac{\min\{\operatorname{BCT}(7, 0x8), \operatorname{BCT}(7, 0xa), \operatorname{BCT}(7, 0x9)\}}{2^4}$$
$$= \frac{\min\{2, 4, 4\}}{2^4} = 2^{-3},$$

where BCT(7, 0x8) = 2, BCT(7, 0xa) = 4, and BCT(7, 0xc) = 4. Finally, the refined probability for the boomerang becomes  $2^4 \cdot 2^{14} \cdot 2^{-60} \cdot 2^{-3} = 2^{-45}$ . This scenario can be further mapped to a Sandwich attack ( $E = E_1 \circ E_m \circ E_0$ ) with probability  $\bar{p}^2 \cdot r \cdot \bar{q}^2$ , where  $\bar{p} = 2^{-12}$ ,  $r = 2^{-3}$ , and  $\bar{q} = 2^{-9}$ . As a result, the data, time, and memory complexities of the distinguishing attack are reduced to  $2^{46}$  plaintexts,  $2^{46}$  XOR operations, and negligible memory, respectively.



Fig. 11: Middle Round Switching Effects using Truncated Differences

**Experimental Verification.** For this boomerang distinguisher, we have experimentally verified both the upper and lower differential characteristics along with their corresponding probabilities. The implementation used for verification is provided in [2].

## 6 Conclusion and Future Works

In this work, we present a comprehensive implementation of bit-oriented MILP models for the FUTURE lightweight block cipher in related-key settings. This approach can be extended to model MDS (or near-MDS) based SPN ciphers in the future. Utilizing this model, we explored related-key differential characteristics across different rounds, identifying a seven-round differential characteristic with a probability of  $2^{-46.4}$ . We further extended this characteristic by adding an extra round, providing a distinguisher with data complexity of  $2^{64}$ , time complexity of  $2^{63}$  XOR operations, and negligible memory requirements. Additionally, we developed a full-round boomerang distinguisher with a probability

of  $2^{-60}$  based on the round-reduced differential characteristics. By applying a one-round middle switch effect, we refined the boomerang's probability from  $2^{-60}$  to  $2^{-45}$ . Consequently, the complexities of the attack are improved to  $2^{46}$ plaintexts,  $2^{46}$  XOR operations, and negligible memory.

In future work, it would be valuable to explore optimizing the probability of the distinguisher, rather than focusing solely on the number of active S-boxes. This could potentially enhance the overall probability of the distinguisher. Additionally, recent advancements in automated tools for cryptanalysis present an opportunity to develop a tool for conducting truncated differential and sandwich attacks, capturing more dependencies in the middle rounds, and further improving the probabilities of differential and boomerang distinguishers. Lastly, another interesting direction for future research would be to propose an efficient key recovery attack based on the distinguishers presented in this work.

## References

- 1. Linear Programming Formulation With Gurobi Python API. https://www.gurobi.com/resources/ch4-linear-programming-with-python.
- 2. Unoptimized MILP Codes and Their Verifications using C. https://drive.google.com/drive/folders/1e5wZ6wy5xd8AZUV1v7E2PtBUzAg5c91.
- 3. Roberto Avanzi. The QARMA block cipher family. almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Trans. Symmetric Cryptol.*, 2017(1):4–44, 2017.
- 4. Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II, volume 9453 of Lecture Notes in Computer Science, pages 411–436. Springer, 2015.
- Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, volume 10529 of Lecture Notes in Computer Science, pages 321–345. Springer, 2017.
- Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015, pages 175:1–175:6. ACM, 2015.
- Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, volume 9815 of Lecture Notes in Computer Science, pages 123–153. Springer, 2016.

- Christof Beierle, Gregor Leander, Amir Moradi, and Shahram Rasoolzadeh. CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. *IACR Trans. Symmetric Cryptol.*, 2019(1):5–45, 2019.
- Eli Biham. New types of cryptanalytic attacks using related keys. J. Cryptol., 7(4):229– 246, 1994.
- Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack rectangling the serpent. In Birgit Pfitzmann, editor, Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding, volume 2045 of Lecture Notes in Computer Science, pages 340–357. Springer, 2001.
- 11. Eli Biham, Orr Dunkelman, and Nathan Keller. New results on boomerang and rectangle attacks. In Joan Daemen and Vincent Rijmen, editors, Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers, volume 2365 of Lecture Notes in Computer Science, pages 1–16. Springer, 2002.
- Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks. In Ronald Cramer, editor, Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, volume 3494 of Lecture Notes in Computer Science, pages 507–525. Springer, 2005.
- Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings, volume 537 of Lecture Notes in Computer Science, pages 2–21. Springer, 1990.
- 14. Alex Biryukov, Christophe De Cannière, and Gustaf Dellkrantz. Cryptanalysis of SAFER++. In Dan Boneh, editor, Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729 of Lecture Notes in Computer Science, pages 195-211. Springer, 2003.
- 15. Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings, volume 5912 of Lecture Notes in Computer Science, pages 1–18. Springer, 2009.
- 16. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultralightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, volume 4727 of Lecture Notes in Computer Science, pages 450–466. Springer, 2007.
- 17. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings, volume 7658 of Lecture Notes in Computer Science, pages 208–225. Springer, 2012.

- Christina Boura and Daniel Coggia. Efficient milp modelings for sboxes and linear layers of spn ciphers. *IACR Transactions on Symmetric Cryptology*, 2020(3):327–361, 2020.
- Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTAN-TAN - A family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings, volume 5747 of Lecture Notes in Computer Science, pages 272–288. Springer, 2009.
- 20. Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, Advances in Cryptology EUROCRYPT 2018 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 May 3, 2018 Proceedings, Part II, volume 10821 of Lecture Notes in Computer Science, pages 683–714. Springer, 2018.
- 21. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002.
- Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. J. Cryptol., 27(4):824–849, 2014.
- 23. Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers 23, pages 268–288. Springer, 2016.
- 24. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, Cryptographic Hardware and Embedded Systems CHES 2011 13th International Workshop, Nara, Japan, September 28 October 1, 2011. Proceedings, volume 6917 of Lecture Notes in Computer Science, pages 326–341. Springer, 2011.
- Kishan Chand Gupta, Sumit Kumar Pandey, and Susanta Samanta. Future: a lightweight block cipher using an optimal diffusion matrix. In *International Conference* on Cryptology in Africa, pages 28–52. Springer, 2022.
- Murat Burhan İlter and Ali Aydın Selçuk. Milp-aided cryptanalysis of the future block cipher. In International Conference on Information Technology and Communications Security, pages 153–167. Springer, 2022.
- John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and serpent. In Bruce Schneier, editor, Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings, volume 1978 of Lecture Notes in Computer Science, pages 75–93. Springer, 2000.
- Ting Li and Yao Sun. Superball: A new approach for MILP modelings of boolean functions. *IACR Trans. Symmetric Cryptol.*, 2022(3):341–367, 2022.
- Han Lin, Jian Zou, and Jiayin Li. The differential meet-in-the-middle attack on FU-TURE and CRAFT. In Proceedings of the 2023 13th International Conference on Communication and Network Security, ICCNS 2023, Fuzhou, China, December 6-8, 2023, pages 151–158. ACM, 2023.
- Sandip Kumar Mondal, Mostafizar Rahman, Santanu Sarkar, and Avishek Adhikari. Yoyo cryptanalysis on future. Int. J. Appl. Cryptogr., 4(3/4):238–249, 2024.
- 31. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *Information Security and*

Cryptology: 7th International Conference, Inscrypt 2011, Beijing, China, November 30–December 3, 2011. Revised Selected Papers 7, pages 57–76. Springer, 2012.

- Sean Murphy. The return of the cryptographic boomerang. IEEE Trans. Inf. Theory, 57(4):2517–2521, 2011.
- National Institute of Standards, Technology (NIST), and Quynh Dang. Secure hash standard (shs), 2012-03-06 00:03:00 2012.
- 34. Debranjan Pal, Vishal Pankaj Chandratreya, and Dipanwita Roy Chowdhury. New techniques for modeling sboxes: An MILP approach. In Jing Deng, Vladimir Kolesnikov, and Alexander A. Schwarzmann, editors, Cryptology and Network Security - 22nd International Conference, CANS 2023, Augusta, GA, USA, October 31 -November 2, 2023, Proceedings, volume 14342 of Lecture Notes in Computer Science, pages 318–340. Springer, 2023.
- Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
- Himadry Sekhar Roy, Prakash Dey, Sandip Kumar Mondal, and Avishek Adhikari. Cryptanalysis of full round FUTURE with multiple biclique structures. *Peer Peer Netw. Appl.*, 17(1):397–409, 2024.
- 37. Yu Sasaki and Yosuke Todo. 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, June 8–9, 2017, Revised Selected Papers 10, pages 150–165. Springer, 2017.
- André Schrottenloher and Marc Stevens. Simplified modeling of MITM attacks for block ciphers: New (quantum) attacks. *IACR Trans. Symmetric Cryptol.*, 2023(3):146– 183, 2023.
- Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In Bart Preneel and Tsuyoshi Takagi, editors, Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings, volume 6917 of Lecture Notes in Computer Science, pages 342–357. Springer, 2011.
- 40. Bing Sun, Zhiqiang Liu, Vincent Rijmen, Ruilin Li, Lei Cheng, Qingju Wang, Hoda Alkhzaimi, and Chao Li. Links among impossible differential, integral and zero correlation linear cryptanalysis. In Rosario Gennaro and Matthew Robshaw, editors, Advances in Cryptology CRYPTO 2015 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I, volume 9215 of Lecture Notes in Computer Science, pages 95–115. Springer, 2015.
- Ling Sun, Wei Wang, and Meiqin Wang. Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. *IET Inf. Secur.*, 14(1):12–20, 2020.
- 42. Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In Dongdai Lin, Shouhuai Xu, and Moti Yung, editors, Information Security and Cryptology 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27-30, 2013, Revised Selected Papers, volume 8567 of Lecture Notes in Computer Science, pages 39–51. Springer, 2013.
- 43. Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai 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. Cryptology ePrint Archive, Paper 2014/747, 2014. https://eprint.iacr.org/2014/747.

- 44. Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. 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, December 7-11, 2014. Proceedings, Part I 20, pages 158–178. Springer, 2014.
- 45. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. \$\textnormal{\textsc{TWINE}}\$: A lightweight block cipher for multiple platforms. In Lars R. Knudsen and Huapeng Wu, editors, Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers, volume 7707 of Lecture Notes in Computer Science, pages 339–354. Springer, 2012.
- 46. Aleksei Udovenko. MILP modeling of boolean functions by minimum number of inequalities. *IACR Cryptol. ePrint Arch.*, page 1099, 2021.
- 47. David A. Wagner. The boomerang attack. In Lars R. Knudsen, editor, Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings, volume 1636 of Lecture Notes in Computer Science, pages 156–170. Springer, 1999.
- 48. Jun Yin, Chuyan Ma, Lijun Lyu, Jian Song, Guang Zeng, Chuangui Ma, and Fushan Wei. Improved cryptanalysis of an ISO standard lightweight block cipher with refined MILP modelling. In Xiaofeng Chen, Dongdai Lin, and Moti Yung, editors, Information Security and Cryptology 13th International Conference, Inscrypt 2017, Xi'an, China, November 3-5, 2017, Revised Selected Papers, volume 10726 of Lecture Notes in Computer Science, pages 404–426. Springer, 2017.

# Appendix A

Μ

	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1]
	1	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0
	0	1	1	0	0	1	0	0	0	0	1	0	0	1	1	0
	0	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0
	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1
	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0
	1	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0
	1	0	0	1	1	0	0	0	0	0	1	1	0	0	1	1
1 =	0	1	0	0	1	1	0	0	1	0	0	1	0	0	0	1
	0	0	1	0	0	1	1	0	1	1	0	0	1	0	0	0
	1	0	0	1	1	0	1	1	0	1	1	0	0	1	0	0
	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	1
	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	0
	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0
	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0
	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0	1

Fig. 12: The Primitive Representation of  $\mu$  When Cipher's State is Represented from MSB to LSB

	Го	1	0	0	1	1	0	0	1	0	0	0	0	1	0	0]	
	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	0	
	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	
	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	
	1	0	0	1	0	0	0	1	1	1	0	0	1	1	0	0	
	1	1	0	1	1	0	0	1	0	0	1	0	0	0	1	0	
	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1	
M <sub>2</sub> =	0	0	1	1	0	0	1	0	1	0	0	0	1	0	0	0	
	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	0	
	1	0	0	1	1	1	0	1	0	1	1	0	0	0	1	0	
	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	
	0	0	1	0	0	0	1	1	1	0	0	1	1	0	0	0	
	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	
	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	0	
	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	
	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	

Fig. 13: The Primitive Representation of  $\mu$  When Cipher's State is Represented from LSB to MSB



Fig. 14: 16 Constraints Correspond to One Column Transformation After the MixColumn Operation

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	4	0	0	0	0	0	4	4	0	0	0	0	0
2	0	4	0	4	0	2	0	2	0	0	0	0	2	0	2	0
3	0	0	0	4	2	0	2	0	0	0	4	0	0	2	0	2
4	0	0	0	0	4	0	4	0	0	0	0	0	0	4	0	4
5	0	0	0	0	$\overline{2}$	$\overline{2}$	$\overline{2}$	$\overline{2}$	0	0	0	0	2	2	2	2
6	0	4	0	4	0	2	0	2	0	0	0	0	2	0	2	0
7	0	0	4	0	0	2	0	2	0	4	0	0	2	0	2	0
8	0	0	0	0	2	0	2	0	4	2	0	2	4	0	0	0
9	0	2	2	0	0	2	2	0	0	0	2	2	0	0	2	2
10	0	0	0	0	0	4	0	0	4	2	0	2	0	2	0	2
11	0	2	2	0	0	0	2	2	0	0	2	2	2	0	0	2
12	0	0	0	0	2	0	2	0	4	2	0	2	0	0	4	0
13	0	2	2	0	2	0	0	2	0	0	2	2	2	2	0	0
14	0	0	0	0	0	0	0	4	4	2	0	2	0	2	0	2
15	0	2	2	0	2	2	0	0	0	0	2	2	0	2	2	0

(a) DDT of S-box

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

(b) Inverse DDT of S-box

Round 0		Stat	e(x)		]		Ke	ey0				Stat	te(x)		]		Stat	e(x)						
	0	8	1	0	1	0	8	1	0		0	0	0	0	Pr = 1	0	0	0	0					
	0	0	1	0		0	0	1	0		0	0	0	0	sc	0	0	0	0	SR o MC				
	0	0	8	0		0	0	8	0		0	0	0	0		0	0	0	0					
	0	0	0	8	1	0	0	0	8		0	0	0	0		0	0	0	0					
Round 1					,					,					, 					,				
		Stat	e(x)		]		Ke	ey1				Stat	te(x)		]		Stat	e(x)						
	0	0	0	0		0	0	0	0		0	0	0	0	$Pr = 2^{-3}$	0	0	0	0					
	0	0	0	0		0	0	0	0		0	0	0	0	sc→	0	0	0	0	SR o MC				
	0	0	0	0		0	0	0	0		0	0	0	0	1	0	0	0	0					
	0	0	0	0	]	0	0	b	0	]	0	0	b	0	]	0	0	2	0					
Г																								
Round 2		Stat	e(x)		]		Ke	ey2				Stat	te(x)				Stat	e(x)						
	0	0	3	0		0	0	3	0		0	0	0	0	Pr = 1	0	0	0	0					
Ļ	0	0	0	1		0	0	0	1	$\rightarrow$	0	0	0	0	sc	0	0	0	0					
-	1	0	0	0		1	0	0	0		0	0	0	0		0	0	0	0					
	0	2	0	0	]	0	2	0	0		0	0	0	0	]	0	0	0	0					
Round 3																								
		Stat	e(x)		]		Ke	ey3				Stat	te(x)				Stat	e(x)		]				
	0	0	0	0		0	0	0	0		0	0	0	0	Pr = 2	0	0	0	0					
L	0	0	0	0	Ш Д	0	0	1	0	$\rightarrow$	0	0	1	0	$\xrightarrow{sc}$	0	0	a	0	SR o MC				
	0	0	0	0		0	0	6	0		0	0	6	0		0	0	5	0					
	0	0	0	0		0	0	0	0		0	0	0	0		0	0	0	0					
Round 4																								
		Stat	e(x)		]		Ke	ey4				Stat	te(x)		D. 0-16		Stat	e(x)						
	0	0	0	0		0	0	0	2		0	0	0	2	Pr = 2	0	0	0	с					
L,	0	0	0	с	] Д	2	0	0	0	$\rightarrow$	2	0	0	с	$\xrightarrow{sc}$	1	0	0	8	SR o MC				
	3	0	0	0		0	2	0	0		3	4	0	0		4	f	0	0					
	0	b	0	0		0	3	0	0	]	0	d	0	0	]	0	с	0	0					
Round 5																								
		Stat	e(x)				Ke	ey5				Stat	te(x)		$p_{n} = 2^{-26}$		Stat	e(x)						
	d	0	0	e		0	0	2	0		d	0	2	e	1 - 2	7	0	e	8					
Ļ	4	0	b	0	A	0	0	с	0	$\rightarrow$	4	0	7	0	$\xrightarrow{sc}$	f	0	9	0	SR o MC				
	0	0	5	4		0	0	0	0		0	0	5	4		0	0	7	4					
	b	0	2	f		0	0	0	0		b	0	2	f		b	0	7	1					
Round 6																								
		Stat	e(x)		]		Ke	ey6				Stat	te(x)		$p_{r} = 2^{-8}$		Stat	e(x)		]		Stat	te(x)	
	4	0	e	0		2	0	0	0		0	0	e	0		0	0	9	0		0	0	7	3
L,	0	8	0	0	Ð	0	8	0	0	$\rightarrow$	0	0	0	0	$\xrightarrow{sc}$	0	0	0	0	SR 0 MC	1	0	0	9
	0	с	0	0		0	с	0	0		0	0	0	0		0	0	0	0		0	1	0	0
	0	0	3	f		0	0	4	0	]	0	0	7	f	]	0	0	2	2	]	0	f	2	0

Fig. 15: Seven Round Related Key Differential Characteristic of FUTURE Cipher



Fig. 16: 17 Number of Constraints to Capture DDT of FUTURE S-box



Fig. 17: 17 Number of Constraints to Capture DDT of PRESENT S-box