A MIQCP-Based Automatic Search Algorithm for Differential-Linear Trails of ARX Ciphers

Guangqiu Lv[1], Chenhui Jin[1] and Ting Cui[1]

[1] PLA SSF Information Engineering University, Zhengzhou 450000, China
[2] LGQ_Running@163.com

Abstract. Differential-linear (DL) cryptanalysis has undergone remarkable advancements since it was first proposed by Langford and Hellman [LH94] in 1994. At CRYPTO 2022, Niu et al. studied the (rotational) DL cryptanalysis of n-bit modulo additions with 2 inputs, i.e., $\boxplus_2$, and presented a technique for evaluating the (rotational) DL correlation of ARX ciphers. However, the problem of how to automatically search for good DL trails on ARX with solvers was left open, which is the focus of this work.

In this paper, we solve this open problem through some techniques to reduce complexity and a transformation technique from matrix multiplication chain to Mixed Integer Quadratically-Constrained Programs (MIQCP). First, the computational complexity of the DL correlation of $\boxplus_2$ is reduced to approximately one-eighth of the state of art, which can be computed by a 2×2 matrix multiplication chain of the same length as before. Some methods to further reduce complexity in special cases have been studied. Additionally, we present how to compute the extended (rotational) DL correlations of $\boxplus_k$ for $k \geq 2$, where two output linear masks of the cipher pairs can be different. Second, to ensure that the existing solver Gurobi[1] can compute DL correlations of $\boxplus_2$, we propose a method to transform an arbitrary matrix multiplication chain into a MIQCP, which forms the foundation of our automatic search of DL trails in ARX ciphers. Third, in ARX ciphers, we use a single DL trail under some explicit conditions to give a good estimate of the correlation, which avoids the exhaustion of intermediate differences. We then derive an automatic method for evaluating the DL correlations of ARX, which we apply to Alzette and some versions of SPECK. Experimentally verified results confirm the validity of our method, with the predicted correlations being close to the experimental ones. To the best of our knowledge, this method finds the best DL distinguishers for these ARX primitives currently. Furthermore, we presented the lowest time-complexity attacks against 12-14 rounds of SPECK32 to date.

Keywords: Automatic cryptanalysis · Differential-linear cryptanalysis · ARX · SPECK · Alzette

1 Introduction

The two main classes of cryptanalysis are the linear and differential attacks. Differential-linear (DL) cryptanalysis is to employ two most important cryptanalysis (differential and linear attacks) to enhance the effectiveness of the individual attacks. Let the cipher be presented as a composition $E = E_1 \circ E_0$ of two parts. The idea of DL cryptanalysis is to apply a (truncated) differential attack and a linear attack on the first part E_0 and the second part E_1, respectively, and then combine them to a single distinguisher over

[1] The solver used in this paper is Gurobi, and some ready-made functions in Gurobi are also used, such as LOG_2 and ABS. The source code is available at https://.

the cipher. DL cryptanalysis was first proposed by Langford and Hellman [LH94] in 1994 to analyse DES. In recent years, we have witnessed remarkable advancements in the development of DL cryptanalysis.

Differential-linear Cryptanalysis. The correlation of an ordinary differential-linear approximation (Δ_{in}, Γ) of the vectorial Boolean function $E : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as $Cor(\Delta_{in}, \Gamma) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\Gamma \cdot (E(x) \oplus E(x \oplus \Delta_{in}))}$, where $\Delta_{in} \in \mathbb{F}_2^n$ and $\Gamma \in \mathbb{F}_2^m$. A classic and trivial analysis method is depicted in Figure 1(a). Let (Δ_{in}, Δ_m) be a differential trail with probability p, and (γ, Γ) be a linear trial for E_1 with correlation q. Then, the overall correlation of DL distinguisher can be estimated with the piling-up lemma [Mat93] as

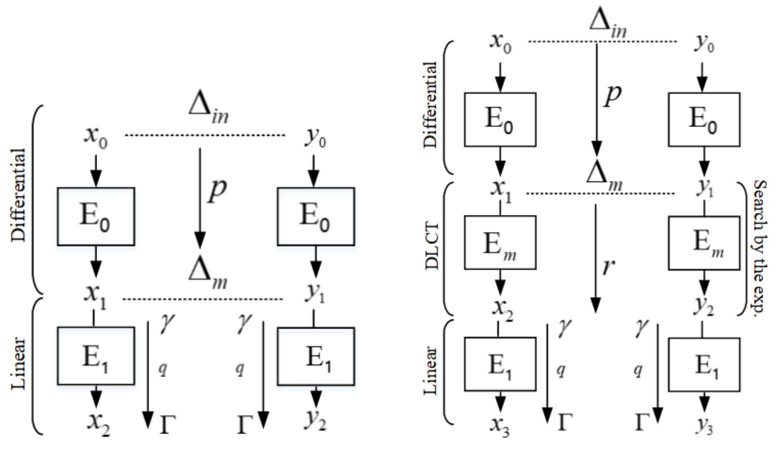
$$Cor = 2 \times \Pr[\Gamma \cdot (E(x_0) \oplus E(y_0))] - 1 = pq^2, \tag{1}$$

since $\Gamma \cdot (E(x_0) \oplus E(y_0)$ can be decomposed into the XOR sum of three terms $\gamma \cdot (E_0(x_0) \oplus E_0(y_0))$, $\gamma \cdot E_0(x_0) \oplus \Gamma \cdot E(x_0)$ and $\gamma \cdot E_0(y_0) \oplus \Gamma \cdot E(y_0)$. The above equation relies on the following two assumptions[LLL21, LSL21].

Assumption 1. *E_0 and E_1 are independent.*

Assumption 2. $\Pr[\gamma \cdot (E_0(x_0) \oplus E_0(y_0)) = 0 | E_0(x_0) \oplus E_0(y_0) \neq \Delta_m] = \frac{1}{2}$

However, it has been observed that Assumption 2 may fail in many cases. In [BLN17], Blondeau et al. presented an exact expression of the correlation in a closed form under Assumption 1 and found that it is possible to state some explicit assumptions under which a single DL trail gives a good estimate of the correlation. However, the exact expression is computationally infeasible due to the need of exhausting all intermediate masks.



(a) The classical analysis of an ordinary differential-linear distinguisher.

(b) A differential-linear distinguisher with Differential-linear Connectivity Table (DLCT). Generally, the correlation of DL trial (Δ_m, γ), denoted as r, is determined by experiments.

Figure 1: Differential-linear cryptanalysis

Table 1: A summary of the results. R-DL = rotational differential-linear, DL = differential-linear, LC = linear characteristic, DC = differential characteristic, DL(NB) = a distinguisher combining DL trail with neutral bit technique of differential propagation. We show differentials with probabilities and LC/DL/R-DL/DL(NB) with correlations.

Permutation	Type	Round	Probability/Correlation		Ref.
			Theory	Exp.	
Alzette	R-DL	4	$2^{-11.37}$	$2^{-7.35}$	[LSL21]
	DL	4	$2^{-0.27}$	$2^{-0.1}$	[LSL21]
	DC	8	$\leq 2^{-32}$	-	[BBCdS+20]
	LC	8	$2^{-15.79}$	-	[BBCdS+20]
	DL	8	$-2^{-8.24}$	$-2^{-5.50}$	[NSLL22]
	DC	9	$\leq 2^{-36}$	-	[BBCdS+20]
	DC	10	$\leq 2^{-42}$	-	[BBCdS+20]
	DL	8	$2^{-4.14}$	$2^{-4.06}$	**Our**
	DL	9	$-2^{-10.08}$	$-2^{-7.60}$	
	DL	10	$2^{-11.00}$	$2^{-10.48}$	
SPECK32	DC	10†	2^{-34}	-	[SWW21]
	LC	10†	2^{-17}	-	[SWW21]
	DL	10	$2^{-15.23}$	$2^{-13.90}$	[NSLL22]
	DL(NB)[3]	10	-2^{-11}	-	[BGG+23]
	DC	11†	2^{-38}	-	[SWW21]
	LC	11†	2^{-19}	-	[SWW21]
	DL(NB)	11	-2^{-14}	-	[BGG+23]
	DL	10	$-2^{-13.37}$	$-2^{-11.58}$	**Our**
	DL(NB)	10	$-2^{-8.58}$	-	
	DL[1]	11	$-2^{-19.37}$	$-2^{-17.09}$	
	DL[1]	11	$-2^{-18.37}$	$-2^{-16.68}$	
	DL(NB)	11	$-2^{-11.09}$	-	
SPECK48	DC	11†	2^{-45}	-	[SWW21]
	LC	11†	2^{-25}	-	[SWW21]
	DL	11	$-2^{-20.46}$	$-2^{-17.55}$	**Our**
SPECK64	DC	11†	2^{-42}	-	[SWW21]
	LC	11†	2^{-24}	-	[SWW21]
	DC	12 †	2^{-46}	-	[SWW21]
	LC	12 †	2^{-27}	-	[SWW21]
	DL	11	$2^{-22.13}$	$2^{-19.44}$	**Our**
	DL	12	$2^{-26.93}$	-	

[1] We random chose 2^8 master keys and compute the average DL correlation by going though the full plaintext space. For random permutation, the experimental correlation should be about $\pm 2^{-20}$. This information leakage can be used to distinguish 11-round SPECK32 from random functions, if given the encrypted ciphertext under multiple random keys. Moreover, given sufficient neutral bits of top short-round differential, it can be converted into a new valid distinguisher for key recovery, denoted by DL(NB). See Section 2.2 and 5.2 for more details.

[2] Entries marked with † is the optimal single differential/linear trail.

[3] To compare with the DL trails without using NBs, we regard the correlations of DL(NB) as $p^{\frac{1}{2}} r q^2$, since the data complexity required is $\mathcal{O}(pr^{-2}q^{-4})$.

Aiming at Assumption 1, Bar-On et al. [BODKW19] found that the dependency between the two subciphers significantly affects the complexity of the DL attack and proposed Differential-Linear Connectivity Table (DLCT) to take into account the dependency. They divided the cipher E into three subciphers E_0, E_m, and E_1 such that $E = E_1 \circ E_m \circ E_0$, and the middle part E_m is experimentally evaluated, as depicted in Figure 1(b). The correlation of DL trail (Δ_m, γ) for E_m is denoted by r. They assumed that the empirical correlations

Table 2: The key recovery attacks of reduced-round SPECK32/64. Diff. = differential, ND = neural distinguisher, GoogLeNet = a distinguisher using GoogLeNet neural network, DL(NB) = a distinguisher combining DL trails with the neutral bit technique of differential propagation.

R	KeySpace	Type	Data	Time	Ref.
12/22	2^{64}	Diff.†	$2^{30.42}$	$2^{33.84}$	[BdST+22]
		ND†	$2^{18.5}$	$2^{43.3}$	[BGL+23]
		GoogLeNet†	2^{25}	$2^{42.3}$	[ZWW22]
		DL(NB)	2^{19}	2^{34}	**Our**
13/22	2^{64}	DL(NB)	2^{24}	2^{52}	[BGG+23]
		Diff.	$2^{31.13}$	$2^{50.16}$	[BdST+22]
		DL(NB)	2^{19}	2^{50}	**Our**
	2^{63}	ND†	2^{29}	$2^{51.5}$	[BGL+23]
		GoogLeNet†	2^{31}	$2^{49.8}$	[ZWW22]
		DL(NB)	2^{25}	2^{41}	**Our**
14/22	2^{64}	Diff.	$2^{60.99}$	$2^{31.75}$	[BdST+22]
		DL(NB)	2^{31}	2^{58}	[BGG+23]
	2^{63}	DL(NB)	2^{25}	2^{57}	**Our**

[1] Entries marked with † (resp. without †) are practical (resp. theoretical) attacks.

obtained by sampling for a sufficiently large number of messages closely match the actual correlations. For the DL cryptanalysis of ARX ciphers, Beierle et al. [BLT20] combined the DL attack with the neutral bit technique for differential part and the partitioning technique for linear part to further reduce the attack complexity.

Though DLCT can be constructed efficiently using the Fast Fourier Transform, a good DL trail (Δ_m, γ) for E_m is usually found experimentally at present, which is due to the huge computation complexity of DLCT. Liu et al. [LLL21] introduced a technique called Differential Algebraic Transitional Form (DATF) for DL cryptanalysis, where DLCT is no longer used. The DATF technique is applicable to ciphers with low algebraic degree of round function, but not to ARX ciphers. The algebraic degree of ARX ciphers is usually high after only a very few rounds as the carry bit within one modular addition already reaches almost maximal degree.

For the theoretical estimation of DL correlation in ARX ciphers, Liu et al. [LSL21] introduced Morawiecki et al.'s technique [MPS13], which is called correlation propagation of difference bits in this paper, to compute the DL correlations in ARX ciphers. They presented the so-called rotational DL cryptanalysis and proposed the open problem that how to compute DL correlation of ARX cihpers with arbitrary output linear masks. To solve this problem, Niu et al. [NSLL22] introduced a chain of 4 × 4 matrix multiplications to compute the DL correlation of modulo additions with arbitrary output linear masks and combined this technique with the correlation propagation technique of difference bits to compute the DL correlation for ARX ciphers. [NSLL22] pointed that the major pain spot of the current development is that

"there is no effective tool that can automatically search for good DL approximations, and thus in practice the search space is severely limited to low Hamming weight output masks."

In this paper, we partially solve this open problem through some techniques to reduce complexity and a transformation technique from matrix multiplication chain to MIQCP.

ARX ciphers and differential/linear cryptanalysis. ARX is an abbreviation for addition (modulo a power of two), word-wise rotation and XOR, and ARX ciphers are generally efficient in software since the above three operations have underlying hardware support in almost all general-purpose processors. For differential and linear attack, the algorithms to efficiently compute differential probability and linear correlation of modulo additions were presented in [LM01] and [Wal03], respectively. Now, there are quite powerful tools (e.g., MILP, SAT, or SMT) to analyze ARX primitives[FWG$^+$16, SWW21].

Our Contributions. This paper achieves the automatic search of DL trails in ARX ciphers step by step, and has three main contributions.

(1) Extended (rotational) differential-linear cryptanalysis of $\boxplus_k$ for $k \geq 2$ and the round function in ARX ciphers. The computational complexity of the DL correlation of $\boxplus_2$ is reduced to approximately one-eighth of [NSLL22], which can be computed by a simple chain of 2×2 matrix multiplications. Furthermore, we reduced the computational complexity in special cases, such as when the DL correlation is equal to 0 or ±1. What's more, we presented how to compute the extended (rotational) differential-linear correlation of $\boxplus_k$ for $k \geq 2$, where the two output linear masks of the left and right branches are different. To verify our results, we have conducted experiments under all extended DL approximations on the 4-bit additions $\boxplus_2^4$ and $\boxplus_3^4$. The results confirm the effectiveness of our approach.

(2) Transformation from Arbitrary Matrix Multiplication Chain to MIQCP. To ensure that the existing solver can compute DL correlation of $\boxplus_2$, we proposed a method to transform an arbitrary matrix multiplication chain into the Mixed Integer Quadratically-Constrained Programs (MIQCP), which serves as the foundation of our automatic search of DL trails in ARX ciphers. This technique has significant potential. For instance, when used in automatic searching for linear approximations of $\boxplus_3$, the above method can accurately compute the correlation of linear approximations, whereas the widely used method currently splits $\boxplus_3$ into two $\boxplus_2$ operations, which yields a less precise value.

(3) An automatic search algorithm for DL trails in ARX ciphers and its application. In ARX ciphers, we use a single DL trail under some explicit conditions to give a good estimate of the correlation. We then developed an automatic method for evaluating the DL correlation of ARX ciphers, which we applied to Alzette and some versions of SPECK. The improved results were experimentally verified to confirm the validity of our method, with the predicted correlations being close to the experimental ones. To the best of our knowledge, this method finds the best differential-linear distinguishers for these ARX primitives. Additionally, we presented the lowest time-complexity attacks against 12-14 rounds of SPECK32 to date.

Outline. In Section 2, we introduce notations and preliminaries for DL cryptanalysis. In Section 3, we presented the extended (rotational) DL cryptanalysis of $\boxplus_k$ for all $k \geq 2$ and the round function in ARX ciphers. In Section 4, we proposed a method to transform an arbitrary matrix multiplication chain into MIQCP and an automated method of searching for DL trials in ARX ciphers. In Section 5, the above method is applied to Alzette and some versions of SPECK, and all improved results are experimentally verified. Section 6 concludes the paper with some open problems.

2 Notations and Preliminaries

The notations we use in this paper are summarised in Table 3.

Table 3: Notations.

Symbol	**Description**
$Floor(x)$	The maximum integer not larger than x.
$\boxplus_k^n(x_0,\dots,x_{k-1})$	The addition function $\boxplus_k^n(x_0,\dots,x_{k-1})=(x_0+\cdots+x_{k-1}) \mod 2^n$. When n is clear from the context, it is written as $\boxplus_k$ for simplicity. $\boxplus_k^n(x_0,\dots,x_{k-1})$ is also called n-bit modulo addition or addition modulo 2^n.
$\boxplus_{k,d\blacktriangleleft c}^n(x_0,\dots,x_{k-1})$	The addition function $\boxplus_{k,d\blacktriangleleft c}^n(x_0,\dots,x_{k-1})=(x_0+\cdots+x_{k-1}+c) \mod 2^n$. c and d are called as the initial carry and the most significant carry, respectively. Here $x_0+\cdots+x_{k-1}+c=d2^n+\boxplus_{k,d\blacktriangleleft c}^n(x_0,\dots,x_{k-1})$ and $c,d\in\{0,1,\dots,k-1\}$. Similarly, $\boxplus_{k,*\blacktriangleleft c}^n(x_0,\dots,x_{k-1})=(x_0+\cdots+x_{k-1}+c) \mod 2^n$ and there is no requirement for the most significant carry.
$\hat{c}_a(x_0,x_1,...,x_{k-1})$	The most significant carry of addition, i.e., $\hat{c}_a(x_0,x_1,...,x_{k-1})=Floor((x_0+\cdots+x_{k-1}+a)/2^n)$.
$x[i]$	The i-th bit of x, written as x_i for simplicity. x_{n-1} (resp. x_0) is the most (resp. least) significant bit of x.
$HIndex(x)$	The integer such that $x[j]=0$ for $HIndex(x)<j<n$ and $x[HIndex(x)]=1$.
$\lfloor x\rfloor_{(t)}$	The least significant t bits of x, i.e., $(x_{t-1},...,x_0)$
$\lceil x\rceil^{(t)}$	The most significant t bits of x, i.e., $(x_{n-1},...,x_{n-t})$
$x\lll t$	Rotation of x by t-bit to the left, written as $\overleftarrow{x}$ for simplicity.
$x\ggg t$	Rotation of x by t-bit to the right, written as $\overrightarrow{x}$ for simplicity.
$\cdot$	The inner product of two vectors.
$\#\mathcal{X}$	The size of a set $\mathcal{X}$.
$HW(x)$	The Hamming weight of x.
$BW(x)$	The bit width of x.
$\Pr[x=0]$	Probability that x equals 0.
$x\Vert y$	Concatenation operation. x_{n-1} is the most significant bit of the new binary string.
$0^t/1^t/*^t$	A vector with t zeros (resp. ones/$*$). Here $*$ indicates that there is no limit, i.e., the bit can be either 1 or 0.
$A[x,y]$	The entry of matrix A in row x and column y, written as $A_{x,y}$ for simplicity.
$\lvert x\rvert$	The absolute value of x.

In this paper, the input differences and output masks are represented in hexadecimal.

2.1 Preliminaries

Lemma 1 (Piling-up Lemma [Mat93])**.** *Let $X_0,X_2,\cdots,X_{n-1}$ be n independent binary random variables with $\Pr[X_i=0]=p_i$. Then, it holds that*

$$2\Pr[X_0\oplus\cdots\oplus X_{n-1}=0]-1=\prod_{i=0}^{n-1}(2p_i-1).$$

Definition 1 (Correlation [BLN17, Mat93])**.** *Let (λ^0,λ^1) be a linear approximation of Boolean function $f:\mathbb{F}_2^n\to\mathbb{F}_2$. The correlation of the linear approximation (λ^0,λ^1) is defined as*

$$Cor_f(\lambda^0,\lambda^1)=2^{-n}\sum_{x\in\mathbb{F}_2^n}(-1)^{\lambda^0\cdot x\oplus\lambda^1\cdot f(x)}$$

.

When f is clear from the context, we may denote $Cor(\lambda^0,\lambda^1)$ as the correlation of linear approximation (λ^0,λ^1).

Definition 2 (Extended Rotational Differential-Linear (ERDL) Correlation)**.** *Let $S:\mathbb{F}_2^n\to\mathbb{F}_2^m$ be a vectorial Boolean function. Denote an extended rotational differential-linear approximation of S by $(t,\alpha,\lambda^0,\lambda^1)$. Then the correlation is defined as*

$$Cor(t,\alpha,\lambda^0,\lambda^1)=\frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n}(-1)^{\lambda^0\cdot(S(x)\lll t)\oplus\lambda^1\cdot S((x\lll t)\oplus\alpha)}, \tag{2}$$

where t, $\alpha \in \mathbb{F}_2^n$ and $\lambda^0, \lambda^1 \in \mathbb{F}_2^n$ are the rotational offset, rotational difference and output linear masks of the extended rotational differential-linear approximation, respectively.

Equation (2) is a generalization of the (rotational) differential-linear cryptanalysis [LSL21] and the extended differential-linear cryptanalysis [CY21]. When $\lambda^0 = \lambda^1$, Equation (2) computes the ordinary rotational differential-linear correlation [LSL21, NSLL22] of S, denoted by $Cor(t, \alpha, \lambda^0)$. When $t = 0$, Equation (2) computes the extended differential-linear correlation [CY21] of S, denoted by $Cor(\alpha, \lambda^0, \lambda^1)$. When $\lambda^0 = \lambda^1$ and $t = 0$, Equation (2) computes the ordinary differential-linear correlation [LH94, BODKW19] of S, denoted by $Cor(\alpha, \lambda^0)$.

In addition, Equation (2) is the definition of the correlation of a DL approximation, which specifies only the input differences and output masks. In this paper, the correlation of a DL approximation is estimated by one DL trail with the same input differences and output masks, where the DL trail additionally specifies the intermediate differences or linear masks relative to the DL approximation.

Though this paper focuses on the automated search of differential-linear trails of ARX, we present a complete extended (rotational) differential-linear cryptanalysis of the primitive $\boxplus_k$ for $k \geq 2$ in Section 3.

2.2 Improvement upon the DL Distinguisher—Neutral Bit

Let us be given a cipher $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$. This section explains how to use a DL approximation with advantage slightly less than $2^{-\frac{n}{2}}$ to construct an effective distinguisher, as long as sufficient neutral bits are given. This shows that DL trails with advantage less than $2^{-\frac{n}{2}}$ may be of great significance for recovering keys.

This section reviews neutral bit technique of differential propagation [BLT20, BGL+23][1], which can reduce the required data complexity $\mathcal{O}(p^{-2}r^{-2}q^{-4})$ to $\mathcal{O}(p^{-1}r^{-2}q^{-4})$. In a usual DL attack on a permutation $E : \mathbb{F}_2^n \to \mathbb{F}_2^n$ as explained in Figure 1(b), we divide the cipher E into three subciphers E_0, E_m, and E_1 such that $E = E_1 \circ E_m \circ E_0$. Let (Δ_{in}, Δ_m) be a differential trail of E_0 with probability p, and (Δ_m, Γ) be a DL trial for $E_1 \circ E_m$ with correlation rq^2, i.e., (Δ_{in}, Γ) is one DL trail of E with correlation prq^2. Generally, the attacker can distinguish the cipher E from a random permutation by preparing $\mathcal{O}(p^{-2}r^{-2}q^{-4})$ chosen plaintexts.

The following is the definition of neutral bits of a differential. It indicates that we can use one neutral bit generate another one confirming pair from one known confirming pair at no cost.

Definition 3 (Neutral bits of a differential (NB) [BC04])**.** *Let $e_i = 0x1 \lll i \in \mathbb{F}_2^n$. A differential of* E *is denoted by $(\Delta_{in}, \Delta_{out})$, and the input plaintext pair and output ciphertext pair of* E *are denoted by (p, p') and (c, c'), respectively. If $p \oplus p' = \Delta_{in}$ and $c \oplus c' = \Delta_{out}$, (p, p') is said to confirm the differential $(\Delta_{in}, \Delta_{out})$. The i-th bit is called a NB for the differential $\Delta_{in} \xrightarrow{\text{E}} \Delta_{out}$ if $(p \oplus e_i, p' \oplus e_i)$ is also a confirming pair for any confirming pair (p, p').*

Definition 4 (Probabilistic NB (PNB) [AFK+08])**.** *Let $e_i = 0x1 \lll i \in \mathbb{F}_2^n$. The i-th bit is a p-PNB of the differential $(\Delta_{in}, \Delta_{out})$ if $(P \oplus e_i, P' \oplus e_i)$ conforms to the differential with probability p for any confirming pair (P, P').*

We call the probability p the neutral probability of PNBs. In sequel attacks, the higher the probability p is, the higher the neutrality quality, and the more useful the neutral bit becomes.

[1] In [BLT20] and [BGL+23], neutral bit technique has different names. For clarity, we call it neutral bit technique of differential propagation in this paper. Note that there is also a technology called the neutral bit technique of linear part in [BLT20]. Only the neutral bit technique of differential propagation is used in this paper.

Definition 5 (Conditional (simultaneous-) NB(-set)s (CSNBS) [BGL+23])**.** *Let $I_s = i_1, i_2, \cdots, i_s$ be a set of indices. Denote $f_{I_s} = \oplus_{i \in I_s} e_i$. Let $\mathcal{C}$ be a set of constraints on the value of an input p, and let $\mathcal{P}_{\mathcal{C}}$ be the set of inputs that fulfill the constraints $\mathcal{C}$. The bit-set I_s is called a CSNBS for the differential $\Delta_{in} \to \Delta_{out}$, if for any conforming pair $(p, p' | p \in \mathcal{P}_{\mathcal{C}})$, $(p \oplus f_{I_s}, p' \oplus f_{I_s})$ is also a conforming pair.*

To amplify the correlation of DL trail (Δ_{in}, Γ), let us be given m neutral bits of the differential (Δ_{in}, Δ_m). If $2^m > r^{-1}q^{-2}$, the DL distinguisher (Δ_{in}, Γ) would work as follows:

1. Randomly generate a pair of plaintext (x_0, x_0'), where $x_0' = x_0 \oplus \Delta_{in}$. Then we use m neutral bits to generate 2^m pairs of plaintext $X = \{(x_0, x_0'), (x_1, x_1') \ldots (x_{2^m-1}, x_{2^m-1}')\}$.

2. Use the cipher E encrypt X and compute

$$Cor = \frac{1}{2^m} \sum_{0 \le i < 2^m} (-1)^{\Gamma \cdot (E(x_i) \oplus E(x_i'))}.$$

3. If we observe a correlation of rq^2 using 2^m pairs, the distinguisher succeeded. Otherwise, start over with Step 1.

Note that $2^m > r^{-1}q^{-2}$. With probability p, all plaintext pairs of X make the output difference of E_0 be Δ_m. In that case, the distinguisher succeeds in step 3. Thus, the data complexity of (Δ_{in}, Γ) required is $\mathcal{O}(p^{-1}r^{-2}q^{-4})$ instead of $\mathcal{O}(p^{-2}r^{-2}q^{-4})$.

In this paper, the differential-linear trails that can use neutral bit technique to reduce data complicity are called **DL(NB)** for short. The core idea of DL(NB) is to perform statistical analysis on all ciphertexts that ensure the establishment of the top short-round differential trail. To compare with the DL trails without using NBs, we regard the correlations of DL(NB) as $p^{\frac{1}{2}}rq^2$, since the data complexity required is $\mathcal{O}(pr^{-2}q^{-4})$. The availability of a DL(NB) is simultaneously determined by the probability of a prepended short-round differential, the correlation of the bottom DL trail and the number of neural bit(-set)s. However, for a sole differential-linear distinguisher, the availability is only determined by the whole correlation. Thus, the differential-linear (DL) distinguisher utilizing neutral bits, also known as DL(NB), can be viewed as a novel type of distinguisher. All DL(NB)s used in this paper are listed in Table 6.

3 Differential-Linear Cryptanalysis of $\boxplus_k$ and the Round Function of ARX ciphers

First of all, we study the extended differential-linear cryptanalysis of additions with k inputs for $k \ge 2$, i.e., $\boxplus_k$. Specially, we reduce the computational complexity of ordinary differential-linear correlation of $\boxplus_2$ to one-eighth of [NSLL22]. Some methods to further reduce complexity in special cases, e.g., the DL correlation is equal to 0 or ±1, have been studied. We also present the extended rotational differential-linear cryptanalysis of $\boxplus_k$ for $k \ge 2$. See Appendix B for more details.

The above focus on the analysis of modulo additions. Second, we further studied how to compute the extended differential-linear correlation of round function $S(x^0, x^1, ..., x^{k-1}) = (\boxplus_k(x^0, x^1, ..., x^{k-1}), x^1, x^2, ..., x^{k-1})$, which is widely used in ARX.

3.1 Extended Differential-linear Correlation of $\boxplus_k$

In the following, we study how to compute the correlation of extended differential-linear approximation of n-bit modulo addition with k inputs, i.e., $\boxplus_k$. This paper reduces the computational complexity of ordinary differential-linear correlation of $\boxplus_2$ to one-eighth of [NSLL22]. Note that Definition 6 used two additions. Moreover, we investigate the case where the initial carries of two additions analyzed are $a, b \in \{0, 1, ..., k-1\}$, respectively. We replace the two functions S used in formula 2 by $\boxplus_{k,*\blacktriangleleft a}$ and $\boxplus_{k,*\blacktriangleleft b}$. For clarity, we present the definition of EDL correlation for modulo additions.

Definition 6. *Let $k > 1$ be an integer and $(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ be an extended DL approximation of $\boxplus_k^n$ with initial carries $a, b \in \{0, \dots, k-1\}$. Here $\alpha^0, ..., \alpha^{k-1} \in \mathbb{F}_2^n$ and $\lambda^0, \lambda^1 \in \mathbb{F}_2^n$ are the input differences and output masks, respectively. Then the extended differential-linear correlation is defined as*

$$Cor_{\substack{*\blacktriangleleft a\\ *\blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = \frac{1}{2^{kn}} \sum_{x^i \in \mathbb{F}_2^n, 0 \le i < k} (-1)^{\substack{\lambda^0 \cdot \boxplus_{k,*\blacktriangleleft a}(x^0, x^1, ..., x^{k-1}) \oplus \\ \lambda^1 \cdot \boxplus_{k,*\blacktriangleleft b}(x^0 \oplus \alpha^0, x^1 \oplus \alpha^1, ..., x^{k-1} \oplus \alpha^{k-1})}}.$$

Theorem 1. *Let $k > 1$ be a fixed integer and $(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ be an extended DL approximation of $\boxplus_k^n$ with initial carries $a, b \in \{0, \dots, k-1\}$. Denote the carry and residue functions by*

$$F : \mathbb{F}_2^k \times \{0, ..., k-1\} \to \{0, ..., k-1\}, F(x, y) = Floor((HW(x) + y)/2),$$
$$R : \mathbb{F}_2^k \times \{0, ..., k-1\} \to \{0, 1\}, R(x, y) = (HW(x) + y) \bmod 2,$$

respectively. Let $l = k^2$, L be the row vector of dimension l with all entries equal to 1, and C be the column vector of dimension l with a single 1 at the $(a \times k + b)$-th row and zero otherwise. Let $A_0, ..., A_{2^{k+2}-1}$ be the $l \times l$ matrices and are defined as

$$\begin{aligned}(A_r)_{c\times k+d, e\times k+f} = \frac{1}{2^k}[&\#\{x \in \mathbb{F}_2^k : \Lambda^0 \cdot R(x, e) \oplus \Lambda^1 \cdot R(x \oplus \delta, f) = 0, F(x, e) = c\} \\ &- \#\{x \in \mathbb{F}_2^k : \Lambda^0 \cdot R(x, e) \oplus \Lambda^1 \cdot R(x \oplus \delta, f) = 1, F(x \oplus \delta, f) = d\}]\end{aligned}$$

for $\Lambda^0, \Lambda^1, \delta^0, \dots, \delta^{k-1} \in \mathbb{F}_2$ and each $c, d, e, f \in \{0, 1, ..., k-1\}$, where $r = \Lambda^0 2^{k+1} + \Lambda^1 2^k + \sum_{j=0}^{k-1} \delta^j 2^j$ and $\delta = \delta^0 \| \delta^1 \dots \| \delta^{k-1}$. Let $z = z_{n-1} \| ... \| z_1 \| z_0$ be defined as $z_i = \lambda_i^0 2^{k+1} + \lambda_i^1 2^k + \sum_{j=0}^{k-1} \alpha_i^j 2^j$. Then the correlation can be computed as

$$Cor_{\substack{*\blacktriangleleft a\\ *\blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = LA_{z_{n-1}} \dots A_{z_1} A_{z_0} C.$$

Proof. Let the inputs of two n-bit modulo additions be $(x^0, \dots, x^{k-1})$ and $(y^0, \dots, y^{k-1})$, respectively. For $0 \le i < n$, there hold that $x^i, y^i \in \mathbb{F}_2^n$ and $y^i = x^i \oplus \alpha^i$. We set the least significant carry (initial carry) $c_0^0 = a$ and $c_0^1 = b$. Then the carries c_i^0, c_i^1 and the sums s_i^0, s_i^1 with $i = 0, ..., n-1$ are defined as follows

$$s_i^0 = R(x_i^0 \| x_i^1 ... x_i^{k-1}, c_i^0), c_{i+1}^0 = F(x_i^0 \| x_i^1 ... x_i^{k-1}, c_i^0),$$
$$s_i^1 = R(y_i^0 \| y_i^1 ... y_i^{k-1}, c_i^1), c_{i+1}^1 = F(y_i^0 \| y_i^1 ... y_i^{k-1}, c_i^1).$$

For $j = 1, ..., n$, let

$$b_j = \bigoplus_{i=0}^{j-1} (\lambda_i^0 \cdot s_i^0 \oplus \lambda_i^1 \cdot s_i^1).$$

Let $P(z, j)$ be the column vector

$$P(z, j)_{c\times k+d} = \Pr[b_j = 0, c_j^0 = c, c_j^1 = d] - \Pr[b_j = 1, c_j^0 = c, c_j^1 = d]$$

for $j = 1, ..., n$ and $0 \le c, d < k$. For $P(z, 1)$, by the definition of A_r, we have

$$P(z, 1) = A_{z0} C.$$

Let $M(z,j)$ be the $l \times l$ matrix defined as

$$\begin{aligned} M(z,j)_{c\times k+d, e\times k+f} = &\Pr[\lambda_j^0 \cdot s_j^0 \oplus \lambda_j^1 \cdot s_j^1 = 0, c_{j+1}^0 = c, c_{j+1}^1 = d | c_j^0 = e, c_j^1 = f] \\ &- \Pr[\lambda_j^0 \cdot s_j^0 \oplus \lambda_j^1 \cdot s_j^1 = 1, c_{j+1}^0 = c, c_{j+1}^1 = d | c_j^0 = e, c_j^1 = f], \end{aligned}$$

for $j \in \{0, ..., n-1\}$.

We denote (c_j^0, c_j^1) and (c_{j-1}^0, c_{j-1}^1) by G and H, respectively. Then $P(z,j)_{c\times k+d}$ for $j > 1$ can be computed as

$$\begin{aligned} &p(z,j)_{c\times k+d} = \Pr[b_j = 0, G = (c,d)] - \Pr[b_j = 1, G = (c,d)] \\ &= \sum_{h=0}^{1} (-1)^h \Pr[b_j = h, G = (c,d)] \\ &= \sum_{h=0}^{1} (-1)^h \sum_{0 \leqslant e,f < k} \sum_{v=0}^{1} \Pr[b_j = h, G = (c,d) | b_{j-1} = v, H = (e,f)] \Pr[b_{j-1} = v, H = (e,f)] \\ &= \sum_{h=0}^{1} (-1)^h \sum_{0 \leqslant e,f < k} \sum_{v=0}^{1} \Pr[\lambda_{j-1}^0 s_{j-1}^0 \oplus \lambda_{j-1}^1 s_{j-1}^1 = h \oplus v, G = (c,d) | b_{j-1} = v, H = (e,f)] \Pr[b_{j-1} = v, H = (e,f)] \\ &= \sum_{h=0}^{1} (-1)^h \sum_{0 \leqslant e,f < k} \sum_{v=0}^{1} \Pr[\lambda_{j-1}^0 s_{j-1}^0 \oplus \lambda_{j-1}^1 s_{j-1}^1 = h \oplus v, G = (c,d) | H = (e,f)] \Pr[b_{j-1} = v, H = (e,f)] \\ &= \sum_{0 \leqslant e,f < k} \sum_{v=0}^{1} \Pr[b_{j-1} = v, H = (e,f)] \sum_{h=0}^{1} (-1)^h \Pr[\lambda_{j-1}^0 s_{j-1}^0 \oplus \lambda_{j-1}^1 s_{j-1}^1 = h \oplus v, G = (c,d) | H = (e,f)] \\ &\overset{t=h\oplus v}{=} \sum_{0 \leqslant e,f < k} \left[\sum_{t=0}^{1} (-1)^t \Pr[\lambda_{j-1}^0 s_{j-1}^0 \oplus \lambda_{j-1}^1 s_{j-1}^1 = t, G = (c,d) | H = (e,f)] \right] \times \left[\sum_{v=0}^{1} (-1)^v \Pr[b_{j-1} = v, H = (e,f)] \right] \\ &= \sum_{0 \leqslant e,f < k} M(z, j-1)_{c\times k+d, e\times k+f} \times P(z, j-1)_{e\times k+f} \end{aligned}$$

Then we have

$$P(z,j) = M(z,j-1)P(z,j-1).$$

On the other hand, we have

$$\begin{aligned} LP(z,n) = &\sum_{0 \le c,d < k} (\Pr[b_n = 0, c_n^0 = c, c_n^1 = d] - \Pr[b_n = 1, c_n^0 = c, c_n^1 = d]) \\ &= \Pr[b_n = 0] - \Pr[b_n = 1] = Cor_{\substack{* \blacktriangleleft a \\ * \blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1). \end{aligned}$$

Since $A_{z_i} = M(z,i)$, it follows that

$$Cor_{\substack{* \blacktriangleleft a \\ * \blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = LA_{z_{n-1}} \dots A_{z_1} A_{z_0} C.$$

□

Note that the functions F and R are the carry and residue functions for the basic school-book method for additions with k binary inputs and one input carry. And e, f and c, d are the input carries and the output carries of $\boxplus_{k, c \blacktriangleleft e}^1$ and $\boxplus_{k, d \blacktriangleleft f}^1$. Under fixed input differences and output masks, the $(A_r)_{c\times k+d, e\times k+f}$ is the differential-linear correlation of $\boxplus_{k, c \blacktriangleleft e}^1$ and $\boxplus_{k, d \blacktriangleleft f}^1$.

This correlation of an extended differential-linear trail of addition modulo 2^n with k inputs can thus be computed by doing matrix multiplication n times and a sum operation. For a fixed k, this is a linear-time algorithm to compute the correlation of extended DL trails for $\boxplus_k^n$.

Next, we investigate the case where $k = 2$ and $\lambda^0 = \lambda^1$ to further reduce the computational complexity.

Lemma 2. *Let the input differences and output mask of two* 1*-bit modulo additions,* $\boxplus_{2,*\blacktriangleleft e}$ *and* $\boxplus_{2,*\blacktriangleleft f}$*, be* $\alpha^0, \alpha^1 \in \mathbb{F}_2$ *and* $\lambda \in \mathbb{F}_2$*, respectively. Here* $e, f \in \mathbb{F}_2$*. Let the inputs of two modulo additions are* (x^0, x^1) *and* (y^0, y^1)*, respectively. Denote the sums, input carries and output carries of two* $\boxplus_2^1$ *by* s^0, s^1*,* c_0^0, c_0^1 *and* c_1^0, c_1^1*, respectively. We set* $c, d, e, f, u, w \in \mathbb{F}_2$*. Let* $M(\alpha^0, \alpha^1, \lambda)$ *be the* 4×4 *matrix defined as*

$$M(\alpha^0, \alpha^1, \lambda)_{c\times 2+d, e\times 2+f} = \sum_{v=0}^{1} (-1)^v \Pr[\lambda \cdot (s^0 \oplus s^1) = v, c_1^0 = c, c_1^1 = d | c_0^0 = e, c_0^1 = f],$$

and $N(\alpha^0, \alpha^1, \lambda)$ *be the* 2×2 *matrix defined as*

$$N(\alpha_j^0, \alpha_j^1, \lambda_j)_{u,w} = \sum_{v=0}^{1} (-1)^v \Pr[\lambda \cdot (s^0 \oplus s^1) = v, c_1^0 \oplus c_1^1 = u | c_0^0 \oplus c_0^1 = w].$$

When $w = e \oplus f$*, for fixed* e, f, u *and* w*, there holds*

$$\sum_{\substack{c,d\in\mathbb{F}_2 \\ c\oplus d=u}} M(\alpha^0, \alpha^1, \lambda)_{c\times 2+d, e\times 2+f} = N(\alpha^0, \alpha^1, \lambda)_{u,w}.$$

Proof. Computing all matrices M and N under all $(\alpha^0, \alpha^1, \lambda) \in \mathbb{F}_2^3$ gives the proof. □

Theorem 2. *Let* $(\alpha^0, \alpha^1, \lambda)$ *be an ordinary DL approximation of* $\boxplus_2^n$ *with initial carries* $a, b \in \mathbb{F}_2$*. Denote the carry and residue functions by*

$$F : \mathbb{F}_2^2 \times \{0, 1\} \to \{0, 1\}, F(x, y) = Floor((HW(x) + y)/2),$$
$$R : \mathbb{F}_2^2 \times \{0, 1\} \to \{0, 1\}, R(x, y) = (HW(x) + y) \bmod 2,$$

respectively. Let L *be* $[1, 1]$*, and* C *be the column vector of dimension* 2 *with a single 1 at the* $(a \oplus b)$*-th row and zero otherwise. Let* $B_0, ..., B_7$ *be the* 2×2 *matrices and are defined as*

$$\begin{aligned}(B_r)_{u,w} = \frac{1}{2^3}[&\#\{(x, e) \in \mathbb{F}_2^2 \times \mathbb{F}_2 : \Lambda \cdot (R(x, e) \oplus R(x \oplus \delta, e \oplus w)) = 0, F(x, e) \oplus F(x \oplus \delta, e \oplus w) = u\} \\ &- \#\{(x, e) \in \mathbb{F}_2^2 \times \mathbb{F}_2 : \Lambda \cdot (R(x, e) \oplus R(x \oplus \delta, e \oplus w)) = 1, F(x, e) \oplus F(x \oplus \delta, e \oplus w) = u\}]\end{aligned}$$

for $\Lambda, \delta^0, \delta^1, u, w \in \mathbb{F}_2$*, where* $r = \Lambda \times 4 + \sum_{j=0}^{1} \delta^j \times 2^j$ *and* $\delta = \delta^0 \| \delta^1$*. Let* $z = z_{n-1} \| ... \| z_1 \| z_0$ *be defined as* $z_i = \lambda_i \times 4 + \sum_{j=0}^{1} \alpha_i^j \times 2^j$ *for* $0 \leq i < n$*. Then the correlation can be computed as*

$$Cor(\alpha^0, \alpha^1, \lambda, a, b) = LB_{z_{n-1}} \dots B_{z_1} B_{z_0} C.$$

Proof. For simplicity, the notations b_j, $M(z, j)$ and $P(z, j)$ in the proof of Theorem 1 are used here.

Let $N(z, j)$ be the 2×2 matrix defined by

$$\begin{aligned}N(z, j)_{u,w} = &\Pr[\lambda_j \cdot (s_j^0 \oplus s_j^1) = 0, c_{j+1}^0 \oplus c_{j+1}^1 = u | c_j^0 \oplus c_j^1 = w] \\ &- \Pr[\lambda_j \cdot (s_j^0 \oplus s_j^1) = 1, c_{j+1}^0 \oplus c_{j+1}^1 = u | c_j^0 \oplus c_j^1 = w],\end{aligned}$$

for $j \in \{0, ..., n - 1\}$, and $Q(z, j)$ be the column vector

$$Q(z, j)_u = \Pr[b_j = 0, c_j^0 \oplus c_j^1 = u] - \Pr[b_j = 1, c_j^0 \oplus c_j^1 = u]$$

for $j = 1, ..., n$ and $u \in \mathbb{F}_2$. For $Q(z, 1)$, according to the definition of B_r and Lemma 2, we have

$$\begin{aligned} Q(z,1) &= \left[\sum_{\substack{c,d \in \mathbb{F}_2 \\ c \oplus d = 0}} M(\alpha^0, \alpha^1, \lambda)_{c\times 2+d, e\times 2+f}, \sum_{\substack{c,d \in \mathbb{F}_2 \\ c \oplus d = 1}} M(\alpha^0, \alpha^1, \lambda)_{c\times 2+d, e\times 2+f} \right]^T \\ &= [N(\alpha_0^0, \alpha_0^1, \lambda_0)_{0, e\oplus f}, N(\alpha_0^0, \alpha_0^1, \lambda_0)_{1, e\oplus f}]^T \\ &= N(z,0)C = B_{z_0}C. \end{aligned}$$

Then $Q(z, j)_u$ for $j > 1$ and $u \in \mathbb{F}_2$ can be computed as

$$\begin{aligned} Q(z,j)_u &= \sum_{\substack{c,d \in \mathbb{F}_2 \\ c \oplus d = u}} P(z,j)_{c\times 2+d} \\ &\overset{Thm.\ 1}{=} \sum_{\substack{c,d,e,f \in \mathbb{F}_2 \\ c \oplus d = u}} M(z,j-1)_{c\times 2+d, e\times 2+f} \times P(z,j-1)_{e\times 2+f} \\ &= \sum_{w \in \mathbb{F}_2} \sum_{\substack{c,d \in \mathbb{F}_2 \\ c \oplus d = u}} \sum_{\substack{e,f \in \mathbb{F}_2 \\ e \oplus f = w}} M(z,j-1)_{c\times 2+d, e\times 2+f} \times P(z,j-1)_{e\times 2+f} \\ &= \sum_{w \in \mathbb{F}_2} \sum_{\substack{e,f \in \mathbb{F}_2 \\ e \oplus f = w}} \left(\sum_{\substack{c,d \in \mathbb{F}_2 \\ c \oplus d = u}} M(z,j-1)_{c\times 2+d, e\times 2+f} \right) \times P(z,j-1)_{e\times 2+f} \\ &\overset{Lemma\ 2}{=} \sum_{w \in \mathbb{F}_2} (N(z,j-1)_{u,w}) \times \left(\sum_{\substack{e,f \in \mathbb{F}_2 \\ e \oplus f = w}} P(z,j-1)_{e\times 2+f} \right) \\ &= \sum_{w \in \mathbb{F}_2} N(z,j-1)_{u,w} \times Q(z,j-1)_w. \end{aligned}$$

Then we have

$$Q(z,j) = N(z,j-1)Q(z,j-1).$$

On the other hand, we have

$$\begin{aligned} LQ(z,n) &= \sum_{u \in \mathbb{F}_2} (\Pr[b_n = 0, c_n^0 \oplus c_n^1 = u] - \Pr[b_n = 1, c_n^0 \oplus c_n^1 = u]) \\ &= \Pr[b_n = 0] - \Pr[b_n = 1] = Cor_{\substack{* \blacktriangleleft a \\ * \blacktriangleleft b}}(\alpha^0, \alpha^1, \lambda). \end{aligned}$$

According the definition of B_{z_i} and $N(z, i)$, we have $B_{z_i} = N(z, i)$ and

$$Cor_{\substack{* \blacktriangleleft a \\ * \blacktriangleleft b}}(\alpha^0, \alpha^1, \lambda) = LB_{z_{n-1}} \dots B_{z_1} B_{z_0} C.$$

□

Compared with [NSLL22] that gives the method to compute ordinary DL correlation for $\boxplus_2^n$, Theorem 1 is a generalization of the method of [NSLL22], which presents the method to compute extended DL correlation for $\boxplus_k^n$ with $k \geq 2$. Theorem 2 show that the correlation computation of ordinary DL trails of n-bit modulo additions with 2 inputs can be completed by a chain of 2×2 matrix multiplications instead of 4×4 matrix multiplications proposed by [NSLL22], which reduces the computational complexity of ordinary DL correlation of $\boxplus_2$ to approximately $\frac{1}{8}$. Using Theorem 2 we get the following matrices for $\boxplus_2$.

$$B_0 = \frac{1}{2}\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \quad B_1 = B_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad B_3 = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}$$
$$B_4 = \frac{1}{2}\begin{bmatrix} 2 & -1 \\ 0 & -1 \end{bmatrix} B_5 = B_6 = \frac{1}{2}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} B_7 = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 1 & -2 \end{bmatrix}$$

Theorem 1 was verified by computing the correlations of all extended DL approximations for 4-bit modulo additions with 2 and 3 inputs, i.e., $\boxplus_2^4$ and $\boxplus_3^4$. An analogous method was used to verify Theorem 2. See the source code provided by this paper for more details. Following corollaries about extended DL correlation of $\boxplus_k$ will reduce the computational complexity in some cases.

Corollary 1. *Let $n \geq 1$ be an integer and $(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ be an extended differential-linear approximation of n-bit modulo additions with initial carries $a, b \in \{0, 1, ..., k-1\}$. For a fixed integer t with $(n - t) \geq 1$, let $(\lfloor\alpha^0\rfloor_{n-t}, ..., \lfloor\alpha^{k-1}\rfloor_{n-t}, \lfloor\lambda^0\rfloor_{n-t}, \lfloor\lambda^1\rfloor_{n-t})$ be a extended differential-linear approximation of $(n-t)$-bit modulo additions with initial carries a, b. If $\lceil\lambda^0\rceil^t = \lceil\lambda^1\rceil^t = 0^t$, it holds*

$$Cor_{\substack{\boxplus_k^n, *\blacktriangleleft a \\ *\blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = Cor_{\substack{\boxplus_k^{(n-t)}, *\blacktriangleleft a \\ *\blacktriangleleft b}}(\lfloor\alpha^0\rfloor_{n-t}, ..., \lfloor\alpha^{k-1}\rfloor_{n-t}, \lfloor\lambda^0\rfloor_{n-t}, \lfloor\lambda^1\rfloor_{n-t}).$$

Proof.

$$\begin{aligned} Cor_{\substack{\boxplus_k^n, *\blacktriangleleft a \\ *\blacktriangleleft b}}(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) &= \frac{1}{2^{kn}} \sum_{x^i \in \mathbb{F}_2^n, 0 \leq i < k} (-1)^{\substack{\lambda^0 \cdot \boxplus_k^n (x^0, x^1, ..., x^{k-1}) \oplus \\ \lambda^1 \cdot \boxplus_k^n (x^0 \oplus \alpha_0, x^1 \oplus \alpha_1, ..., x^{k-1} \oplus \alpha_{k-1})}} \\ &\overset{\lceil\lambda^0\rceil^t = \lceil\lambda^1\rceil^t = 0^t}{=} \frac{1}{2^{k(n-t)}} \sum_{x^i \in \mathbb{F}_2^{n-t}, 0 \leq i < k} (-1)^{\substack{\lfloor\lambda^0\rfloor_{n-t} \cdot \lfloor\boxplus_k^{n-t}(x^0, x^1, ..., x^{k-1})\rfloor_{n-t} \oplus \\ \lfloor\lambda^1\rfloor_{n-t} \cdot \lfloor\boxplus_k^{n-t}(x^0 \oplus \alpha_0, x^1 \oplus \alpha_1, ..., x^{k-1} \oplus \alpha_{k-1})\rfloor_{n-t}}} \\ &= Cor_{\substack{\boxplus_k^{(n-t)}, *\blacktriangleleft a \\ *\blacktriangleleft b}}(\lfloor\alpha^0\rfloor_{n-t}, ..., \lfloor\alpha^{k-1}\rfloor_{n-t}, \lfloor\lambda^0\rfloor_{n-t}, \lfloor\lambda^1\rfloor_{n-t}) \end{aligned}$$

□

Corollary 1 indicates that $\lceil\alpha^i\rceil^t$ for $0 \leq i < k$ will not affect the extended DL correlation if $\lceil\lambda^0 \vee \lambda^1\rceil^t = 0^t$. The following corollary can further simplify the computation of ordinary DL correlation of $\boxplus_k$ for $k \geq 2$. Corollary 2 shows that if the input differences and output mask satisfy a particular form, it is free to know that the ordinary DL correlation of $\boxplus_k^n$ is ± 1.

Corollary 2. *Let $(\alpha^0, \ldots, \alpha^{k-1}, \lambda)$ be an ordinary differential-linear approximation of $\boxplus_k^n$. For a fixed integer $t \in \{0, 1, ..., n-1\}$, if $\lceil\lambda\rceil^t = 0^t$ and $\lfloor\alpha^i\rfloor_{n-t-1} = 0^{n-t-1}$ for $0 \leq i < k-1$, it holds that the ordinary DL correlation $Cor(\alpha^0, \ldots, \alpha^{k-1}, \lambda) = \pm 1$.*

Proof. Denote the two additions by s^0 and s^1, respectively. If $\lfloor\alpha^i\rfloor_{n-t-1} = 0^{n-t-1}$ for $0 \leq i < k-1$, there hold that

$$\begin{aligned} \lfloor s^0\rfloor_{n-t-1} &= \lfloor s^1\rfloor_{n-t-1}, \\ s^0_{n-t-1} \oplus s^1_{n-t-1} &= \sum_{i=0}^{k-1} \alpha^i_{n-t-1}. \end{aligned}$$

Due to Corollary 1, we have

$$
\begin{aligned}
Cor(\alpha^0,\ldots,\alpha^{k-1},\lambda) &= Cor(\lfloor\alpha^0\rfloor_{n-t},\ldots,\lfloor\alpha^{k-1}\rfloor_{n-t},\lfloor\lambda\rfloor_{n-t}) \\
&= \frac{1}{2^{k(n-t)}} \sum_{x^i\in\mathbb{F}_2^{(n-t)},0\leq i<k} (-1)^{\lfloor\lambda\rfloor_{n-t}\cdot(\lfloor s^0\rfloor_{n-t}\oplus\lfloor s^1\rfloor_{n-t})} \\
&= \frac{1}{2^{k(n-t)}} \sum_{x^i\in\mathbb{F}_2^{(n-t)},0\leq i<k} (-1)^{\lambda_{n-t-1}\cdot(\sum\limits_{i=0}^{k-1}\alpha^i_{n-t-1})} \\
&= \pm 1
\end{aligned}
$$

□

3.2 Computing the Differential-linear Correlation of the Round Function of ARX ciphers

In this section, we use the results in previous sections and Morawiecki's technique [MPS13] to compute the DL correlations of round functions of ARX ciphers. For simplicity, we call Morawiecki's technique as the correlation propagation technique of difference bits.

To apply Morawiecki's technique for evaluating the DL correlations, we studied how to compute the extended DL correlation of the building block $S(x^0,x^1,...,x^{k-1}) = (\boxplus_k(x^0,x^1,...,x^{k-1}),x^1,x^2,...,x^{k-1})$ for $k\geq 2$ with the knowledge of $\Pr[x^j_i\oplus y^j_i=1]$ for all $0\leq i<n$ and $0\leq j<k$. We remark that a similar method can be used to compute the extended rotational DL correlation of the building block $S(x^0,x^1,...,x^{k-1})$. Let $(\alpha^0,...,\alpha^{k-1})$ be the input differences of an DL approximation of $S(x^0,x^1,...,x^{k-1})$. In Morawiecki's technique, the prerequisite is that the events $\alpha^i_j=1$ are mutually independent for all $0\leq j<n$ and $0\leq i<k$ [LSL21, NSLL22].

Theorem 3. *Let $k>1$ be a fixed integer and $(\alpha^0,...,\alpha^{k-1},\lambda^0,\lambda^1)$ be an extended DL approximation of $\boxplus^n_k$. Let $\Pr[\alpha^i_j=1]=p^i_j$ for all $i\in\{0,...,k-1\}$ and $j\in\{0,...,n-1\}$. The events $\alpha^i_j=1$ are mutually independent for all $0\leq j<n$ and $0\leq i<k$. Let $l=k^2$, L be the row vector of dimension l with all entries equal to 1, and C be the column vector of dimension l with a single 1 at the 0-th row and zero otherwise. Then the correlation, denoted by $Cor(\alpha^0,...,\alpha^{k-1},\lambda^0,\lambda^1)$, can be computed as*

$$Cor(\alpha^0,...,\alpha^{k-1},\lambda^0,\lambda^1) = LH_{n-1}\ldots H_1H_0C,$$

where H_i is a $l\times l$ matrix and is defined as

$$H_i = \sum_{\substack{\beta\in\{0,1,\ldots,2^k-1\} \\ \beta=\beta_{k-1}\|\cdots\beta_1\|\beta_0}} \prod_{j\in\{0,1,..,k-1\}} \left[(1-\beta_j)-(-1)^{\beta_j}p^j_i\right] A_{\lambda^0_i 2^{k+1}+\lambda^1_i 2^k+\beta}$$

Proof. Note that $z=z_{n-1}\|...\|z_1\|z_0$ be the word associated with one trail, where $z_i=\lambda^0_i2^{k+1}+\lambda^1_i2^k+\sum\limits_{j=0}^{k-1}\alpha^j_i2^j$. Due to that the events $\alpha^i_j=1$ are mutually independent for all

$0 \leq j < n$ and $0 \leq i < k$, we have

$$
\begin{aligned}
& Cor(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) \\
& = \sum_{(\alpha^0,...,\alpha^{k-1}) \in \mathbb{F}_2^{kn}} \Pr[InputDiff. = (\alpha^0, ..., \alpha^{k-1})] Cor(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) \\
& = \sum_{(\alpha^0,...,\alpha^{k-1}) \in \mathbb{F}_2^{kn}} \prod_{i \in \{0,...,n-1\}} \prod_{j \in \{0,1,..,k-1\}} \left[(1-\alpha_i^j) - (-1)^{\alpha_i^j} p_i^j\right] \times Cor(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) \\
& = \sum_{(\alpha^0,...,\alpha^{k-1}) \in \mathbb{F}_2^{kn}} L \prod_{i \in \{0,...,n-1\}} \prod_{j \in \{0,1,..,k-1\}} \left[(1-\alpha_i^j) - (-1)^{\alpha_i^j} p_i^j\right] A_{z_i} C \\
& = L \prod_{i = \in \{0,...,n-1\}} \left[\sum_{\beta \in \{0,1,...,2^k-1\}} \prod_{j \in \{0,1,..,k-1\}} \left[(1-\beta_i^j) - (-1)^{\beta_i^j} p_i^j\right] A_{\lambda_i^0 2^{k+1} + \lambda_i^1 2^k + \beta} \right] C.
\end{aligned}
$$

□

Corollary 3. *Let $(\alpha^0, \alpha^1, \lambda^0, \lambda^1)$ be an extended differential-linear approximation of $\boxplus_2^n$ with initial carries $a, b \in \mathbb{F}_2$. Let $\Pr[\alpha_j^i = 1] = p_j^i$ for all $i \in \{0,1\}$ and $j \in \{0, ..., n-1\}$. The events $\alpha_j^i = 1$ are mutually independent for all $0 \leq j < n$ and $0 \leq i < 2$. Let $h = HIndex(\lambda^0 \vee \lambda^1)$. Then the correlation, denoted by $Cor(\alpha^0, \alpha^1, \lambda^0, \lambda^1)$, will be 0 if $p_h^0 = 0.5$ or $p_h^1 = 0.5$.*

Proof. According to Corollary 1, the most significant $(n-h-1)$ bits of α^0, α^1, λ^0 and λ^1 do not affect the $Cor(\alpha^0, \alpha^1, \lambda^0, \lambda^1)$.

For $p_h^0 = 0.5$, according to Theorem 3, it holds that

$$
\begin{aligned}
H_h = & \sum_{\beta \in \{0,1,2,3\}} \prod_{j \in \{0,1\}} \left[(1-\beta_j) - (-1)^{\beta_j} p_h^j\right] A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + \beta} \\
& = 0.5 \times p_h^1 \times (A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 0} + A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 1}) + \\
& \quad 0.5 \times (1 - p_h^1) \times (A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 2} + A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 3}).
\end{aligned}
$$

Let L (resp. $\bar{L}$) be the row vector of dimension 4 with all entries equal to 1 (resp. 0). When $(\lambda^0 \vee \lambda^1)_h = 1$, according to the matrices A_{z_i} with $z_i \geq 4$ in Appendix A, it holds that

$$
\begin{aligned}
LH_h & = 0.5 \times p_h^1 \times L(A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 0} + A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 1}) + \\
& \quad 0.5 \times (1 - p_h^1) \times L(A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 2} + A_{\lambda_h^0 2^3 + \lambda_h^1 2^2 + 3}) \\
& = 0.5 \times p_h^1 \times \bar{L} + 0.5 \times (1 - p_h^1) \times \bar{L} \\
& = \bar{L}
\end{aligned}
.
$$

Thus, it holds that

$$
Cor(\alpha^0, \alpha^1, \lambda^0, \lambda^1) = LH_{n-1} \ldots H_1 H_0 C = LH_h \ldots H_1 H_0 C = \bar{L} H_{h-1} \ldots H_1 H_0 C = 0.
$$

Similarly, one can derive the conclusion for $p_h^1 = 0.5$, and we omit the details. □

Theorem 3 only considers the DL cryptanalysis of modulo additions. Next, we will study the computation of extended DL correlation of a general function $S(x^0, x^1, ..., x^{k-1}) = (\boxplus_k(x^0, x^1, ..., x^{k-1}), x^1, x^2, ..., x^{k-1})$, which is a basic building block of ARX ciphers.

Lemma 3. *Let $(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ be one extended differential-linear approximation of $S(x^0, x^1, ..., x^{k-1}) = (\boxplus_k(x^0, x^1, ..., x^{k-1}), x^1, x^2, ..., x^{k-1})$ with $x^i, \alpha^j \in \mathbb{F}_2^n$ for*

$0 \leq i, j < k$ and $\lambda^0, \lambda^1 \in \mathbb{F}_2^{kn}$. Here $\lfloor \lambda^0 \rfloor_{kn-n} = \lfloor \lambda^1 \rfloor_{kn-n}$. Then the correlation $Cor_S(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ can be computed as

$$Cor_S(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = L \left[\prod_{i=0}^{n-1} (-1)^{(\lambda^1_{i+(k-2)n}, \lambda^1_{i+(k-3)n}, ..., \lambda^1_i) \cdot (\alpha^1_i, ..., \alpha^{k-1}_i)} A_{z_i} \right] C,$$

where

$$z_i = \lambda^0_{i+(k-1)n} 2^{k+1} + \lambda^1_{i+(k-1)n} 2^k + \sum_{j=0}^{k-1} \alpha^j_i 2^j.$$

Proof. Let the inputs of those two S be $(x^0, \ldots, x^{k-1})$ and $(y^0, \ldots, y^{k-1})$, respectively. Let $y^i = x^i \oplus \alpha^i$ for $0 \leq i < k$. Denote $\boxplus_k(x^0, ..., x^{k-1})$ and $\boxplus_k(y^0, ..., y^{k-1})$ by s^0 and s^1, respectively. Then we have

$$\begin{aligned}
& Cor_S(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lambda^0, \lambda^1) \\
= & \frac{1}{2^{kn}} \sum_{x^i \in \mathbb{F}_2^n, 0 \leqslant i < k} (-1)^{\lambda^0 \cdot S(x^0, ..., x^{k-1}) \oplus \lambda^1 \cdot S(y^0, ..., y^{k-1})} \\
= & \frac{1}{2^{kn}} \sum_{x^i \in \mathbb{F}_2^n, 0 \leq i < k} (-1)^{\lceil \lambda^0 \rceil^n \cdot s^0 \oplus \lceil \lambda^1 \rceil^n \cdot s^1 \oplus \lfloor \lambda^1 \rfloor_{kn-n} \cdot (\alpha^1, ..., \alpha^{k-1})} \\
= & (-1)^{\lfloor \lambda^1 \rfloor_{kn-n} \cdot (\alpha^1, ..., \alpha^{k-1})} \times Cor_{\boxplus_k^n}(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lceil \lambda^0 \rceil^n, \lceil \lambda^1 \rceil^n).
\end{aligned}$$

Here $Cor_{\boxplus_k^n}(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lceil \lambda^0 \rceil^n, \lceil \lambda^1 \rceil^n)$ is the correlation of DL approximation $(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lceil \lambda^0 \rceil^n, \lceil \lambda^1 \rceil^n)$ of $\boxplus_k^n$. Applying Theorem 1 to $Cor_{\boxplus_k^n}(\alpha^0, \alpha^1, ..., \alpha^{k-1}, \lceil \lambda^0 \rceil^n, \lceil \lambda^1 \rceil^n)$ gives the proof. □

Lemma 3 studied the extended differential-linear cryptanalysis of S under fixed input differences $(\alpha^0, ..., \alpha^{k-1})$. Next, Lemma 3 and Corollary 1 lead to the following generalization of Theorem 3, where the events $\alpha^i_j = 1$ are mutually independent for all $0 \leq j < n$ and $0 \leq i < k$.

Corollary 4. *Let $(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ be one extended differential-linear approximation of $S(x^0, ..., x^{k-1}) = (\boxplus_k(x^0, ..., x^{k-1}), x^1, x^2, ..., x^{k-1})$ with $x^i, \alpha^j \in \mathbb{F}_2^n$ for $0 \leq i, j < k$ and $\lambda^0, \lambda^1 \in \mathbb{F}_2^{kn}$. Here $\lfloor \lambda^0 \rfloor_{kn-n} = \lfloor \lambda^1 \rfloor_{kn-n}$. Let $\Pr[\alpha^i_j = 1] = p^i_j$ for all $i \in \{0, ..., k-1\}$ and $j \in \{0, ..., n-1\}$. The events $\alpha^i_j = 1$ are mutually independent for all $0 \leq j < n$ and $0 \leq i < k$. For a fixed integer t with $(n-t) \geq 1$, let $\lceil \lambda^0 \rceil^t = \lceil \lambda^1 \rceil^t = 0^t$. Then the correlation $Cor_S(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1)$ can be computed as*

$$Cor_S(\alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = LH_{n-t-1} \ldots H_1 H_0 C \times (-1)^{\overset{k-1}{\underset{j=1}{\oplus}} (\lambda^1_{n \times (k-j)-1}, ..., \lambda^1_{n \times (k-j)-t}) \cdot \lceil \alpha^j \rceil^t},$$

where H_i is a $k^2 \times k^2$ matrix for $0 \leq i < n-t$ and is defined as

$$H_i = \sum_{\beta \in \{0,1,...,2^k-1\}} \left[\begin{array}{c} (-1)^{(\lambda^1_{i+(k-2)n}, \lambda^1_{i+(k-3)n}, ..., \lambda^1_i) \cdot \lfloor \beta \rfloor_{k-1}} \times \prod_{j \in \{0,1,..,k-1\}} \left[(1-\beta_j) - (-1)^{\beta_j} p^j_i \right] \\ \times A_{\lambda^0_{i+(k-1)n} 2^{k+1} + \lambda^1_{i+(k-1)n} 2^k + \beta} \end{array} \right]$$

Proof. Applying Theorem 3, Lemma 3 and Corollary 1 give the proof. □

4 Automatic Search for Differential-linear Trails of ARX Ciphers

This section intends to search ordinary DL trails of ARX ciphers with existing solvers. The solver used in this paper is Gurobi. First, we study how to compute the correlation of DL trails of $\boxplus_k$ with current existing solver, which is the foundation of our automatic search of DL trails in ARX ciphers. Second, in ARX ciphers, we use a single DL trail under some explicit conditions to give a good estimate of the correlation. Then, we further simplify the correlation calculation of DL trails in ARX ciphers to ensure that the search of differential-linear trails can be solved by the solver.

4.1 The Transformation from Matrix Multiplication Chain to MIQCP

This section study that how to compute an arbitrary matrix multiplication chain with current existing solver. Taking differential cryptanalysis and linear cryptanalysis as examples, the differential probability and the linear approximation correlation of $\boxplus_k$ for $k \geq 2$ can be computed by using a chain of matrix multiplications[LM01, Wal03, NW06]. However, at FSE 2016, Fu et al. [FWG+16] used a Boolean function and a compact finite automaton to compute the differential probability and the linear approximation correlation of $\boxplus_2$, respectively. To our knowledge, there is no good method to solve the problem that using existing solvers to compute an arbitrary matrix multiplication chain, and it is still an open problem to directly model differential propagation and linear approximation of $\boxplus_k$ for $k > 2$ with current existing solvers. For example, in the automatic search of linear approximations of $\boxplus_3$, the usual method is to split $\boxplus_3$ into two $\boxplus_2$, and these two $\boxplus_2$ are modeled separately with existing solvers. Thus, the final estimation is approximate but not accurate. In this section, we focus on transforming an arbitrary matrix multiplication chain into a Mixed Integer Quadratically-Constrained Programs (MIQCP), which can be handled by current existing solvers.

Next, we take the correlation computation of an ordinary DL approximation of $\boxplus_2$ (Theorem 2) as an example to introduce our method. The notations of the proof of Theorem 2 is used here. There are three steps to transform an arbitrary matrix multiplication chain into a MIQCP.

Step 1: Convert all entries of matrices to integers.

First, it is trivial to have

$$Cor_{\substack{*\blacktriangleleft a\\ *\blacktriangleleft b}}(\alpha^0, \alpha^1, \lambda) = \frac{1}{2^n} L B'_{z_{n-1}} \dots B'_{z_1} B'_{z_0} C,$$

where $B'_{z_i} = 2 \times B_{z_i}$ for all $0 \leq i < n$. Let $Q'(z, i) = 2^j \times Q(z, i)$. $Q'(z, i)$ and B'_{z_i} are used as intermediate dummy variables in the MIQCP model.

Step 2: Model the relationship between all entries of matrices, input differences and output masks.

All entries of B'_{z_i} are determined by z_i, where $z_i = \lambda_i \times 4 + \sum_{j=0}^{1} \alpha_i^j \times 2^j$. Let $B'_{z_i}[x, y]$ be the entry of matrix B'_{z_i} in row x and column y. It is necessary to describe all possible $(B'_{z_i}[x, y], \lambda_i, \alpha_i^0, \alpha_i^1)$ patterns for all $(x, y) \in \mathbb{F}_2^2$ and $0 \leq z_i < 8$. A convex hull of a finite set $P[z, i, x, y] = \{(B'_{z_i}[x, y], \lambda_i, \alpha_i^0, \alpha_i^1) : \lambda_i, \alpha_i^0, \alpha_i^1 \in \mathbb{F}_2\}$ is the smallest convex set that contains $P[z, i, x, y]$, denoted by $Conv(P[z, i, x, y])$. There are two methods to model the relationship according to whether $Conv(P[z, i, x, y]) = P[z, i, x, y]$ or not.

Case 1. $P[z, i, x, y] = Conv(P[z, i, x, y])$ As described in [SHW+14], using the inequality_generator() function in the sage.geometry polyhedron class of the SAGE Computer Algebra System can generate the linear inequalities to describe all points in $Conv(P[z, i, x, y])$. For binary vectors, [SHW+14] presented a method to exclude one arbitrary point without influence other points. Lemma 4 models an arbitrary subset of $\mathbb{F}_2^n$ with finite inequalities.

Lemma 4 ([SHW+14])**.** *For any subset $X \subseteq \mathbb{F}_2^n$, denote one point of X by $(x_0, \dots, x_{n-2}, x_{n-1}) \in X$. For a given point $\delta = (\delta_0, \delta_1, ..., \delta_{n-1}) \in \mathbb{F}_2^n$ and $\delta \notin X$, then all points of X but δ satisfy $\sum_{i=0}^{n-1}[\delta_i + (-1)^{\delta_i} x_i] \geq 1$.*

Theorem 4 ([BJ72, SHW+14]). *Assume that $x \in \mathbb{F}_2^n$ and $Conv(X)$ is the convex hull of a set $X \subseteq \mathbb{F}_2^n$. Then $X = Conv(X)$.*

Theorem 4 showed that for $X \subseteq \mathbb{F}_2^n$, it holds that $Conv(X) = X$. However, since the values of $B'_{z_i}[x, y]$ are integers, not binaries, convex hull $Conv(P[z, i, x, y])$ may contain redundant points apart from $P[z, i, x, y]$.

Case 2. $P[z, i, x, y] \subsetneqq Conv(P[z, i, x, y])$ If $P[z, i, x, y] \subsetneqq Conv(P[z, i, x, y])$, we will use the following method to describe all possible $(B'_{z_i}[x, y], \lambda_i, \alpha_i^0, \alpha_i^1)$ patterns accurately.

Lemma 5. *Let P be a subset of integer set $\mathbb{Z}$ and be denoted as $P = \{p_0, ..., p_{t-1} : p_i \in \mathbb{Z}\}$. For any subset $X \subseteq \{P \times \mathbb{F}_2^{n-1}\}$, denote a point of X by $(y, x_t, x_{t+1}, \ldots, x_{n+t-2})$. For a given point $\delta = (\delta_0, \delta_1, ..., \delta_{n-1}) \notin X$, where $\delta_0 \in P$ and $\delta_i \in \mathbb{F}_2$ for $0 < i < n$, let $\Delta \in \mathbb{F}_2^t$ be the one-hot code decoding (t-bit vector) of δ_0, i.e., $\delta_0 = \sum_{i=0}^{t-1}[\Delta_i \times p_i]$ and $\sum_{i=0}^{t-1} \Delta_i = 1$. Let $x_0, x_1, \ldots, x_{t-1}$ be the intermediate dummy bit variables. Then all points of X but δ satisfy*

$$\begin{cases} \sum_{i=0}^{t-1}[\Delta_i + (-1)^{\Delta_i} x_i] + \sum_{i=1}^{n-1}[\delta_i + (-1)^{\delta_i} x_{i+t-1}] \geq 1 \\ \sum_{i=0}^{t-1} x_i = 1, \sum_{i=0}^{t-1}[x_i \times p_i] = y \end{cases}.$$

Proof. Here $(x_0, x_1, ..., x_{t-1})$ is the one-hot code format of y, and each y has a unique bit vector $(x_0, x_1, ..., x_{t-1})$. Only excluding the point δ from $P \times \mathbb{F}_2^{n-1}$ is converted to excluding the point $(\Delta_0, \Delta_1, \ldots, \Delta_{t-1}, \delta_1, ..., \delta_{n-1})$ from $\mathbb{F}_2^{n+t-1}$. Then, using Lemma 4 gives the proof.

□

Using above method, the relationship between all entries of matrices, input differences and output masks can be accurately described without any redundant points.

Step 3: Compute $Q'(z, i)$ for $1 \leq i \leq n$ step by step.

Since the highest algebraic degree of the equation supported by Gurobi is 2, we use the method of introducing intermediate variables $Q'(z, i)$ and B'_{z_i} to complete the matrix multiplication chain step by step. We have $Cor_{\substack{*\blacktriangleleft a \\ *\blacktriangleleft b}}(\alpha^0, \alpha^1, \lambda) = \frac{1}{2^n} LQ'(z, n)$, and $Q'(z, i)$ can be computed by adding the following equation to the MIQCP model.

$$Q'(z, i+1)[x] = \sum_{y \in \mathbb{F}_2} B'_{z_i}[x, y] Q'(z, i)[y] \quad for\ all\ x \in \mathbb{F}_2,\ 0 \leq i < n$$

Ordinary Differential-linear Model of $\boxplus_2^n$. For the n-bit modular addition operation $\boxplus_2^n$, we use α and β to stand for the input differences and denote the output mask as λ. Let $B'_0, B'_1 \ldots B'_{n-1}$ be 2×2 integer matrices and $Q'_0, Q'_1 \ldots Q'_n$ be 2-dimensional integer column vectors. We denote the integer of matrix B'_i in row x and column y (resp. the integer of Q'_i in row x) by $B'_i[x, y]$ (resp. $Q'_i[x]$). Then the correlation of DL approximation, i.e., $Cor(\alpha, \beta, \lambda)$, can be computed as

$$Cor(\alpha, \beta, \lambda) = (Q'_n[0] + Q'_n[1])/2^n$$

if and only the values of α, β, λ validate all the assertions listed below.

$$Q'_0[0] = 1, Q'_0[1] = 0$$

$$\left.\begin{array}{l}
-1 \le B'_i[0,0] \le 2, -1 \le B'_i[0,1] \le 1 \\
-1 \le B'_i[1,0] \le 1, -2 \le B'_i[1,1] \le 2 \\
-2^{i+1} \le Q'_{i+1}[0] \le 2^{i+1}, -2^{i+1} \le Q'_{i+1}[1] \le 2^{i+1} \\
\\
-B'_i[0,0] - \beta_i \ge -2 \\
B'_i[0,0] - 2\alpha_i - 2\beta_i \ge -3 \\
B'_i[0,0] + \alpha_i + \beta_i + 2\lambda_i \ge 2 \\
B'_i[0,0] + 3\alpha_i + 3\beta_i \ge 2 \\
-B'_i[0,0] + 2\alpha_i - 3\beta_i - 2\lambda_i \ge -4 \\
-B'_i[0,0] - 3\alpha_i + 2\beta_i - 2\lambda_i \ge -4 \\
\\
-B'_i[0,1] - \alpha_i - \beta_i \ge -2 \\
B'_i[0,1] - 2\alpha_i + \beta_i \ge -1 \\
B'_i[0,1] + \alpha_i - 2\beta_i \ge -1 \\
B'_i[0,1] + \alpha_i + 2\lambda_i \ge 1 \\
-B'_i[0,1] + 2\alpha_i + 2\beta_i - 2\lambda_i \ge -1 \\
\\
B'_i[1,0] + \alpha_i + \beta_i \ge 0 \\
2B'_i[1,0] - \alpha_i - \beta_i + 3\lambda_i \ge 0 \\
B'_i[1,0] - 2\alpha_i - 2\beta_i \ge -3 \\
-B'_i[1,0] - \alpha_i + 2\beta_i - 2\lambda_i \ge -2 \\
-2B'_i[1,0] + \alpha_i + \beta_i - \lambda_i \ge -1 \\
-B'_i[1,0] + 2\alpha_i - \beta_i - 2\lambda_i \ge -2 \\
\\
-B'_i[1,1] + 2\alpha_i + 2\beta_i - 2\lambda_i \ge -1 \\
B'_i[1,1] + 3\alpha_i - 2\beta_i \ge -1 \\
B'_i[1,1] + \beta_i + 2\lambda_i \ge 1 \\
B'_i[1,1] - 2\alpha_i + 3\beta_i \ge -1 \\
-2B'_i[1,1] + \alpha_i + \beta_i - \lambda_i \ge -2 \\
-B'_i[1,1] - 3\alpha_i - 3\beta_i - 4\lambda_i \ge -8 \\
B'_i[1,1] - \alpha_i - \beta_i + 4\lambda_i \ge 0 \\
\\
Q'_{i+1}[0] = B'_i[0,0] \times Q'_i[0] + B'_i[0,1] \times Q'_i[1] \\
Q'_{i+1}[1] = B'_i[1,0] \times Q'_i[0] + B'_i[1,1] \times Q'_i[1]
\end{array}\right\} 0 \le i < n$$

To be clear, the above is the implementation of ordinary DL model of additions $\boxplus_2^n$ in Gurobi. Moreover, we provide a Gurobi sample at

https://

for computing the correlations of ordinary DL approximations of modulo additions with arbitrary output linear masks.

In addition, there is a problem in implementing above method with Gurobi. Since in Gurobi [Gur22] the range of integer variables is from -2^{31} to 2^{31}, the DL correlation of modulo 2^n addition for $n \ge 32$ is unable to be handled. In this paper, to solve this problem, we used a fixed coefficient to scale the final DL correlation, which will ensure that all variables used do not exceed the range specified by Gurobi.

For instance, if $x_1 = x_2 \times x_3$ and variable x_1 exceeds the range specified by Gurobi, we will use

$$x'_1 = x'_2 \times x'_3$$

in MIQCP model, where $x'_2 = x_2/100$ and $x'_3 = x_3/100$. Here 100 can be changed to other fixed numbers, as long as variable x'_1 does not exceed the range. In conclusion, this is caused by the Gurobi itself rather than our method, and we will leave this issue for future research.

Discussion. Liu et al. [LSL21] first proposed to use quadratic constraint programming(QCP) to compute the DL correlations of additions with output linear masks of Hamming weight one. This paper proposes a method based on MIQCP to automatically compute the DL correlations of additions with arbitrary output masks, which forms the foundation of our automatic search of DL trails in ARX ciphers.

At this point, the computation of ordinary DL correlation of $\boxplus_2$ is transformed to MIQCP, which can be further handled by the existing solver Gurobi. We remark that a similar method can be used to transform an arbitrary matrix multiplication chain into MIQCP. It is noted that this technology has great potential, e.g., if using the above method in the automatic searching for linear trails of ARX ciphers based on $\boxplus_3$, the solver can accurately give the linear approximation correlation of $\boxplus_3$, rather than splitting $\boxplus_3$ into two $\boxplus_2$ and obtaining a less accurate value.

4.2 Automatic Searching for DL Trails of DLCT in ARX ciphers

Recall that the cipher E is divided into three subciphers E_0, E_m, and E_1 such that $E = E_1 \circ E_m \circ E_0$. Our method of automatically searching for DL trails of E_m is shown in Figure 2. In the following, we provide some methods to further reduce the complexity. Besides Corollary 1 and 2, we focus on the following ideas:

First, though the method to compute an arbitrary matrix multiplication chain is presented in Section 4.1, the computational complexity remains high. For computing DL correlation of ARX ciphers, Liu et al. [LSL21] first proposed to apply Morawiecki et al.'s method on rotational cryptanalysis [MPS13] to several rounds of round function. To reduce complexity, we applied Morawiecki et al.'s method to several rounds of round function as well. The core idea of Morawiecki et al.'s method is that given the independency assumptions on the neighbouring bits, the probability that each difference bit of each round is 1 can be determined iteratively. The following conclusions are used in this paper. We refer to [LSL21] for more details.

Proposition 1 (XOR-rule [LSL21])**.** *For a fixed integer $t \geq 0$, let a, b, a' and b' be n–bit vectors with $\Pr[a_{i-t} \neq a'_i] = p_i$ and $\Pr[b_{i-t} \neq b'_i] = q_i$. Given the independency assumptions on the neighbouring bits, we have*

$$\Pr[(a \oplus b)_{i-t} \neq (a' \oplus b')_i] = p_i + q_i - 2p_i q_i.$$

Theorem 5 ($\boxplus$-rule for DL [LSL21])**.** *Let x, y, x' and y' be n-bit string, such that $\Pr[x_i \neq x'_i] = p_i$ and $\Pr[y_i \neq y'_i] = q_i$. Given the independency assumptions on the neighbouring bits, the differential-linear probability for modular addition can be computed as*

$$\Pr[(x \boxplus y)_i \neq (x' \boxplus y')_i] = p_i + q_i - 2p_i q_i - 2p_i s_i - 2q_i s_i + 4p_i q_i s_i,$$

where $s_0 = 0$ and

$$s_{i+1} = p_i q_i s_i - \frac{p_i q_i + p_i s_i + q_i s_i}{2} + \frac{p_i + q_i + s_i}{2}, i \leq n-1.$$

Compared to Theorem 2, Theorem 5 is more inaccurate but less complex. For serval rounds of ARX ciphers, [LSL21, NSLL22] pointed out that the theoretical estimations of this method are close to the experimental results. In fact, the independency assumption is not strictly valid. Liu et al. [LSL21] pointed out that when using Morawiecki et al.'s method, "the probabilities will rapidly collapse to $\frac{1}{2}$ for all one-bit output masks after a few iterative evaluations of the round function", which may give inaccurate analysis results and ignore some valuable DL trails. In order to trade off the complexity against accuracy, this method is applied to at most three rounds of ARX ciphers in this paper.

Second, due to the need to exhaust the input differences, Theorem 3 and Corollary 4 are difficult to implement directly with the existing solver. In this paper, we use a single

DL trail under some explicit conditions to estimate the DL correlation of $\boxplus_2$ when given the independency assumptions on the neighbouring bits.

Assumption 3 (The estimation of DL correlation of $\boxplus_2$)**.** *Let the input differences and output mask of an ordinary DL approximation of $\boxplus_2$ be $\alpha^0, \alpha^1 \in \mathbb{F}_2^n$ and $\lambda \in \mathbb{F}_2^n$, respectively. Let $\Pr[\alpha_j^i = 1] = p_j^i$ for all $i \in \{0,1\}$ and $j \in \{0, ..., n-1\}$. The events $\alpha_j^i = 1$ are mutually independent for all $0 \leq j < n$ and $0 \leq i < 2$. Denote the DL correlation of $\boxplus_2$ for fixed input differences α^0, α^1 and output mask λ by $Cor(\alpha^0, \alpha^1, \lambda)$. Denote the probability of the input differences being α^0 and α^1 by p_{α^0,α^1}. Then the overall DL correlation, denoted by Cor, can be estimated by*

$$\begin{aligned} Cor &= \sum_{\alpha^0,\alpha^1 \in \mathbb{F}_2^n} p_{\alpha^0,\alpha^1} \times Cor(\alpha^0, \alpha^1, \lambda) \\ &\approx \max_{\alpha^0,\alpha^1 \in \mathbb{F}_2^n} (p_{\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}} \times Cor(\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}, \lfloor \lambda \rfloor_{h+1})), \end{aligned}$$

where

$$\begin{aligned} &h = HIndex(\lambda), \\ &|p_h^j - 0.5| \neq 0 \ for \ j \in \{0,1\}, \\ &p_{\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}} = \prod_{i=0}^{h} p_i^0 \times \prod_{i=0}^{h} p_i^1. \end{aligned}$$

The core idea of estimating a DL approximation by one DL trail is that the largest approximately represents the whole. In Morawiecki et al.'s method, $p_{\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}}$ is easy to compute when given the independency assumptions on the neighbouring bits. Let $h = HIndex(\lambda)$. According to Corollary 1, the highest $(n-h-1)$ bits of α^0, α^1 and λ do not affect the DL correlation. According to Corollary 3, $|p_h^j - 0.5| \neq 0$ ensures that $Cor \neq 0$. This method can further reduce the computational complexity of DL correlation when using existing solver.

Assumption 4 (The estimation of DL correlation of round function $S(x^0, x^1) = (\boxplus_2(x^0, x^1), x^1)$)**.** *Let the input differences and output mask of one ordinary DL approximation be $\alpha^0, \alpha^1 \in \mathbb{F}_2^n$ and $\lambda \in \mathbb{F}_2^{2n}$, respectively. Let $\Pr[\alpha_j^i = 1] = p_j^i$ for all $i \in \{0,1\}$ and $j \in \{0, ..., n-1\}$. The events $\alpha_j^i = 1$ are mutually independent for all $0 \leq j < n$ and $0 \leq i < 2$. For fixed input differences α^0, α^1 and output mask λ, denote the DL correlation of S and $\boxplus_2$ by $Cor_S(\alpha^0, \alpha^1, \lambda)$ and $Cor_{\boxplus_2}(\alpha^0, \alpha^1, \lambda)$, respectively. Denote the probability of the input differences being α^0 and α^1 by p_{α^0,α^1}. Then the overall DL correlation, denoted by Cor, can be estimated by*

$$\begin{aligned} Cor &= \sum_{\alpha^0,\alpha^1 \in \mathbb{F}_2^n} p_{\alpha^0,\alpha^1} \times Cor_S(\alpha^0, \alpha^1, \lambda) \\ &\overset{Lemma 1}{\approx} \left[\sum_{\alpha^0,\alpha^1 \in \mathbb{F}_2^n} p_{\alpha^0,\alpha^1} \times Cor_{\boxplus_2}(\alpha^0, \alpha^1, \lambda) \right] \times \prod_{\substack{j=0 \\ \lambda_j \neq 0}}^{n-1} (1 - 2p_j^1) \\ &\approx \max_{\alpha^0,\alpha^1 \in \mathbb{F}_2^n} (p_{\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}} \times Cor_{\boxplus_2}(\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}, \lfloor \lambda \rfloor_{h+1})) \times \prod_{\substack{j=0 \\ \lambda_j \neq 0}}^{n-1} (1 - 2p_j^1), \end{aligned}$$

where

$$
\begin{aligned}
&h = HIndex(\lambda),\\
&|p_h^j - 0.5| \neq 0 \ for \ j \in \{0, 1\},\\
&p_{\lfloor \alpha^0 \rfloor_{h+1}, \lfloor \alpha^1 \rfloor_{h+1}} = \prod_{i=0}^{h} p_i^0 \times \prod_{i=0}^{h} p_i^1.
\end{aligned}
$$

An analogous method can be used to estimate the DL correlation of round function $S(x^0, x^1) = (\boxplus_2(x^0, x^1)), x^1)$, as showed in Proposition 4. Here Piling-up Lemma and Proposition 3 are used to reduce complexity and get rid of the exhaustion of input differences.

Recall that the cipher E is divided into three subciphers E_0, E_m, and E_1 such that $E = E_1 \circ E_m \circ E_0$. We have presented a method to automatically search a good DL trail of E_m, as depicted in Figure 2. Combining our method with the automatic search methods for Differential and linear trails of ARX ciphers [FWG+16], one will naturally get an automated search method for DL trails of ARX ciphers.

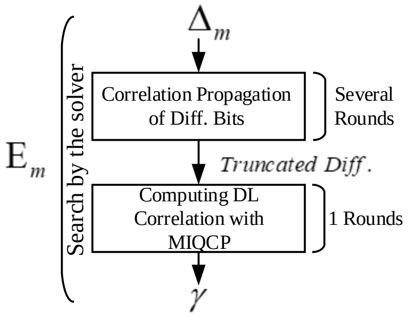


Figure 2: Our method for automatically searching a good DL trail of E_m. We applied Morawiecki et al.'s method [MPS13, LSL21] to several rounds of round function and combined Morawiecki et al.'s method with the method in Section 4.1 to compute the correlation of an ordinary DL trail of E_m.

To search a $(b+1)$-round DL trail of E_m, we applied Morawiecki et al.'s technique and MIQCP to b-round and 1-round ARX ciphers, respectively. By prepending an a-round differential and concatenating a c-round linear trail, the above $(b+1)$-round DL trail of E_m is extended to $(a+b+1+c)$ rounds. For clarity, the configuration of the above $(a+b+1+c)$-round DL trail is denoted by $a+b+1+c$. In this paper, configuration and the number of rounds are abbreviated as **Conf**. and **R**.

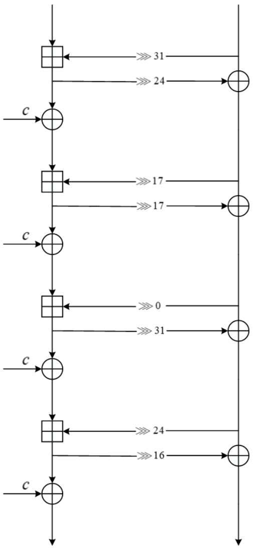
One trick for obtaining good DL trails. As the number of rounds increases, it becomes increasingly difficult to find a good DL trail. The rule of thumb to obtain a good DL trail is to extend the optimal differential trail for a certain number of rounds by a short-round DL trail or extend one good DL trial by a short-round differential/linear trail. For the runtime of the searching algorithm, we spent about several hours on one personal laptop computer (8-core, AMD CPU Ryzen7 4800H, 2.9 GHz). Note that we aim to only find better trails than the previous ones, but we cannot guarantee they are the best trails. We leave it as a future work with acceleration techniques.

5 Applications to ARX Primitives

In this section, we apply the method of automatically searching for DL trails to the ARX primitives Alzette and SPECK. All results are verified experimentally (some in segments).

5.1 Cryptanalysis of 64-bit ARX-box Alzette

Alzette [BBCdS+20] is a 64-bit ARX-based S-box designed by Beierle at al., which is suitable for a larger number of platform architectures. It is the main building block of Sparkle-suite [BBdS+20], one of NIST lightweight crypto standardization finalists. Alzette is parameterized by a constant $c \in \mathbb{F}_2^{32}$ and is defined as a permutation of $\mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$ for each c. The algorithm evaluated in this paper is depicted in Figure 3.



$$x \leftarrow x \boxplus (y \ggg 31), y \leftarrow y \boxplus (x \ggg 24), x \leftarrow x \oplus c$$
$$x \leftarrow x \boxplus (y \ggg 17), y \leftarrow y \boxplus (x \ggg 17), x \leftarrow x \oplus c$$
$$x \leftarrow x \boxplus (y \ggg 0), \ y \leftarrow y \boxplus (x \ggg 31), x \leftarrow x \oplus c$$
$$x \leftarrow x \boxplus (y \ggg 24), y \leftarrow y \boxplus (x \ggg 16), x \leftarrow x \oplus c$$

Figure 3: The Alzette instance

Table 4: The (rotational) differential-linear distinguishers for round-reduced Alzette, where the constants used are 0xB7E15162 and 0x38B4DA56.

R	Conf.	Input Diff.	Output Mask	Correlation		Ref.
				Theory	Exp.[1]	
8	-	(80020100,00010080)	(80000000,00008000)	$-2^{-8.24}$	$-2^{-5.50}$	[NSLL22]
8	2+3+1+2	(80020100,00010080)	(c0010181,01800001)	$2^{-4.14}$	$2^{-4.06}$	**Our**
9	4+3+1+1	(80020100,00010080)	(80600080,60008000)	$-2^{-10.08}$	$-2^{-7.60}$	
10	4+3+1+2	(80020502,00010280)	(03810005,80020180)	$2^{-11.00}$	$2^{-10.48}$	

[1] The number of samples is 2^{30}.

For Alzette, the DL trails searched are shown in Table 4. The experimental correlations given in Table 4 are obtained with 2^{30} random input pairs with the predefined input differences. For 8, 9, 10 rounds of our DL trials in Table 4, the prepositioned differential trails are $(80020100, 00010080) \xrightarrow{2-round,2^{-2}} (00000000, 00010000)$, $(80020100, 00010080) \xrightarrow{4-round,2^{-6}} (01010000, 00030101)$ and $(80020502, 00010280) \xrightarrow{4-round,2^{-8}}$ $(0x00000402, 04020004)$, respectively. The gap between the theoretical estimation and the experimental result for 9-round ARX-box Alzette may come from two aspects: (1) It is due to the accumulated effect of many single DL trails with same input differences and output mask. (2) Morawiecki et al.'s method itself has a marginal error in precision, which is pointed out in [LSL21, NSLL22]. We will leave this issue for future research.

5.2 Cryptanalysis of SPECK

SPECK [BSS+13] is a lightweight iterative cipher designed by the US National Security Agency. In this work, we focus on the versions with 32/48/64-bit block sizes, whose round functions are depicted in Figure 4.

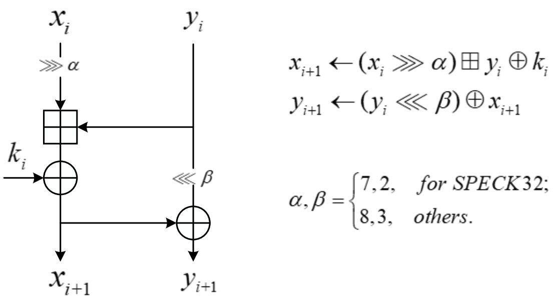

x_i
y_i
$\ggg \alpha$
k_i
$\lll \beta$
x_{i+1}
y_{i+1}
$x_{i+1} \leftarrow (x_i \ggg \alpha) \boxplus y_i \oplus k_i$
$y_{i+1} \leftarrow (y_i \lll \beta) \oplus x_{i+1}$
$\alpha, \beta = \begin{cases} 7, 2, & for\ SPECK32; \\ 8, 3, & others. \end{cases}$

Figure 4: The SPECK instance

Table 5: The differential-linear distinguishers for round-reduced SPECK32/48/64.

	R	Conf.	Input Diff.	Intermediate Diff.[1]	Output Mask	Correlation		Ref.
						Theory	Exp.	
SPECK32	10	-	(0a20,4205)	-	(5820,4020)	$2^{-15.23}$	$2^{-13.90}$	[NSLL22]
	10	5+2+1+2	(0211,0a04)	(8000,840a)	(5820,4020)	$-2^{-13.37}$	$-2^{-11.58}$	**Our**
	10 [4]	-	(0211,0a04)	(0040,0000)	(5820,4020)	$-2^{-8.58}$	-	
	11[2]	6+2+1+2	(8020,4101)	(8000,840a)	(7020,6020)	$-2^{-19.37}$	$-2^{-17.09}$	
	11[2]	6+2+1+2	(0a20,4205)	(8000,840a)	(7020,6020)	$-2^{-18.37}$	$-2^{-16.68}$	
	11[4]	-	(8020,4101)	(0040,0000)	(7020,6020)	$-2^{-11.09}$	-	
SPECK48	11	6+2+1+2	(020082,120200)	(80a000,85a420)	(08a805,098804)	$-2^{-20.46}$	$-2^{-17.55}$	**Our**
SPECK64	11	6+2+1+2	(92400040, 40104200)	(80008004, 84008020)	(4100400c, 4c004000)	$2^{-22.13}$	$2^{-19.47}$	**Our**
	12	6+3+1+2	(12029282, 02020282)	(01202000, 08202000)	(61804028, 68004020)	$2^{26.93}$	-	**Our**

[1] The output difference of the prepositioned differential trail.

[2] We random chose 2^8 master keys and compute the average DL correlation by going though the full plaintext space. For random permutation, the experimental correlation should be about $\pm 2^{-20}$. This information leakage can be used in the distinction between 11-round SPECK32 and random functions, if given the encrypted ciphertext under multiple random keys. Moreover, given sufficient neutral bits of top short-round differential, it can be converted into a new valid distinguisher.

[3] The sample sizes for 11-round DL trail on SPECK32 and SPECK48 are 2^{42} and 2^{46}, respectively.

[4] Distinguishers combining DL trails with the neutral bit technique, denoted by DL(NB). To compare with the DL trails without using NBs, we regard the correlations of DL(NB) as $p^{\frac{1}{2}} r q^2$, since the data complexity required is $\mathcal{O}(pr^{-2}q^{-4})$.

For SPECK, the DL trails searched are shown in Table 5. Unless otherwise noted, the size of the statistical samples is 2^{30}. Here the SPECK32 with a 64-bit key is considered as a vectorial Boolean function $E : \mathbb{F}_2^{96} \rightarrow \mathbb{F}_2^{32}$. Thus, we used 2^{40} samples to count the 11-round DL correlation of SPECK32 when the theoretical correlation is $-2^{-17.37}$.

DL distinguishers combined with neutral bit technique. All DL(NB)s used in this paper are listed in Table 6. For SPECK32, we divide the 10-round DL trail $(0x0211\_0a04, 0x5820\_4020)$ into a 2-round differential $(0x0211\_0a04, 0x0040\_0000)$ with theoretical probability 2^{-6} and a DL trail $(0x0040\_0000, 0x5820\_4020)$ with a correlation of about $-2^{-5.58}$. Thus, if given sufficient neutral bits (i.e., at least $5.58 \times 2 \approx 12$ bits), the DL(NB) will be valid and the data required is $\mathcal{O}(2^{5.58 \times 2 + 6})$. Table 7 presented 14 neutral bits (bit-sets) of the 2-round differential $(0x0211\_0a04, 0x0040\_0000)$. In this case, we regard the correlation of this DL(NB) is about $-2^{-8.58}$.

Table 6: DL distinguishers combined with sufficient neutral bits for round-reduced SPECK32. To compare with the DL trails without using NBs, we regard the correlations of DL(NB) as $p^{\frac{1}{2}}rq^2$, since the data complexity required is $\mathcal{O}(pr^{-2}q^{-4})$.

	R[1]	**Input Diff.**	**Intermediate Diff.**[2]	**Output Mask**	**Prob.**	**Cor.**	$\mathbf{p^{\frac{1}{2}}rq^2}$	**Ref.**
SPECK32	1+9	(2050,2040)	(8000,0100)	(3854,3844)	2^{-2}	-2^{-10}	-2^{-11}	[NSLL22]
	1+10	(2a10,0004)	(2050,2040)	(3854,3844)	2^{-4}	-2^{-12}	-2^{-14}	
	2+8	(0211,0a04)	(0040,0000)	(5820,4020)	2^{-6}	$-2^{-5.58}$	$-2^{-8.58}$	**Our**
	3+8	(8020,4101)	(0040,0000)	(7020,6020)	2^{-12}	$-2^{-5.09}$	$-2^{-11.09}$	
	3+8	(8060,4101)	(0040,0000)	(7020,6020)	2^{-12}	$-2^{-5.09}$	$-2^{-11.09}$	
	3+8	(8021,4101)	(0040,0000)	(7020,6020)	2^{-12}	$-2^{-5.09}$	$-2^{-11.09}$	
	3+8	(8061,4101)	(0040,0000)	(7020,6020)	2^{-12}	$-2^{-5.09}$	$-2^{-11.09}$	

[1] $a+b$ indicates a DL(NB) combining a a-round differential and a b-round DL trail.
[2] The output difference of the prepositioned differential trail.
[3] *Prob.* = Theoretical probability of the prepended short-round differential. *Cor.* = Theoretical correlation of the bottom DL trail, which is equal to the experimental correlation of the whole DL trail divided by *Prob.* Here $p^{\frac{1}{2}}rq^2 = \sqrt{Prob.} \times Cor.$
[4] DL(NB) is valid only if sufficient NB is given. See Table 7 and Table 9 for the NBs we used.

For the 11-round DL trail $(0x0a20\_4205, 0x7020\_6020)$, we divide it into a 3-round differential $(0x0a20\_4205, 0x0040\_0000)$ with theoretical probability 2^{-11} and a 8-round DL trail $(0x0040\_0000, 0x7020\_6020)$ with a correlation of about $-2^{-5.68}$[2]. However, the NBs/SNBSs of the short-round differential are very scarce. We leave the search of NBs/SNBSs of 3-round differential $(0x0a20\_4205, 0x0040\_0000)$ for future research.

Similarly, we divide the 11-round DL trail $(0x8020\_4101, 0x7020\_6020)$ into a 3-round differential $(0x8020\_4101, 0x0040\_0000)$ with theoretical probability 2^{-12} and a 8-round DL trail $(0x0040\_0000, 0x7020\_6020)$ with a correlation of about $-2^{-5.09}$. When requiring the output difference is $0x0040\_0000$, there are four sub-optimal 3-round differentials with probability 2^{-12}, as shown in Table 8. It means that there are another three DL trails with correlations of about $-2^{-17.09}$. If given sufficient neutral bits (i.e., at least $5.09 \times 2 \approx 11$ bits), the DL(NB) will be valid and the data required is $\mathcal{O}(2^{5.09\times 2+12})$. Table 9 presented 13 neutral bits (bit-sets) of the four differentials of Table 8. In this case, we regard the correlations of these four DL(NB)s are about $-2^{-(5.09+\frac{12}{2})} = -2^{-11.09}$. For the validity verification of DL(NB), see the source code provided by this paper.

5.3 Improved Differential-linear Theoretical Attack of SPECK32

This section introduces the general framework of DL attack combined with neutral bit technology. we presented the lowest time-complexity attacks against 12-14 rounds of SPECK32 to date.

We call a ciphertext structure for many pairs of ciphertext generated by a pair of plaintext using neutral bits(bit-sets). Note that for every neutral bit (bit-set) used, the number of ciphertexts in a ciphertext structure doubles.

Bao et al.[BGL+23] presented a re-pairing ciphertext method to reduce the data complexity. Algorithm 1 is a sample using two differences Δ_1 and Δ_2. $\Delta_{NB} = \Delta_1 \oplus \Delta_2$ is the difference corresponding to one neutral bit(-set)s. The plaintext is denoted by p. Once two pairs of input plaintexts $(p, p\oplus\Delta_1)$,$(p\oplus\Delta_{NB}, p\oplus\Delta_1\oplus\Delta_{NB})$ are generated for the differential using Δ_1 as input, one can re-pair the inputs as $(p, p\oplus\Delta_1\oplus\Delta_{NB})$,$(p\oplus\Delta_{NB}, p\oplus\Delta_1)$ and obtain a pair of plaintexts for the differential using Δ_2 as input, which means that one pair of ciphertexts account for one query rather than two queries. This is the case where we used DL(NB) $(0x8020\_4101, 0x7020\_6020)$ and DL(NB) $(0x8060\_4101, 0x7020\_6020)$ to recover key in this paper.

[2] In fact, when requiring the output difference is $0x0040\_0000$, there are two optimal 3-round differentials with probability 2^{-11}, i.e., $(0x0a20\_4205, 0x0040\_0000)$, $(0x0a60\_4205, 0x0040\_0000)$. It means that the correlation of 11-round DL trail $(0x0a60\_4205, 0x7020\_6020)$ is about $-2^{-16.68}$.

The parameters for DL attack combined with neutral bit technology are denoted as follows. The set of neutral bit(-set)s is denoted by NBs.

1. n_{kg} is the number of possible values for the guessed bit of k.

2. n_{cts} is the number of ciphertext structures used in one attack, which is usually several times the reciprocal of top short-round differential probability.

3. n_b is the number of ciphertext pairs in each ciphertext structure and $n_b = 2^{|NBs|}$.

The attack procedure is as follows.

1. Use Algorithm 1 to generate n_{cts} ciphertext structures with size of n_b.

2. For each of guessed key $gk_0, gk_1 \dots gk_{n_{gk}-1}$, one need to decrypt these n_{cts} ciphertext structures. Then use the decrypted data to calculate the correlation of each ciphertext structure. For each guessed key, the maximum correlation of all ciphertext structures can be regarded as the score of the current key.

3. Sort all guessed key by the corresponding score, and the right key is expected to be in the first place.

Algorithm 1: Generate ciphertext structures using the neutral bit(-set)s of differential propagation [BGL+23]: The following is an example of using two input differences Δ_1 and Δ_2, which uses the re-pairing ciphertext method to reduce the data complexity by half. $Range(n)$ represents a set of nonnegative integers less than n. $Random(a, b)$ returns a random integer between a and b. Here $n_b = 2^{|NBs|}$.

```
Input: Number of ciphertext structures used in one attack $n_{cts}$, a set of neutral
       bit(-set)s $NBs$, two input differences $\Delta_1$ and $\Delta_2$, $\Delta_{NB} = \Delta_1 \oplus \Delta_2$, $\Delta_{NB}$ is
       the difference corresponding to $NBs[-1]$.
Output: $CTS$—a 3-D array of ciphertexts, in which $CTS[i][j][0:1]$ and $CTS[i][:][:]$
        represent one pair of ciphertexts and one ciphertext structure respectively.
$P :=$ a array of size $n_{cts}/2 \times n_b \times 2$;
$CTS :=$ a array of size $n_{cts} \times n_b \times 2$;
$P[i][0][0] \leftarrow Random(0, 2^{32} - 1)$, for $0 \le i < n_{cts}/2$ ;
for $0 \le i < |NBs|$ do
    $Diff \leftarrow \bigoplus_{j \in NBs[i]} e_j$. Here $e_j = 0x1 \lll j$ and $NBs[i]$ is a set of indices.;
    $P[g][k + 2^i][0] \leftarrow P[g][k][0] \oplus Diff$, for $g \in Range(n_{cts}/2)$ and $k \in Range(2^i)$;
$P[i][j][1] \leftarrow P[i][j][0] \oplus \Delta_1$, for $i \in Range(n_{cts}/2)$ and $j \in Range(n_b)$;
$CTS[i][j][k] \leftarrow Encrypt(P[i][j][k])$, for all $j$, $k$ and $i \in Range(n_{cts}/2)$;
$CTS[n_{cts}/2 : n_{cts} - 1][:][:] \leftarrow CTS[0 : n_{cts}/2 - 1][:][:]$;
$CTS[i][j][1] \leftrightarrow CTS[i][n_b/2 + j][1]$, for $n_{cts}/2 \le i < n_{cts}$, $j \in Range(n_b/2)$;
return $CTS$;
```

12-round SPECK32 key recovery attack. We use the DL(NB) $(0x0211\_0a04, 0x5820\_4020)$ to mount a 12-round SPECK32 key recovery attack. We note that for SPECK, since no key participates in any operation before the first nonlinear operation, any differential can be extended by one round at no cost. For 12-round SPECK32, the last round key is denote by rk_{11}. We target to recover 15 bits of round key rk_{11} after the distinguisher, since only $\lfloor rk_{11} \rfloor_{15}$ can affect the correlation. Concrete parameters and the complexity are as follows.

$$n_{kg} = 2^{15}, \quad n_{cts} = 2^6, \quad n_b = 2^{12}$$

The data complexity is $n_{cts} \times n_b \times 2 = 2^{19}$. Here the data complexity for generating a pair of ciphertext is 2. The time complexity is $n_{kg} \times n_{cts} \times n_b \times 2 = 2^{34}$.

13-round SPECK32 key recovery attack. By exhausting additional 16 bits of round key, we can use the above 12-round SPECK32 key recovery attack to attack 13-round SPECK32. The data complexity and time complexity are 2^{19} and 2^{50}, respectively.

Next, we use the DL(NB) $(0x8020\_4101, 0x7020\_6020)$ and the DL(NB) $(0x8060\_4101, 0x7020\_6020)$ to mount a 13-round SPECK32 key recovery attack. For 13-round SPECK32, the last round key is denote by rk_{12}. Only $\lfloor rk_{12} \rfloor_{15}$ can affect the correlation. Denote the random data pairs with difference Δ_{in} and the first round key by $(\tilde{x}\|\tilde{y}, \tilde{x}'\|\tilde{y}')$ and rk_0, respectively Note that the input of cipher E are $\mathit{DecryptOneRound}(\tilde{x}\|\tilde{y}, 0)$ and $\mathit{DecryptOneRound}(\tilde{x}'\|\tilde{y}', 0)$, which ensures that the difference after one round is Δ_{in}. Here we use the first 11 NBs in Table 8. As noted by [BGL⁺23], to satisfy the conditions for comforming pairs, i.e., $\begin{cases} \tilde{x}[7] = rk_0[7] \\ \tilde{x}[15] \oplus \tilde{y}[8] = rk_0[15] \oplus rk_0[8] \\ \tilde{x}[0] \oplus \tilde{y}[9] = rk_0[0] \oplus rk_0[9] \end{cases}$, and the conditions for two CSNBSs, i.e., $\begin{cases} \tilde{x}[12] \oplus \tilde{y}[5] \oplus 1 = rk_0[12] \oplus rk_0[5] \\ \tilde{y}[1] = rk_0[1] \end{cases}$, we need to guess five bits of rk_0, i.e., $\begin{cases} rk_0[7], rk_0[15] \oplus rk_0[8], rk_0[0] \oplus rk_0[9] \\ rk_0[12] \oplus rk_0[5], rk_0[1] \end{cases}$. Concrete parameters and the complexity are as follows.

$$n_{kg} = 2^{15+5} = 2^{20}, \quad n_{cts} = 2^{12-3} = 2^9, \quad n_b = 2^{11}$$

[BGL+23] pointed out that for each linear condition in Table 8, once it is fulfilled, the probability of the differential increases by a factor of 2. We generate the random pairs satisfying three conditions in Table 8. Thus, n_{cts} is $2^{12-3} = 2^9$ instead of 2^{12}.

Since all generated ciphertext structures are determined by the guesses of

$$\begin{cases} rk_0[7], rk_0[15] \oplus rk_0[8], rk_0[0] \oplus rk_0[9] \\ rk_0[12] \oplus rk_0[5], rk_0[1] \end{cases},$$

the data complexity is $2^5 \times n_{cts} \times n_b = 2^{25}$. Note that if Algorithm 1 is used, the data complexity for generating a pair of ciphertext is 1. The time complexity is $n_{kg} \times n_{cts} \times n_b \times 2 = 2^{41}$.

However, [BGL+23] pointed out that these two prepended classical differentials, i.e., $(0x8020\_4101, 0x0040\_0000)$ and $(0x8060\_4101, 0x0040\_0000)$, are vaild to keys fulfilling $rk_2[12] \neq rk_2[11]$. Here rk_2 is the third round key for SPECK32. Thus, the presented attack works for 2^{63} keys.

14-round SPECK32 key recovery attack. By exhausting additional 16 bits of round key, we can use the 13-round SPECK32 key recovery attack to attack 14-round SPECK32. The data complexity and time complexity are 2^{25} and 2^{57}, respectively. All results are listed in Table 2.

6 Conclusion and Open Problems

We present a method to compute the extended (rotational) DL correlation of $\boxplus_k$ for $k \geq 2$, where two output linear masks of the cipher pairs can be different. And we present an automated method for evaluating the DL correlations of ARX ciphers, which partially solve the open problem proposed by Niu et al. at CRYPTO 2022. The automated method is applied to some ARX primitives, and improved results are obtained. To the best of our knowledge, this method finds the best differential-linear distinguishers for these ARX primitives at present. Finally, we would like to give some open problems deserving further investigations.

First, it is possible to use an analogous automated method to evaluate DL correlation for other non-ARX ciphers, as long as an automated search method for good DL trails of DLCT is found. The method provided by this paper may be a good template, which combines the correlation propagation technology of difference bits with some theories of DL cryptanalysis for specific ciphers. Secondly, some acceleration techniques need to be studied to find better differential-linear charcateristics on longer rounds. Acceleration technologies in the search of differential/linear characteristics, such as the idea of less variables, Matsui' bounding condition, and divide-and-conquer approach, may be helpful.

References

[AFK+08] Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of latin dances: analysis of salsa, chacha, and rumba. In *International Workshop on Fast Software Encryption*, pages 470–488. Springer, 2008.

[BBCdS+20] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Alzette: A 64-bit arx-box. In *Annual International Cryptology Conference*, pages 419–448. Springer, 2020.

[BBdS+20] Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Lightweight aead and hashing using the sparkle permutation family. *IACR Transactions on Symmetric Cryptology*, pages 208–261, 2020.

[BC04] Eli Biham and Rafi Chen. Near-collisions of sha-0. In *Advances in Cryptology–CRYPTO 2004: 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004. Proceedings 24*, pages 290–305. Springer, 2004.

[BdST+22] Alex Biryukov, Luan Cardoso dos Santos, Je Sen Teh, Aleksei Udovenko, and Vesselin Velichkov. Meet-in-the-filter and dynamic counting with applications to speck. *Cryptology ePrint Archive*, 2022.

[BGG+23] Emanuele Bellini, David Gerault, Juan Grados, Rusydi Makarim, and Thomas Peyrin. Fully automated differential-linear attacks against arx ciphers. Cryptology ePrint Archive, Paper 2023/181, 2023. https://eprint.iacr.org/2023/181.

[BGL+23] Zhenzhen Bao, Jian Guo, Meicheng Liu, Li Ma, and Yi Tu. Enhancing differential-neural cryptanalysis. In *Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part I*, pages 318–347. Springer, 2023.

[bih]

[BJ72] Egon Balas and Robert Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23(1):61–69, 1972.

[BLN17] Céline Blondeau, Gregor Leander, and Kaisa Nyberg. Differential-linear cryptanalysis revisited. *Journal of Cryptology*, 30(3):859–888, 2017.

[BLT20] Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to arx ciphers. In *Annual International Cryptology Conference*, pages 329–358. Springer, 2020.

[BODKW19] Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. Dlct: a new tool for differential-linear cryptanalysis. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 313–342. Springer, 2019.

[BSS$^+$13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *cryptology eprint archive*, 2013.

[CY21] Yi Chen and Hongbo Yu. Bridging machine learning and cryptanalysis via edlct. *Cryptology ePrint Archive*, 2021.

[FWG$^+$16] Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. Milp-based automatic search algorithms for differential and linear trails for speck. In *International Conference on Fast Software Encryption*, pages 268–288. Springer, 2016.

[Gur22] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022.

[LDWRA17] Yunwen Liu, Glenn De Witte, Adrián Ranea, and Tomer Ashur. Rotational-xor cryptanalysis of reduced-round speck. *IACR Transactions on Symmetric Cryptology*, pages 24–36, 2017.

[LH94] Susan K Langford and Martin E Hellman. Differential-linear cryptanalysis. In *Annual International Cryptology Conference*, pages 17–25. Springer, 1994.

[LLL21] Meicheng Liu, Xiaojuan Lu, and Dongdai Lin. Differential-linear cryptanalysis from an algebraic perspective. In *Annual International Cryptology Conference*, pages 247–277. Springer, 2021.

[LM01] Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In *International Workshop on Fast Software Encryption*, pages 336–350. Springer, 2001.

[LSL21] Yunwen Liu, Siwei Sun, and Chao Li. Rotational cryptanalysis from a differential-linear perspective. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 741–770. Springer, 2021.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 386–397. Springer, 1993.

[MPS13] Paweł Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced keccak. In *International Workshop on Fast Software Encryption*, pages 241–262. Springer, 2013.

[NSLL22] Zhongfeng Niu, Siwei Sun, Yunwen Liu, and Chao Li. Rotational differential-linear distinguishers of arx ciphers with arbitrary output linear masks. *Cryptology ePrint Archive*, 2022.

[NW06] Kaisa Nyberg and Johan Wallén. Improved linear distinguishers for snow 2.0. In *International Workshop on Fast Software Encryption*, pages 144–162. Springer, 2006.

[SHW+14] 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*, 2014.

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Accelerating the search of differential and linear characteristics with the sat method. *IACR Transactions on Symmetric Cryptology*, pages 269–315, 2021.

[Wal03] Johan Wallén. On the differential and linear properties of addition. 2003.

[ZWW22] Liu Zhang, Zilong Wang, and Boyang Wang. Improving differential-neural cryptanalysis with inception blocks. *Cryptology ePrint Archive*, 2022.

Supplementary Materials

A The Matrices $A_{8\lambda^0+4\lambda^1+2\alpha^1+\alpha^0}$ of $\boxplus_2$ for All $(\lambda^0, \lambda^1, \alpha^1, \alpha^0) \in \mathbb{F}_2^4$

$$A_0 = \frac{1}{4}\begin{bmatrix} 3 & 1 & 1 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 3 \end{bmatrix} \qquad A_1 = A_2 = \frac{1}{4}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \qquad A_3 = \frac{1}{4}\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A_4 = \frac{1}{4}\begin{bmatrix} -1 & -1 & 1 & -1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix} \qquad A_5 = A_6 = \frac{1}{4}\begin{bmatrix} 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad A_7 = \frac{1}{4}\begin{bmatrix} -2 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A_8 = \frac{1}{4}\begin{bmatrix} -1 & 1 & -1 & -1 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 1 & -1 & 1 \end{bmatrix} \qquad A_9 = A_{10} = \frac{1}{4}\begin{bmatrix} 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad A_{11} = \frac{1}{4}\begin{bmatrix} -2 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A_{12} = \frac{1}{4}\begin{bmatrix} 3 & -1 & -1 & 1 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 1 & -1 & -1 & 3 \end{bmatrix} \qquad A_{13} = A_{14} = \frac{1}{4}\begin{bmatrix} -2 & 1 & 1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & 1 & 1 & -2 \end{bmatrix} \qquad A_{15} = \frac{1}{4}\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & -3 & -1 & 1 \\ 1 & -1 & -3 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

B Extended Rotational Differential-Linear Correlation of $\boxplus_k$

Definition 7. *Let $k > 1$ be a fixed integer. Let the input differences and output masks of two n-bit modulo additions, $\boxplus_{k,*\blacktriangleleft a}$ and $\boxplus_{k,*\blacktriangleleft b}$, be $\alpha^0, ..., \alpha^{k-1} \in \mathbb{F}_2^n$ and $\lambda^0, \lambda^1 \in \mathbb{F}_2^n$, respectively. Then the extended rotational differential-linear correlation can be computed as*

$$Cor_{\substack{*\blacktriangleleft a \\ *\blacktriangleleft b}}(t, \alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = \frac{1}{2^{kn}} \sum_{\substack{(x^i \lll t) \oplus \alpha^i = y^i \\ x^i \in \mathbb{F}_2^n, 0 \le i < k}} (-1)^{\substack{\lambda^0 \cdot (\boxplus_{k,*\blacktriangleleft a}(x^0, x^1, ..., x^{k-1}) \lll t) \\ \oplus \lambda^1 \cdot \boxplus_{k,*\blacktriangleleft b}(y^0, y^1, ..., y^{k-1})}} . \tag{3}$$

Lemma 6. *The extended rotational differential-linear correlation of $\boxplus_k$ with rotational offset t, rotational difference $\alpha^0, ..., \alpha^{k-1}$, linear masks (λ^0, λ^1), and fixed least significant carries $a, b \in \{0, ..., k-1\}$ can be computed as*

$$Cor_{\substack{*\blacktriangleleft a \\ *\blacktriangleleft b}}(t, \alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = \sum_{u,v \in \{0,...,k-1\}} \left(Cor_{\substack{u\blacktriangleleft a \\ *\blacktriangleleft v}}(\lceil \alpha^0 \rceil^{n-t}, ..., \lceil \alpha^{k-1} \rceil^{n-t}, \lfloor \lambda^0 \rfloor^{n-t}, \lfloor \lambda^1 \rfloor^{n-t}) \right)$$
$$\times \left(Cor_{\substack{*\blacktriangleleft u \\ v\blacktriangleleft b}}(\lfloor \alpha^0 \rfloor^{t}, ..., \lfloor \alpha^{k-1} \rfloor^{t}, \lceil \lambda^0 \rceil^{t}, \lceil \lambda^1 \rceil^{t}) \right)$$

where $u, v, a, b \in \{0, ..., k-1\}$.

Proof.

$$
\begin{aligned}
&Cor_{\substack{*\blacktriangleleft a\\ *\blacktriangleleft b}}(t,\alpha^0,...,\alpha^{k-1},\lambda^0,\lambda^1)\\
&=\frac{1}{2^{kn}}\sum_{x^i\in\mathbb{F}_2^n,0\le i<k}(-1)^{\lambda^0\cdot s^0\oplus\lambda^1\cdot s^1}\\
&=\frac{1}{2^{kn}}\sum_{x^i\in\mathbb{F}_2^n,0\le i<k}(-1)^{\lceil\lambda^0\rceil^{n-t}\cdot\lceil s^0\rceil^{n-t}\oplus\lceil\lambda^1\rceil^{n-t}\cdot\lceil s^1\rceil^{n-t}}(-1)^{\lfloor\lambda^0\rfloor^t\cdot\lfloor s^0\rfloor^t\oplus\lfloor\lambda^1\rfloor^t\cdot\lfloor s^1\rfloor^t},\\
&=\frac{1}{2^{kn}}\sum_{x^i\in\mathbb{F}_2^n,0\le i<k}(-1)^{\lceil\lambda^0\rceil^{n-t}\cdot s^2\oplus\lceil\lambda^1\rceil^{n-t}\cdot s^3}(-1)^{\lfloor\lambda^0\rfloor^t\cdot s^4\oplus\lfloor\lambda^1\rfloor^t\cdot s^5}\\
&=\frac{1}{2^{kn}}\sum_{x^i\in\mathbb{F}_2^n,0\le i<k}(-1)^{\lceil\lambda^0\rceil^{n-t}\cdot s^2\oplus\lceil\lambda^1\rceil^{n-t}\cdot s^6}(-1)^{\lfloor\lambda^0\rfloor^t\cdot s^4\oplus\lfloor\lambda^1\rfloor^t\cdot s^7}\\
&=\frac{1}{2^{kn}}\sum_{\substack{u,v\in\{0,...,k-1\}\\ x^i\in\mathbb{F}_2^n,0\le i<k}}(-1)^{\lceil\lambda^0\rceil^{n-t}\cdot s^2\oplus\lceil\lambda^1\rceil^{n-t}\cdot s^6}(-1)^{\lfloor\lambda^0\rfloor^t\cdot s^4\oplus\lfloor\lambda^1\rfloor^t\cdot s^7}\\
&=\sum_{u,v\in\{0,...,k-1\}}\left(\frac{1}{2^{k(n-t)}}\sum_{\substack{c,d\in\{0,...,k-1\}\\ x^i\in\mathbb{F}_2^n,0\le i<k}}(-1)^{\lceil\lambda^0\rceil^{n-t}\cdot s^2\oplus\lceil\lambda^1\rceil^{n-t}\cdot s^8}\right)\times\left(\frac{1}{2^{kt}}\sum_{\substack{c,d\in\{0,...,k-1\}\\ x^i\in\mathbb{F}_2^n,0\le i<k}}(-1)^{\lfloor\lambda^0\rfloor^t\cdot s^9\oplus\lfloor\lambda^1\rfloor^t\cdot s^7}\right)\\
&=\sum_{u,v\in\{0,...,k-1\}}\Big(Cor_{\substack{u\blacktriangleleft a\\ *\blacktriangleleft v}}(\lceil\alpha^0\rceil^{n-t},...,\lceil\alpha^{k-1}\rceil^{n-t},\lceil\lambda^0\rceil^{n-t},\lceil\lambda^1\rceil^{n-t})\Big)\\
&\qquad\qquad\times\Big(Cor_{\substack{*\blacktriangleleft u\\ v\blacktriangleleft b}}(\lfloor\alpha^0\rfloor^t,...,\lfloor\alpha^{k-1}\rfloor^t,\lfloor\lambda^0\rfloor^t,\lfloor\lambda^1\rfloor^t)\Big)
\end{aligned}
$$

where

$$
\begin{aligned}
&s^0=\boxplus_{k,*\blacktriangleleft a}(x^0,...,x^{k-1})\lll t,\\
&s^1=\boxplus_{k,*\blacktriangleleft b}(x^0\lll t\oplus\alpha^0,...,x^{k-1}\lll t\oplus\alpha^{k-1}),\\
&s^2=\boxplus_{k,u\blacktriangleleft a}(\lfloor x^0\rfloor_{n-t},...,\lfloor x^{k-1}\rfloor_{n-t}),\\
&s^3=\boxplus_{k,*\blacktriangleleft v}(\lceil x^0\lll t\oplus\alpha^0\rceil^{n-t},...,\lceil x^{k-1}\lll t\oplus\alpha^{k-1}\rceil^{n-t}),\\
&s^4=\boxplus_{k,*\blacktriangleleft u}(\lceil x^0\rceil_t,...,\lceil x^{k-1}\rceil^t),\\
&s^5=\boxplus_{k,v\blacktriangleleft b}(\lfloor x^0\lll t\oplus\alpha^0\rfloor_t,...,\lfloor x^{k-1}\lll t\oplus\alpha^{k-1}\rfloor_t),\\
&s^6=\boxplus_{k,*\blacktriangleleft v}(\lfloor x^0\oplus(\alpha^0\ggg t)\rfloor_{n-t},...,\lfloor x^{k-1}\oplus(\alpha^{k-1}\ggg t)\rfloor_{n-t}),\\
&s^7=\boxplus_{k,v\blacktriangleleft b}(\lceil x^0\oplus(\alpha^0\ggg t)\rceil^t,...,\lceil x^{k-1}\oplus(\alpha^{k-1}\ggg t)\rceil^t),\\
&s^8=\boxplus_{k,c\blacktriangleleft v}(\lfloor x^0\oplus(\alpha^0\ggg t)\rfloor_{n-t},...,\lfloor x^{k-1}\oplus(\alpha^{k-1}\ggg t)\rfloor_{n-t}),\\
&s^9=\boxplus_{k,d\blacktriangleleft u}(\lceil x^0\rceil_t,...,\lceil x^{k-1}\rceil^t),\\
&u,v,a,b,c,d\in\{0,...,k-1\},\\
&u=\hat{c}(s^2),v=\hat{c}(s^5)=\hat{c}(s^7),\\
&c=\hat{c}(s^8),d=\hat{c}(s^9).
\end{aligned}
$$

□

Next, we use a matrix multiplication chain to compute the extended DL correlation of $\boxplus_k$ for $k\geq 2$.

Theorem 6. *Let $k>1$ be a fixed integer and $l=k\times k$. Let $(t,\alpha^0,...,\alpha^{k-1},\lambda^0,\lambda^1)$ be an extended rotational differential-linear trail of $\boxplus^n_{k,*\blacktriangleleft a}$ and $\boxplus^n_{k,*\blacktriangleleft b}$ for $a,b\in\{0,1,...,k-1\}$.*

Let $z = z_{n-1}, ..., z_1, z_0$ be the word associated with the trail, where $z_i = \lambda_i^0 2^{k+1} + \lambda_i^1 2^k + \sum_{j=0}^{k-1} \alpha_i^j 2^j$. Then the correlation can be computed as

$$Cor_{\substack{*\blacktriangleleft a \\ *\blacktriangleleft b}}(t, \alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = \sum_{0\le v,j<k} \left(\left[\prod_{0\le i<t} A_{z_i} \right] H(b) \left[\prod_{t\le i<n} A_{z_i} \right] \right)_{jk+v,ak+v}.$$

where $H(b)$ is a $l \times l$ matrix with $H(b)_{u\times k+b, u\times k+i} = 1$ for all $0 \le u, i < k$ and zero otherwise.

Proof. Let $C(x, y)$ is a column vector of dimension l with a single 1 in $(x \times k + y)$-th row and zero otherwise. Let $L(*, v)$ is a row vector of dimension l with a 1 in $(i \times k + v)$-th column for all $0 \le i < k$ and zero otherwise. Let $L(u, *)$ is a row vector of dimension l with a 1 in $(u \times k + i)$-th column for all $0 \le i < k$ and zero otherwise.

According to Lemma 6, we have

$$\begin{aligned}
Cor_{\substack{*\blacktriangleleft a \\ *\blacktriangleleft b}}(t, \alpha^0, ..., \alpha^{k-1}, \lambda^0, \lambda^1) = & \sum_{0\le u,v<k} \left(Cor_{\substack{u\blacktriangleleft a \\ *\blacktriangleleft v}}(\lceil\alpha^0\rceil^{n-t}, ..., \lceil\alpha^{k-1}\rceil^{n-t}, \lceil\lambda^0\rceil^{n-t}, \lceil\lambda^1\rceil^{n-t}) \right) \\
& \times \left(Cor_{\substack{*\blacktriangleleft u \\ v\blacktriangleleft b}}(\lfloor\alpha^0\rfloor^t, ..., \lfloor\alpha^{k-1}\rfloor^t, \lfloor\lambda^0\rfloor^t, \lfloor\lambda^1\rfloor^t) \right) \\
= & \sum_{0\le v<k} \sum_{0\le u<k} \left(L(*, v) \left[\prod_{0\le i<t} A_{z_i} \right] C(u, b) \right) \times \left(L(u, *) \left[\prod_{t\le i<n} A_{z_i} \right] C(a, v) \right) \\
= & \sum_{0\le v<k} \left(L(*, v) \left[\prod_{0\le i<t} A_{z_i} \right] H(b) \left[\prod_{t\le i<n} A_{z_i} \right] C(a, v) \right) \\
= & \sum_{0\le v,j<k} \left(\left[\prod_{0\le i<t} A_{z_i} \right] H(b) \left[\prod_{t\le i<n} A_{z_i} \right] \right)_{jk+v,ak+v}.
\end{aligned}$$

□

C Neutral Bit-sets Used in Key Recovery Attacks

Based on one pair of plaintext with one differential established, neutral bit-sets (NB) are mainly used to generate more plaintext pairs with the differential trail established. In this study, 2-round differential $(0x0211\_0a04, 0x0040\_0000)$ is used in the 10-round DL trail we presented. Reference [BGL+23] presents the probabilistic Simultaneous-neutral bit-sets (SNBS) for 2-round differential trail (0x0211_0a04,0x0040_0000) of SPECK32, as shown in Table 7.

Table 7: (Probabilistic) SNBS's for 2-round differential trail (0x0211_0a04,0x0040_0000) of SPECK32/64.

NB	Pr.	NB	Pr.	NB	Pr.	NB	Pr.	NB	Pr.	NB	Pr.	NB	Pr.
[20]	1	[21]	1	[22]	1	[9,16]	1	[2,11,25]	1	[14]	0.965	[15]	0.938
[6,29]	0.91	[23]	0.812	[30]	0.809	[7]	0.806	[0]	0.754	[11,27]	0.736	[8]	0.664

When requiring the output difference is $0x0040\_0000$, there are two optimal 3-round differentials with probability 2^{-11}, i.e., $(0x0a20\_4205, 0x0040\_0000)$, $(0x0a60\_4205, 0x0040\_0000)$.

However, the NBs/SNBSs of these two differentials are very scarce. There are four sub-optimal 3-round differentials with probability 2^{-12}. Table 8 presents these four differential and three sufficient conditions to conform the differentials.

Table 8: Three sufficient conditions conform the 3-round sub-optimal differentials with output difference $0x0040\_0000$ [BGL+23]. Let the result after 1-round encryption of SPECK32 be (x, y). Here $c = ((x \ggg 7) \boxplus y) \oplus ((x \ggg 7) \oplus y)$. [BGL+23] pointed that for each linear condition, once it is fulfilled, the probability of the differential increases by a factor of 2.

InputDiff.	**0x8020_4101**	**0x8060_4101**	**0x8021_4101**	**0x8061_4101**
Round 1	**0x0201_0604**			
Round 2	**0x1800_0010**			
Round 3	**0x0040_0000**			
$x[7]$	0	0	0	0
$x[5] \oplus y[14]$	1	0	1	0
$x[15] \oplus y[8]$	0	0	1	1
	$x[0] \oplus y[9] = 0$	$x[0] \oplus y[9] = 0$	$c[9] \oplus y[9] = 0$	$c[9] \oplus y[9] = 0$

Table 9: Neutral bit/bit-sets for 3-round differentials ($0x8020\_4101$, $0x0040\_0000$), ($0x8060\_4101$, $0x0040\_0000$), ($0x8021\_4101$, $0x0040\_0000$), ($0x8061\_4101$, $0x0040\_0000$) of SPECK32 [BGL+23].

NB	**0x8020_4101**		**0x8060_4101**		**0x8021_4101**		**0x8061_4101**		
	Pre.[2]	**Post.**[3]	**Pre.**	**Post.**	**Pre.**	**Post.**	**Pre.**	**Post.**	**Condition**[1]
[22]	0.995	1.000	0.995	1.000	0.996	1.000	0.997	1.000	-
[20]	0.986	1.000	0.997	1.000	0.996	1.000	0.995	1.000	-
[13]	0.986	1.000	0.989	1.000	0.988	1.000	0.992	1.000	-
[12,19]	0.986	1.000	0.995	1.000	0.993	1.000	0.986	1.000	-
[14,21]	0.855	0.860	0.874	0.871	0.881	0.873	0.881	0.876	-
[6,29]	0.901	0.902	0.898	0.893	0.721	0.706	0.721	0.723	-
[30]	0.803	0.818	0.818	0.860	0.442	0.442	0.412	0.407	-
[0,8,31]	0.855	0.859	0.858	0.881	0.000	0.000	0.000	0.000	-
[5,28]	0.495	1.000	0.495	1.000	0.481	1.000	0.469	1.000	$x[12] \oplus y[5] = 1$
[15,24]	0.482	1.000	0.542	1.000	0.498	1.000	0.496	1.000	$y[1] = 0$
[4,27,29]	0.672	0.916	0.648	0.905	0.535	0.736	0.536	0.718	$x[11] \oplus y[4] = 1$
[6,11,12,18]	0.445	0.903	0.456	0.906	0.333	0.701	0.382	0.726	$x[2] \oplus y[11] = 0$

[1] A condition at the end of a row is specific to the bit-set at the same row. '-' means that there is no condition for the corresponding bit-set.

[2] Pre.: probability obtained using 1000 correct pairs without imposing the conditions.

[3] Post.: probability obtained using 1000 correct pairs and imposing all conditions in the last column.

D The Differences Between Our Work and Others [BGG+23].

We and Bellini et al. [BGG+23] independently conducted automatic searches for differential-linear (DL) trails of ARX ciphers. However, there are differences between our work and that of Bellini et al.

In particular, Bellini et al. relied on Morawiecki et al.'s correlation propagation method, which is called correlation propagation of difference bits in this paper. Liu et al. [LSL21] pointed out at EUROCRYPT 2021 that this method may give inaccurate analysis results and overlook some valuable DL trails since the probabilities quickly converge to $\frac{1}{2}$ for all one-bit output masks after a few iterative evaluations of the round function.

Additionally, Niu et al. demonstrated at CRYPTO 2020 that Morawiecki et al.'s method cannot compute accurate DL correlations of additions, and presented a matrix multiplication chain for obtaining more precise results. In this work, we simplified Niu et al.'s method and applied it to the automatic search of DL trails for ARX ciphers.

Table 10: The differences between our work and [BGG+23].

	Bellini et al.′s Work[BGG+23]	**Ours**
Time	First public on January 15, 2023.	Our paper has been submitted to *** since the second half of 2022.
DL cryptanalysis of additions.	None	Section 3.
Implementation of log2 function	Approximating by using piecewise linear functions	ready-made functions in Gurobi, i.e. LOG_2 and ABS.
Distinguishers/Key recovery attack	-	Our work is better.
Application	SPECK32	Alzette,SPECK32/48/64