Rotational Differential-Linear Distinguishers of ARX Ciphers with Arbitrary Output Linear Masks

Zhongfeng Niu[1,2,3], Siwei Sun[2,5⋆], Yunwen Liu, Chao Li[4]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China niuzhongfeng@iie.ac.cn
[2] School of Cryptology, University of Chinese Academy of Sciences, China sunsiwei@ucas.ac.cn
[3] School of Cyber Security, University of Chinese Academy of Sciences, China
[4] College of Liberal arts and Science, National University of Defense Technology, China
[5] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

Abstract. The rotational differential-linear attacks, proposed at EUROCRYPT 2021, is a generalization of differential-linear attacks by replacing the differential part of the attacks with rotational differentials. At EUROCRYPT 2021, Liu et al. presented a method based on Morawiecki et al.'s technique (FSE 2013) for evaluating the rotational differential-linear correlations for the special cases where the output linear masks are unit vectors. With this method, some powerful (rotational) differential-linear distinguishers with output linear masks being unit vectors against `FRIET`, `Xoodoo`, and `Alzette` were discovered. However, how to compute the rotational differential-linear correlations for arbitrary output masks was left open. In this work, we partially solve this open problem by presenting an efficient algorithm for computing the (rotational) differential-linear correlation of modulo additions for arbitrary output linear masks, based on which a technique for evaluating the (rotational) differential-linear correlation of ARX ciphers is derived. We apply the technique to `Alzette`, `SipHash`, `ChaCha`, and `SPECK`. As a result, significantly improved (rotational) differential-linear distinguishers including *deterministic* ones are identified. All results of this work are practical and experimentally verified to confirm the validity of our methods. In addition, we try to explain the experimental distinguishers employed in FSE 2008, FSE 2016, and CRYPTO 2020 against `ChaCha`. The predicted correlations are close to the experimental ones.

Keywords: Rotational differential-linear, Correlation, ARX, `Alzette`, `SipHash`, `SPECK`, `ChaCha`

1 Introduction

Building symmetric-key primitives with modulo additions, rotations, and XORs is a common practice in the community of symmetric-key cryptography. The result-

⋆ The corresponding author

ing primitives are collectively referred as ARX designs and their representatives can be found everywhere, including

- Block ciphers: FEAL [42], Bel-T [38], LEA [24], TEA [16], XTEA [41], HIGHT [25], SPECK [6], SPARX [19];
- Stream ciphers: Salsa20 [11], ChaCha20 [10];
- Hash functions: SHA3 finalists Skein [22] and BLAKE [5];
- Cryptographic permutations: Alzette [7], Sparkle [8];
- MAC algorithms: SipHash [3], Chaskey [35].

Some ARX designs are standardized or widely deployed in real world applications. For example, HIGHT, LEA, and Chaskey are standardized in ISO/IEC 18033-3:2010, ISO/IEC 29192-2:2019, and ISO/IEC 29192-6:2019, respectively. ChaCha is used with HMAC-SHA1 and Poly1305 in the transport layer security (TLS) protocol. Chaskey is deployed in commercial products by some automotive suppliers and major industrial control systems. Skein has been added to FreeBSD and is optionally used for authentication tags in the ZRTP protocol. Variants of BLAKE are included in OpenSSL and WolfSSL. In addition, instances of SipHash are used in the dnscache instances of all OpenDNS resolvers and employed as hash() in Python for all major platforms.

The popularity of ARX designs can be attributed to the following reasons. Firstly, modulo additions provide both diffusion and confusion functionalities, making it possible to construct secure primitives without relying on the table look-ups associated with the S-box based designs, which increases the resilience against timing side-channel attacks. Secondly, the native support of the modulo additions in modern CPUs allows particularly fast software implementations of ARX ciphers. Finally, the code describing an ARX primitive is relatively simple and small, making this approach especially appealing for application scenarios where the memory footprint is highly constrained. In a systematic work for evaluating the performance and resource consumption of lightweight block ciphers on three major micro-controller platforms (8-bit AVR, 16-bit MSP, and 32-bit ARM) [18], Dinu et al. concluded:

> *"... state-of-the art ARX and ARX-like designs are not only very fast, but also extremely small in terms of RAM footprint and code size."*

Cryptanalysis of ARX Primitives. ARX designs hold a special position in the development of techniques for analyzing symmetric-key primitives. The block cipher FEAL [42], probably the first ARX cipher presented in the literature, has acted as a catalyst in the discovery of differential and linear cryptanalysis. However, compared to S-box based designs, the development of the theories and tools for the analysis of ARX-like primitives tends to lag behind when the involved additions operate on n-bit words with $n \geq 16$.

In S-box based designs, typically the employed S-boxes are small permutations (e.g., permutations over $\mathbb{F}_2^4$ or $\mathbb{F}_2^8$) whose differential property can be computed by enumerating the input pairs. In contrast, the modulo additions often operate

on large words (e.g., 32-bit or 64-bit words). In such cases, computing the probability of a given differential $(\alpha, \beta) \to \gamma$ by enumeration is computationally infeasible. The first algorithm for computing the differential probabilities of modulo additions efficiently was not available until 2001 [31]. After two years, Wallén showed how to compute the correlations of the linear approximations of modulo additions efficiently [44]. Subsequently, alternative descriptions of the cryptographic properties of modulo additions with S-functions [37] and finite automaton [40] appeared. The development of the tools for constructing or finding differential or linear trails of ARX-like ciphers has gone through multiple stages. At first, tools working as helpers for manual analysis were developed [27–29]. Then, dedicated search algorithms are designed to identify differential trails with high probabilities [12, 13]. Now, we have constraint-based (MILP, SAT, or SMT) tools which are quite powerful and convenient in designing and analyzing ARX primitives [23, 36].

In recent years, we witness remarkable advancement in the cryptanalysis of ARX primitives [2, 9, 15, 26, 30, 32, 33]. Nevertheless, there are full of open problems concerning the cryptanalysis of ARX designs. For example, we do not know how to compute the accurate probabilities or correlations of the differential or linear approximations of a chain of modulo additions [21]. There are attacks published at top crypto conferences relying on experimental distinguishers without a theoretical interpretation [9, 15], and we refer the reader to Supplementary Material H in the extended version of this paper [39] for a systematic summary of these experimental distinguishers. Most recently, Liu et al. presented the so-called rotational differential-linear cryptanalysis and proposed the open problem on computing the (rotational) differential-linear correlations of modulo additions with output linear masks of Hamming weight greater than one [32], which is the major problem we are going to solve in this work.

Contribution. First of all, we solve the open problem proposed in [32]. We present a method for computing the (rotational) differential-linear correlation of the modulo addition for arbitrary output linear masks based on a delicate partition of $\mathbb{F}_2^m \times \mathbb{F}_2^m$ into subsets, where the elements in each subset fulfill certain equations. The method is extremely efficient, and the time complexity of computing the (rotational) differential-linear correlation of $x \boxplus y \mod 2^n$ for a specific rotational differential-linear approximation can be roughly estimated by the complexity of n 4×4 matrix multiplications.

Based on the above method and Morawiecki et al.'s technique [34], we propose a method for computing the generalized (rotational) differential-linear correlation of ARX ciphers with arbitrary output linear masks when the probabilities of $x_{i-t} \neq x_i$ for all relevant i's and a specific t are given. Compared with the formulas given in [32], the new ones are not only applicable for output linear masks whose Hamming weights are greater than one, but also weaken the assumptions required for the formulas to hold. We apply the method to `Alzette`, `SipHash`, `ChaCha`, and `SPECK`. We identify new and significantly improved (rotational) differential-linear distinguishers. All the new distinguishers are highly biased or even *deterministic*,

and all of them are experimentally verified. The results are summarized in Table 1.

Table 1: A summary of the results. R-DL = rotational differential-linear, DL = differential-linear, RD = rotational differential, LC = linear characteristic, DC = differential characteristic. We show differentials with probabilities and LC/DL/R-DL with correlations. Note that the 10-round RD distinguisher for `SPECK32` works only for $2^{28.10}$ weak keys, and the constants used in the experiments for `Alzette` are `0xB7E15162` and `0x38B4DA56`.

Permutation	Type	# Round	Probability/Correlation		Ref.
			Theoretical	Experimental	
`Alzette`	DC	4	2^{-6}	–	[7]
	R-DL	4	$2^{-11.37}$	$2^{-7.35}$	[32]
	DL	4	$2^{-0.27}$	$2^{-0.1}$	[32]
	DC	8	$\leq 2^{-32}$	–	[7]
	DL	4	1	1	Sect. 6.1
	R-DL	4	$2^{-5.57}$	$2^{-3.14}$	Sect. 6.1
	DL	5	$-2^{-0.33}$	$-2^{-0.13}$	Sect. 6.1
	DL	6	$2^{-4.95}$	$2^{-1.45}$	Sect. 6.1
	DL	8	$-2^{-8.24}$	$-2^{-5.50}$	Sect. 6.1
`SipHash`	DC	4	2^{-35}	–	[20]
	DL	3	$2^{-2.19}$	$2^{-0.78}$	Sect. 6.2
	DL	4	$2^{-12.45}$	$2^{-6.03}$	Sect. 6.2
`SPECK32`	DC	8	2^{-24}	–	[1]
	LC	9	2^{-14}	–	[23]
	DC	10	$2^{-31.01}$	–	[43]
	RD	10*	$2^{-19.15}$	–	[33]
	DL	8	$2^{-8.23}$	$2^{-6.87}$	Sect. 6.3
	DL	9	$2^{-10.23}$	$2^{-8.93}$	Sect. 6.3
	DL	10	$2^{-15.23}$	$2^{-13.90}$	Sect. 6.3
`ChaCha`	DL	4	–	$2^{-1.19}$	[14]
	DL	4	$2^{-0.02}$	$2^{-0.98}$	Sect. 6.4

In addition, we attempt to give theoretical interpretations of the experimental distinguishers employed in CRYPTO 2020 [9], FSE 2008 [4], and FSE 2016 [14] against `ChaCha`. The results of the analysis are summarized in Table 13 in the Supplementary Material G in the extended version [39], from which we can see that the predicted correlations are close to the experimental ones.

2 Notations and Preliminaries

For a finite set $\mathbb{D}$, $\#\mathbb{D}$ denotes the number of elements in $\mathbb{D}$. Let $\mathbb{F}_2 = \{0, 1\}$ be the binary field. We denote by x_i the i-th bit of a vector $\mathbf{x} = (x_{n-1}, \cdots, x_0) \in \mathbb{F}_2^n$. In addition, $\lceil \mathbf{x} \rceil^{(t)} = (x_{n-1}, \cdots, x_{n-t})$ denotes the most significant t bits of $\mathbf{x}$, and $\lfloor \mathbf{x} \rfloor^{(t)} = (x_{t-1}, \cdots, x_0)$ denotes the least significant t bits of $\mathbf{x}$. Concrete values in $\mathbb{F}_2^n$ are specified in hexadecimal or binary notations. For example, we use $\mathtt{0x1F12}$ or $\mathtt{1F12}$ to denote the binary string $(\mathtt{0001\ 1111\ 0001\ 0010})_2$. Given two n-bit vectors $\mathbf{x} = (x_{n-1}, \cdots, x_0)$ and $\mathbf{y} = (y_{n-1}, \cdots, y_0)$, the inner product of $\mathbf{x}$ and $\mathbf{y}$ is defined as $\mathbf{x} \cdot \mathbf{y} = x_{n-1}y_{n-1} \oplus \cdots \oplus x_0 y_0$. For a constant vector $\boldsymbol{\lambda} \in \mathbb{F}_2^n$, $\boldsymbol{\lambda}^\perp$ represents the set $\{\mathbf{x} \in \mathbb{F}_2^n : \boldsymbol{\lambda} \cdot \mathbf{x} = 0\}$. Rotation of $\mathbf{x}$ by t bits to the left is denoted by $\mathbf{x} \lll t$, and when t is clear from the context, $\mathbf{x} \lll t$ is written as $\overleftarrow{\mathbf{x}}$ for simplicity. The rotational-xor difference (RX-difference) with offset t of two bit strings $\mathbf{x}$ and $\mathbf{x}'$ in $\mathbb{F}_2^n$ are defined as $(x \lll t) \oplus x'$.

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function. We use $\overleftarrow{F}$ to denote the function mapping $\mathbf{x}$ to $F(\mathbf{x}) \lll t$ for some non-negative integer t. The correlation of the rotational differential-linear approximation of F with rotation offset t, RX-difference $\boldsymbol{\alpha} \in \mathbb{F}_2^n$, and output linear mask $\boldsymbol{\lambda} \in \mathbb{F}_2^m$ is defined as

$$\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\alpha},\boldsymbol{\lambda}}(F) = \frac{1}{2^n} \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda} \cdot ((F(\mathbf{x}) \lll t) \oplus F((\mathbf{x} \lll t) \oplus \boldsymbol{\alpha}))}. \tag{1}$$

When $t = 0$, Equation (1) computes the ordinary differential-linear correlation of F, which is denoted by $\mathcal{C}^{\text{DL}}_{\boldsymbol{\alpha},\boldsymbol{\lambda}}(F)$. When F is clear from the context, we may drop F and use $\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\alpha},\boldsymbol{\lambda}}$ and $\mathcal{C}^{\text{DL}}_{\boldsymbol{\alpha},\boldsymbol{\lambda}}$ instead.

Let $\mathbf{M}_i$ for $0 \le i < n$ be $k \times k$ matrices, we use $\prod_{i=0}^{n-1} \mathbf{M}_i$ to denote the product with the specified order $\mathbf{M}_{n-1} \cdots \mathbf{M}_0$.

2.1 Modulo Addition with an Initial Carry Bit

Let $\boxplus_b^{(n)} : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the operation mapping $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ to

$$\mathbf{x} \boxplus_b^{(n)} \mathbf{y} = \mathbf{x} + \mathbf{y} + b \mod 2^n,$$

where $b \in \mathbb{F}_2$. For the sake of simplicity, we may omit the subscript b when $b = 0$ or the superscript (n) when n is clear from the context.

Example 1. Let $\mathbf{x} = \mathtt{0xE9} = (\mathtt{11101001})_2$ and $\mathbf{y} = \mathtt{0xA3} = (\mathtt{10100011})_2$ be 8-bit strings. Then, $\mathbf{x} \boxplus \mathbf{y} = \mathbf{x} \boxplus_0^{(8)} \mathbf{y} = \mathtt{0x8C} = (\mathtt{10001100})_2$, and $\mathbf{x} \boxplus_1^{(8)} \mathbf{y} = \mathtt{0x8D} = (\mathtt{10001101})_2$.

For $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, the carry vector of $(\mathbf{x}, \mathbf{y})$ with initial carry bit $b \in \mathbb{F}_2$ is defined to be an $(n+1)$-bit vector $\mathfrak{c}_b(\mathbf{x}, \mathbf{y}) = (c_n, c_{n-1}, \cdots, c_0)$ such that

$$c_i = \begin{cases} b, & i = 0 \\ x_{i-1}y_{i-1} \oplus x_{i-1}c_{i-1} \oplus y_{i-1}c_{i-1}, & 1 \le i \le n \end{cases}.$$

We call $\mathfrak{c}_b(\mathbf{x},\mathbf{y})[n]$ the *most significant carry* of $\mathbf{x} \boxplus_b^{(n)} \mathbf{y}$, denoted as $\hat{\mathfrak{c}}_b(\mathbf{x},\mathbf{y})$. Under these notations, $\mathbf{x} \boxplus_b^{(n)} \mathbf{y} = \mathbf{x} \oplus \mathbf{y} \oplus \lfloor \mathfrak{c}_b(\mathbf{x},\mathbf{y}) \rfloor^{(n-1)}$. Moreover,

$$\mathfrak{c}_b(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) = \lfloor \mathfrak{c}_b(\mathbf{x},\mathbf{y}) \rfloor^{(k+1)}$$

is a $(k+1)$-bit vector, and $\hat{\mathfrak{c}}_b(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) = \mathfrak{c}_b(\mathbf{x},\mathbf{y})[k]$.

Example 2. Let $\mathbf{x} = \mathtt{0xE9} = (\mathtt{11101001})_2$ and $\mathbf{y} = \mathtt{0xA3} = (\mathtt{10100011})_2$ be 8-bit strings. Then, $\mathfrak{c}_0(\mathbf{x},\mathbf{y}) = (\mathtt{111000110})_2 \in \mathbb{F}_2^9$, $\mathfrak{c}_1(\mathbf{x},\mathbf{y}) = (\mathtt{111000111})_2 \in \mathbb{F}_2^9$, $\hat{\mathfrak{c}}_0(\mathbf{x},\mathbf{y}) = \hat{\mathfrak{c}}_1(\mathbf{x},\mathbf{y}) = 1$, $\mathfrak{c}_0(\lceil \mathbf{x} \rceil^{(4)}, \lceil \mathbf{y} \rceil^{(4)}) = \mathfrak{c}_0((\mathtt{1110})_2, (\mathtt{1010})_2) = (\mathtt{11100})_2 \in \mathbb{F}_2^5$, and $\mathfrak{c}_1(\lfloor \mathbf{x} \rfloor^{(4)}, \lfloor \mathbf{y} \rfloor^{(4)}) = \mathfrak{c}_1((\mathtt{1001})_2, (\mathtt{0011})_2) = (\mathtt{00111})_2 \in \mathbb{F}_2^5$. Moreover, $\hat{\mathfrak{c}}_1(\lfloor \mathbf{x} \rfloor^{(4)}, \lfloor \mathbf{y} \rfloor^{(4)}) = 0$, and $\hat{\mathfrak{c}}_0(\lceil \mathbf{x} \rceil^{(4)}, \lceil \mathbf{y} \rceil^{(4)}) = 1$

Finally, the following lemma is frequently used in the subsequent sections.

Lemma 1. *For* $(a,b) \in \mathbb{F}_2 \times \mathbb{F}_2$, $(-1)^{a \oplus b} = (-1)^a (-1)^b$.

2.2 Useful Partitions of $\mathbb{F}_2^k \times \mathbb{F}_2^k$

We now present some partition schemes of the sets $\mathbb{F}_2^k \times \mathbb{F}_2^k$ for $k \le n$. Note that *being familiar with these partition schemes is essential for understanding the methodology of this paper.*

Definition 1. *Given* $(a,b) \in \mathbb{F}_2^2$, $(u,v) \in \mathbb{F}_2^2$, *and* $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, *for* $1 \le k \le n$, *we use* $\mathbb{D}^{(k)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\boldsymbol{\alpha},\boldsymbol{\beta}) \subseteq \mathbb{F}_2^k \times \mathbb{F}_2^k$ *to denote the set*

$$\{(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k : (\hat{\mathfrak{c}}_a(\boldsymbol{x},\boldsymbol{y}), \hat{\mathfrak{c}}_b(\boldsymbol{x} \oplus \lfloor \boldsymbol{\alpha} \rfloor^{(k)}, \boldsymbol{y} \oplus \lfloor \boldsymbol{\beta} \rfloor^{(k)}) = (u,v)\}.$$

Under this notation, we have

$$\mathbb{D}^{(n)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \{(\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : (\hat{\mathfrak{c}}_a(\mathbf{x},\mathbf{y}), \hat{\mathfrak{c}}_b(\mathbf{x} \oplus \boldsymbol{\alpha}, \mathbf{y} \oplus \boldsymbol{\beta}) = (u,v)\}.$$

and $\mathbb{D}^{(1)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\alpha_i,\beta_i) = \{(x,y) \in \mathbb{F}_2^2 : (\hat{\mathfrak{c}}_a(x,y), \hat{\mathfrak{c}}_b(x \oplus \alpha_i, y \oplus \beta_i) = (u,v)\} \subseteq \mathbb{F}_2^2$, which is the solution set of the following system of equations

$$\begin{cases} xy \oplus xa \oplus ya = u \\ (x \oplus \alpha_i)(y \oplus \beta_i) \oplus (x \oplus \alpha_i)b \oplus (y \oplus \beta_i)b = v \end{cases}.$$

In our notation, $\mathbb{D}^{(k)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\boldsymbol{\alpha},\boldsymbol{\beta})$ only depends on the least significant k bits of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and thus some readers may think it is more natural to always write $\mathbb{D}^{(k)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\lfloor \boldsymbol{\alpha} \rfloor^{(k)}, \lfloor \boldsymbol{\beta} \rfloor^{(k)})$. However, to make the notations shorter, we prefer the former one.

Example 3. $\mathbb{D}^{(1)}_{\substack{0 \blacktriangleleft 0 \\ 1 \blacktriangleleft 1}}(\mathtt{0},\mathtt{1}) = \{\mathtt{00},\mathtt{10}\}$, $\mathbb{D}^{(1)}_{\substack{1 \blacktriangleleft 1 \\ 1 \blacktriangleleft 1}}(\mathtt{0},\mathtt{0}) = \{\mathtt{01},\mathtt{10},\mathtt{11}\}$, and $\mathbb{D}^{(1)}_{\substack{0 \blacktriangleleft 0 \\ 0 \blacktriangleleft 1}}(\mathtt{1},\mathtt{1}) = \emptyset$. We refer the reader to Supplementary Material A in the extended version of this paper [39] for $\mathbb{D}^{(1)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\alpha,\beta)$ with all combinations of $(\alpha,\beta,a,b,u,v) \in \mathbb{F}_2^6$.

Example 4. Let $\boldsymbol{\alpha} = \mathtt{011}$ and $\boldsymbol{\beta} = \mathtt{100} \in \mathbb{F}_2^3$. Then, $\mathbb{D}^{(3)}_{\substack{0 \blacktriangleleft 0 \\ 1 \blacktriangleleft 1}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \subseteq \mathbb{F}_2^6$ contains the following twenty elements written as binary vectors:

$\mathtt{000000}$, $\mathtt{000001}$, $\mathtt{000010}$, $\mathtt{000011}$, $\mathtt{001001}$, $\mathtt{001010}$, $\mathtt{001011}$, $\mathtt{010010}$, $\mathtt{010011}$, $\mathtt{011011}$,
$\mathtt{100000}$, $\mathtt{100001}$, $\mathtt{100010}$, $\mathtt{100011}$, $\mathtt{101000}$, $\mathtt{110001}$, $\mathtt{101010}$, $\mathtt{110000}$, $\mathtt{101001}$, $\mathtt{111000}$,

$\mathbb{D}^{(2)}_{\substack{1 \blacktriangleleft 1 \\ 1 \blacktriangleleft 1}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \{\mathtt{0011}, \mathtt{0110}, \mathtt{0111}, \mathtt{1010}, \mathtt{1011}, \mathtt{1111}\}$, and $\mathbb{D}^{(2)}_{\substack{0 \blacktriangleleft 0 \\ 0 \blacktriangleleft 1}}(\lceil \boldsymbol{\alpha} \rceil^{(2)}, \lceil \boldsymbol{\beta} \rceil^{(2)}) = \{\mathtt{0010}, \mathtt{0011}, \mathtt{0100}, \mathtt{0110}\}$.

Lemma 2. *For any fixed $(a, b) \in \mathbb{F}_2^2$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$,*

$$\mathbb{F}_2^n \times \mathbb{F}_2^n = \bigcup_{(u,v) \in \mathbb{F}_2^2} \mathbb{D}^{(n)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{2}$$

and the necessary and sufficient condition for

$$\mathbb{D}^{(n)}_{\substack{u \blacktriangleleft a \\ v \blacktriangleleft b}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \bigcap \mathbb{D}^{(n)}_{\substack{u' \blacktriangleleft a \\ v' \blacktriangleleft b}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \neq \emptyset$$

is $(u, v) = (u', v')$.

Proof. According to Definition 1, Equation (2) is obvious. The second part holds because the solution sets of

$$\begin{cases} \hat{\mathfrak{c}}_a(\mathbf{x}, \mathbf{y}) = u \\ \hat{\mathfrak{c}}_b(\mathbf{x} \oplus \boldsymbol{\alpha}, \mathbf{y} \oplus \boldsymbol{\beta}) = v \end{cases} \quad \text{and} \quad \begin{cases} \hat{\mathfrak{c}}_a(\mathbf{x}, \mathbf{y}) = u' \\ \hat{\mathfrak{c}}_b(\mathbf{x} \oplus \boldsymbol{\alpha}, \mathbf{y} \oplus \boldsymbol{\beta}) = v' \end{cases}$$

have a common solution if and only if $(u, v) = (u', v')$. □

Lemma 3. *Let $\mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}} \parallel \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ be the set of all $(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ such that*

$$\begin{cases} (\lceil \mathbf{x} \rceil^{(t)}, \lceil \mathbf{y} \rceil^{(t)}) \in \mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}}(\lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lfloor \boldsymbol{\beta} \rfloor^{(t)}) \\ (\lfloor \mathbf{x} \rfloor^{(n-t)}, \lfloor \mathbf{y} \rfloor^{(n-t)}) \in \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\lceil \boldsymbol{\alpha} \rceil^{(n-t)}, \lceil \boldsymbol{\beta} \rceil^{(n-t)}) \end{cases}. \tag{3}$$

Then, the necessary and sufficient condition for

$$\left(\mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}} \parallel \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \right) \bigcap \left(\mathbb{D}^{(t)}_{\substack{b' \blacktriangleleft u' \\ v' \blacktriangleleft 0}} \parallel \mathbb{D}^{(n-t)}_{\substack{u' \blacktriangleleft 0 \\ a' \blacktriangleleft v'}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \right) \neq \emptyset \tag{4}$$

is $(a, b, u, v) = (a', b', u', v')$. Moreover, we have

$$\bigcup_{(a,b) \in \mathbb{F}_2^2} \bigcup_{(u,v) \in \mathbb{F}_2^2} \left(\mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}} \parallel \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \right) = \mathbb{F}_2^n \times \mathbb{F}_2^n.$$

Proof. Equation (4) implies that

$$\begin{cases} \mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}}(\lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lfloor \boldsymbol{\beta} \rfloor^{(t)}) \bigcap \mathbb{D}^{(t)}_{\substack{b' \blacktriangleleft u' \\ v' \blacktriangleleft 0}}(\lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lfloor \boldsymbol{\beta} \rfloor^{(t)}) \neq \emptyset \\ \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\lceil \boldsymbol{\alpha} \rceil^{(n-t)}, \lceil \boldsymbol{\beta} \rceil^{(n-t)}) \bigcap \mathbb{D}^{(n-t)}_{\substack{u' \blacktriangleleft 0 \\ a' \blacktriangleleft v'}}(\lceil \boldsymbol{\alpha} \rceil^{(n-t)}, \lceil \boldsymbol{\beta} \rceil^{(n-t)}) \neq \emptyset \end{cases},$$

which in turn implies $v = v'$, $u = u'$, $a = a'$, and $b = b'$ according to Definition 1. The second part of the lemma comes from the fact that any element in $\mathbb{F}_2^{2n}$ must satisfy Equation (3) for some (a, b, u, v). □

Remark 1. To make the description of our methods compact and expressive, the symbols employed in this work are complex. Therefore, we accompany the paper with a `SageMath Notebook` file at https://github.com/ZhongfengNiu/rot-differential-linear to help the readers to familiarize with the notations.

3 Ordinary Differential-Linear Correlation of $\boxplus$

Before diving into the details, we emphasize that this section is the key part of the paper, and there is no essential difference between the methods for computing the ordinary differential-linear correlation and the *rotational* differential-linear correlation. For the ease of the reader, we single out this section to avoid the technical complexities introduced by the rotational differentials. We strongly encourage the reader to go through the details of the proofs in this section. Moreover, we provide a `SageMath Notebook` file at https://github.com/ZhongfengNiu/rot-differential-linear for computing the correlations of ordinary and rotational differential-linear approximations of modulo additions with arbitrary output linear masks.

Definition 2. *The differential-linear correlation of* $S(\mathbf{x},\mathbf{y}) = \mathbf{x} \boxplus \mathbf{y}$ *with input difference* $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, and output linear mask* $\boldsymbol{\lambda} \in \mathbb{F}_2^n$ *is defined as*

$$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}}(S) = \frac{1}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda}\cdot(S(\mathbf{x}\oplus\boldsymbol{\alpha},\mathbf{y}\oplus\boldsymbol{\beta})\oplus S(\mathbf{x},\mathbf{y}))}.$$

Let $F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) = (-1)^{\lfloor\boldsymbol{\lambda}\rfloor^{(k)}\cdot\left(\lfloor S(\mathbf{x}\oplus\boldsymbol{\alpha},\mathbf{y}\oplus\boldsymbol{\beta})\rfloor^{(k)} \,\oplus\, \lfloor S(\mathbf{x},\mathbf{y})\rfloor^{(k)}\right)}$ for $1 \le k \le n$. Thus, $F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y})$ can be fully determined by the least significant k-bits of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\lambda}$, $\mathbf{x}$, and $\mathbf{y}$. Under this notation, we have

$$2^{2n}\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}}(S) = \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}), \tag{5}$$

In addition, according to the partition given by Equation (2),

$$\sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) = \sum_{(u,v)\in\mathbb{F}_2^2} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(n)}_{\substack{u\blacktriangleleft 0\\ v\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}),$$

or in the matrix notation, we have

$$\sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) = \begin{pmatrix}1&1&1&1\end{pmatrix} \begin{pmatrix} \sum\limits_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(n)}_{\substack{0\blacktriangleleft 0\\ 0\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(n)}_{\substack{0\blacktriangleleft 0\\ 1\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(n)}_{\substack{1\blacktriangleleft 0\\ 0\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(n)}_{\substack{1\blacktriangleleft 0\\ 1\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \end{pmatrix}. \tag{6}$$

For $1 \le k \le n$, let $\mathbf{V}^{(k)}$ be the column vector

$$\begin{pmatrix} \sum_{(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) \in \mathbb{D}^{(k)}_{\substack{0 \blacktriangleleft 0 \\ 0 \blacktriangleleft 0}}(\boldsymbol{\alpha}, \boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) \\ \sum_{(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) \in \mathbb{D}^{(k)}_{\substack{0 \blacktriangleleft 0 \\ 1 \blacktriangleleft 0}}(\boldsymbol{\alpha}, \boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) \\ \sum_{(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) \in \mathbb{D}^{(k)}_{\substack{1 \blacktriangleleft 0 \\ 0 \blacktriangleleft 0}}(\boldsymbol{\alpha}, \boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) \\ \sum_{(\lfloor \mathbf{x} \rfloor^{(k)}, \lfloor \mathbf{y} \rfloor^{(k)}) \in \mathbb{D}^{(k)}_{\substack{1 \blacktriangleleft 0 \\ 1 \blacktriangleleft 0}}(\boldsymbol{\alpha}, \boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) \end{pmatrix}.$$

Then, according to Equation (5) and Equation (6),

$$2^{2n} \mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}}(S) = (1,1,1,1)\mathbf{V}^{(n)}.$$

Now, we are going to derive a recursive relationship between $\mathbf{V}^{(k)}$ and $\mathbf{V}^{(k-1)}$.

Lemma 4. *For $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\lambda}$, $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{F}_2^n$, Let $\mathbf{z} = \mathbf{x} \boxplus \mathbf{y}$ and $\mathbf{z}' = \mathbf{x}' \boxplus \mathbf{y}'$, where $\mathbf{x}' = \mathbf{x} \oplus \boldsymbol{\alpha}$ and $\mathbf{y}' = \mathbf{y} \oplus \boldsymbol{\beta}$. Then we have*

$$F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) = (-1)^{\lambda_{k-1} \cdot (\alpha_{k-1} \oplus \beta_{k-1} \oplus u \oplus v)} F^{(k-1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}),$$

where $u = \hat{\mathfrak{c}}_0(\lfloor \mathbf{x} \rfloor^{(k-1)}, \lfloor \mathbf{y} \rfloor^{(k-1)})$ and $v = \hat{\mathfrak{c}}_0(\lfloor \mathbf{x}' \rfloor^{(k-1)}, \lfloor \mathbf{y}' \rfloor^{(k-1)})$.

Proof. It comes from the fact that

$$\begin{aligned} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}) &= (-1)^{\lfloor \boldsymbol{\lambda} \rfloor^{(k)} \cdot (\lfloor \mathbf{z} \rfloor^{(k)} \oplus \lfloor \mathbf{z}' \rfloor^{(k)})} \\ &= (-1)^{\lambda_{k-1} \cdot (z_{k-1} \oplus z'_{k-1})} (-1)^{\lfloor \lambda \rfloor^{(k-1)} \cdot (\lfloor \mathbf{z} \rfloor^{(k-1)} \oplus \lfloor \mathbf{z}' \rfloor^{(k-1)})} \\ &= (-1)^{\lambda_{k-1} \cdot (z_{k-1} \oplus z'_{k-1})} F^{(k-1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x}, \mathbf{y}), \end{aligned}$$

and $z_{k-1} \oplus z'_{k-1} = \alpha_{k-1} \oplus \beta_{k-1} \oplus \hat{\mathfrak{c}}_0(\lfloor \mathbf{x} \rfloor^{(k-1)}, \lfloor \mathbf{y} \rfloor^{(k-1)}) \oplus \hat{\mathfrak{c}}_0(\lfloor \mathbf{x}' \rfloor^{(k-1)}, \lfloor \mathbf{y}' \rfloor^{(k-1)})$. □

For $(a, b, u, v) \in \mathbb{F}_2^4$, let

$$\pi_{2a+b,2u+v}(\alpha_t, \beta_t, \lambda_t) = (-1)^{\lambda_t(\alpha_t \oplus \beta_t \oplus u \oplus v)} \#\mathbb{D}^{(1)}_{\substack{a \blacktriangleleft u \\ b \blacktriangleleft v}}(\alpha_t, \beta_t).$$

and

$$\mathbf{M}_{\alpha_t,\beta_t,\lambda_t} = \begin{pmatrix} \pi_{0,0}(\alpha_t,\beta_t,\lambda_t), \pi_{0,1}(\alpha_t,\beta_t,\lambda_t), \pi_{0,2}(\alpha_t,\beta_t,\lambda_t), \pi_{0,3}(\alpha_t,\beta_t,\lambda_t) \\ \pi_{1,0}(\alpha_t,\beta_t,\lambda_t), \pi_{1,1}(\alpha_t,\beta_t,\lambda_t), \pi_{1,2}(\alpha_t,\beta_t,\lambda_t), \pi_{1,3}(\alpha_t,\beta_t,\lambda_t) \\ \pi_{2,0}(\alpha_t,\beta_t,\lambda_t), \pi_{2,1}(\alpha_t,\beta_t,\lambda_t), \pi_{2,2}(\alpha_t,\beta_t,\lambda_t), \pi_{2,3}(\alpha_t,\beta_t,\lambda_t) \\ \pi_{3,0}(\alpha_t,\beta_t,\lambda_t), \pi_{3,1}(\alpha_t,\beta_t,\lambda_t), \pi_{3,2}(\alpha_t,\beta_t,\lambda_t), \pi_{3,3}(\alpha_t,\beta_t,\lambda_t) \end{pmatrix}.$$

Note that $\#\mathbb{D}^{(1)}_{\substack{a \blacktriangleleft u \\ b \blacktriangleleft v}}(\alpha_t, \beta_t)$ can be derived from Table 8 in Supplementary Material A in the extended version [39], and the concrete values for $\mathbf{M}_{\alpha_t,\beta_t,\lambda_t}$ for all possible $(\alpha_t, \beta_t, \lambda_t) \in \mathbb{F}_2 \times \mathbb{F}_2 \times \mathbb{F}_2$ are listed in Supplementary Material C in the extended version [39]. Then, we have the following two lemmas.

Lemma 5. $\mathbf{V}^{(1)} = \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}(1,0,0,0)^T$.

Proof. Since $F^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) = (-1)^{\lambda_0\cdot((x_0\oplus\alpha_0)\oplus(y_0\oplus\beta_0)\oplus(x_0\oplus y_0))} = (-1)^{\lambda_0\cdot(\alpha_0\oplus\beta_0)}$, $\mathbf{V}^{(1)}$ is equal to

$$\begin{pmatrix} \sum\limits_{\substack{(x_0,y_0)\in\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft 0\\0\blacktriangleleft 0}}(\alpha_0,\beta_0)}} F^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{\substack{(x_0,y_0)\in\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft 0\\1\blacktriangleleft 0}}(\alpha_0,\beta_0)}} F^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{\substack{(x_0,y_0)\in\mathbb{D}^{(1)}_{\substack{1\blacktriangleleft 0\\0\blacktriangleleft 0}}(\alpha_0,\beta_0)}} F^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\ \sum\limits_{\substack{(x_0,y_0)\in\mathbb{D}^{(1)}_{\substack{1\blacktriangleleft 0\\1\blacktriangleleft 0}}(\alpha_0,\beta_0)}} F^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \end{pmatrix} = \begin{pmatrix} (-1)^{\lambda_0\cdot(\alpha_0\oplus\beta_0)}\#\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft 0\\0\blacktriangleleft 0}}(\alpha_0,\beta_0) \\ (-1)^{\lambda_0\cdot(\alpha_0\oplus\beta_0)}\#\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft 0\\1\blacktriangleleft 0}}(\alpha_0,\beta_0) \\ (-1)^{\lambda_0\cdot(\alpha_0\oplus\beta_0)}\#\mathbb{D}^{(1)}_{\substack{1\blacktriangleleft 0\\0\blacktriangleleft 0}}(\alpha_0,\beta_0) \\ (-1)^{\lambda_0\cdot(\alpha_0\oplus\beta_0)}\#\mathbb{D}^{(1)}_{\substack{1\blacktriangleleft 0\\1\blacktriangleleft 0}}(\alpha_0,\beta_0) \end{pmatrix} = \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}.$$

□

Lemma 6. *For* $1 \le k < n$, $\mathbf{V}^{(k+1)} = \mathbf{M}_{\alpha_k,\beta_k,\lambda_k}\mathbf{V}^{(k)}$.

Proof. We only need to prove

$$\mathbf{V}^{(k+1)}[i] = \sum_{j=0}^{3}\mathbf{M}_{\alpha_k,\beta_k,\lambda_k}[i][j]\mathbf{V}^{(k)}[j] = \sum_{j=0}^{3}\pi_{i,j}(\alpha_k,\beta_k,\lambda_k)\mathbf{V}^{(k)}[j] \tag{7}$$

for $0 \le i < 4$. Here, we only show that Equation (7) holds for $i = 0$. For $1 \le i < 4$, the proof is similar. Let $u = \hat{\mathfrak{c}}_0(\lfloor\mathbf{x}\rfloor^{(k)}, \lfloor\mathbf{y}\rfloor^{(k)})$ and $v = \hat{\mathfrak{c}}_0(\lfloor\mathbf{x}'\rfloor^{(k)}, \lfloor\mathbf{y}'\rfloor^{(k)})$. Then, we have

$$\begin{aligned}
\mathbf{V}^{(k+1)}[0] &= \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(k+1)}_{\substack{0\blacktriangleleft 0\\0\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(k+1)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\
&= \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{D}^{(k+1)}_{\substack{0\blacktriangleleft 0\\0\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} (-1)^{\lambda_k\cdot(\alpha_k\oplus\beta_k\oplus u\oplus v)} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\
&= \sum_{(u,v)\in\mathbb{F}_2^2}\ \sum_{(x_k,y_k)\in\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft u\\0\blacktriangleleft v}}(\alpha_k,\beta_k)}\ \sum_{(\lfloor x\rfloor^{(k)},\lfloor y\rfloor^{(k)})\in\mathbb{D}^{(k)}_{\substack{u\blacktriangleleft 0\\v\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} (-1)^{\lambda_k\cdot(\alpha_k\oplus\beta_k\oplus u\oplus v)} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\
&= \sum_{(u,v)\in\mathbb{F}_2^2} (-1)^{\lambda_k\cdot(\alpha_k\oplus\beta_k\oplus u\oplus v)} \left(\sum_{(x_k,y_k)\in\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft u\\0\blacktriangleleft v}}(\alpha_k,\beta_k)} 1\right) \sum_{(\lfloor x\rfloor^{(k)},\lfloor y\rfloor^{(k)})\in\mathbb{D}^{(k)}_{\substack{u\blacktriangleleft 0\\v\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\
&= \sum_{(u,v)\in\mathbb{F}_2^2} (-1)^{\lambda_k\cdot(\alpha_k\oplus\beta_k\oplus u\oplus v)} \#\mathbb{D}^{(1)}_{\substack{0\blacktriangleleft u\\0\blacktriangleleft v}}(\alpha_k,\beta_k) \sum_{(\lfloor x\rfloor^{(k)},\lfloor y\rfloor^{(k)})\in\mathbb{D}^{(k)}_{\substack{u\blacktriangleleft 0\\v\blacktriangleleft 0}}(\boldsymbol{\alpha},\boldsymbol{\beta})} F^{(k)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y}) \\
&= \sum_{j=0}^{3}\pi_{0,j}(\alpha_k,\beta_k,\lambda_k)\mathbf{V}^{(k)}[j].
\end{aligned}$$

□

Theorem 1. *The differential-linear correlation of the modulo addition* $\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}} = \frac{1}{2^{2n}} \cdot \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} F^{(n)}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(\mathbf{x},\mathbf{y})$*, can be computed as*

$$\frac{1}{2^{2n}}\left(1,1,1,1\right)\mathbf{V}^{(n)} = \frac{1}{2^{2n}}\left(1,1,1,1\right)\mathbf{M}^{(n-1)}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}}\cdots\mathbf{M}^{(0)}_{\alpha_0,\beta_0,\lambda_0}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}.$$

Proof. It comes from Lemma 5 and Lemma 6. □

Next, we present two simple corollaries to show the applications of Theorem 1. Note that these corollaries can also be proved by Definition 2.

Corollary 1. *For any given* $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, and* $\boldsymbol{\lambda} \in \mathbb{F}_2^n$ *such that* $\lceil\boldsymbol{\lambda}\rceil^{(n-1)} = 0^{n-1}$*. The absolute differential-linear correlation of* $\boxplus^{(n)}$ *is* $|\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}}| = 1$*.*

Proof. Since $(1,1,1,1)\mathbf{M}_{\alpha_t,\beta_t,0} = 2^2\cdot(1,1,1,1)$ for arbitrary $(\alpha_t,\beta_t)\in\mathbb{F}_2^2$,

$$\begin{aligned}\left(1,1,1,1\right)\mathbf{V}^{(n)} &= \left(1,1,1,1\right)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},0}\cdots\mathbf{M}_{\alpha_1,\beta_1,0}\mathbf{M}_{\alpha_0,\beta_0,\lambda_0}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\\ &= 2^{2(n-1)}\left(1,1,1,1\right)\mathbf{M}_{\alpha_0,\beta_0,\lambda_0}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \pm 2^{2n}.\end{aligned}$$

□

Corollary 2. *For any* $\boldsymbol{\lambda} \in \mathbb{F}_2^n$*, and* $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{F}_2^{2n}$ *such that* $\lfloor\boldsymbol{\alpha}\rfloor^{(n-1)} = \lfloor\boldsymbol{\beta}\rfloor^{(n-1)} = 0^{n-1}$*, The absolute differential-linear correlation of* $\boxplus^{(n)}$ *is* $|\mathcal{C}^{\mathrm{DL}}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}| = 1$*.*

Proof. Let r be a real number. Then,

$$\mathbf{M}_{0,0,\lambda_t}(r,0,0,1-r)^T = 2^2\cdot(r',0,0,1-r')^T,$$

for some real number r'. Therefore,

$$\begin{aligned}\left(1,1,1,1\right)\mathbf{V}^{(n)} &= \left(1,1,1,1\right)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}}\mathbf{M}_{0,0,\lambda_{n-2}}\cdots\mathbf{M}_{0,0,\lambda_0}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\\ &= 2^{2(n-1)}\cdot\left(1,1,1,1\right)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}}\begin{pmatrix}p\\0\\0\\1-p\end{pmatrix}\\ &= \pm 2^{2n}\cdot\left(1,\pm1,\pm1,1\right)\begin{pmatrix}p\\0\\0\\1-p\end{pmatrix} = \pm 2^{2n}.\end{aligned}$$

□

Next, we give some concrete analysis of differential-linear approximations of modulo additions over 32-bit integers with output linear masks being $\mathbf{e}_i \oplus \mathbf{e}_{i+1}$ whose Hamming weights are 2, where $\mathbf{e}_i$ denotes the ith unit vector. Note that in this work the least significant bit is indexed by 0, and thus $\mathbf{e}_0 = \mathtt{00\cdots001}$. The analysis of 64-bit and 128-bit modulo additions can be found in Supplementary Material E in the extended version of this paper [39].

Table 2: The correlations of example differential-linear approximations of $\boxplus_0^{(32)}$.

i	0	1	2	3	4	5	6	7	8	9	10	11
$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	0	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{7}{8}$	$\frac{15}{16}$	$\frac{31}{32}$	$\frac{1}{64}$	$-\frac{65}{128}$	0	$\frac{1}{2}$	$-\frac{3}{4}$	0
i	12	13	14	15	16	17	18	19	20	21	22	23
$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	0	$-\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{5}{8}$	0	0	0
i	24	25	26	27	28	29	30					
$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	$-\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$					

Example 5. Consider the 32-bit modulo addition. Let $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathbb{F}^{32} \times \mathbb{F}_2^{32}$ be the input difference

$$\begin{cases} \boldsymbol{\alpha} = (\mathtt{10111010110001000011011111000001})_2 \\ \boldsymbol{\beta} = (\mathtt{10000100001001111110111011000000})_2 \end{cases}.$$

Then, the differential-linear correlations $\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$ can be computed with Theorem 1, and the results are listed in Table 2.

4 Rotational Differential-Linear Correlation of $\boxplus$

Definition 3. *According to Equation (1), the correlation of the modulo addition $S(\mathbf{x},\mathbf{y}) = \mathbf{x} \boxplus \mathbf{y}$ with input difference $(\boldsymbol{\alpha},\boldsymbol{\beta})$ and output linear mask $\boldsymbol{\lambda}$ is*

$$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}}(S) = \frac{1}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda}\cdot\left[(((\mathbf{x}\lll t)\oplus\boldsymbol{\alpha})\boxplus((\mathbf{y}\lll t)\oplus\boldsymbol{\beta}))\oplus((\mathbf{x}\boxplus\mathbf{y})\lll t)\right]}.$$

Lemma 7. *The rotational differential-linear correlation of $\boxplus$ with rotational offset t, rotational difference $(\boldsymbol{\alpha},\boldsymbol{\beta})$, and linear mask $\boldsymbol{\lambda}$ can be computed as*

$$\frac{1}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda}\cdot\boldsymbol{\Delta}} = \frac{1}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} (-1)^{\lceil\boldsymbol{\lambda}\rceil^{(n-t)}\cdot\lceil\boldsymbol{\Delta}\rceil^{(n-t)}} (-1)^{\lfloor\boldsymbol{\lambda}\rfloor^{(t)}\cdot\lfloor\boldsymbol{\Delta}\rfloor^{(t)}},$$

where

$$\begin{cases} \boldsymbol{\Delta} = \big(((\mathbf{x}\lll t)\oplus\boldsymbol{\alpha})\boxplus((\mathbf{y}\lll t)\oplus\boldsymbol{\beta})\big)\oplus\big((\mathbf{x}\boxplus\mathbf{y})\lll t\big) \\ \lceil\boldsymbol{\Delta}\rceil^{(n-t)} = \big((\lfloor\mathbf{x}\rfloor^{(n-t)}\oplus\lceil\boldsymbol{\alpha}\rceil^{(n-t)})\boxplus_v(\lfloor\mathbf{y}\rfloor^{(n-t)}\oplus\lceil\boldsymbol{\beta}\rceil^{(n-t)})\big)\oplus\Big(\lfloor\mathbf{x}\rfloor^{(n-t)}\boxplus\lfloor\mathbf{y}\rfloor^{(n-t)}\Big) \\ \lfloor\boldsymbol{\Delta}\rfloor^{(t)} = \big((\lceil\mathbf{x}\rceil^{(t)}\oplus\lfloor\boldsymbol{\alpha}\rfloor^{(t)})\boxplus(\lceil\mathbf{y}\rceil^{(t)}\oplus\lfloor\boldsymbol{\beta}\rfloor^{(t)})\big)\oplus\big(\lceil\mathbf{x}\rceil^{(t)}\boxplus_u\lceil\mathbf{y}\rceil^{(t)}\big) \end{cases},$$

and

$$\begin{cases} u = \mathfrak{c}_0(\lfloor \mathbf{x} \rfloor^{(n-t)}, \lfloor \mathbf{y} \rfloor^{(n-t)}) \\ v = \mathfrak{c}_0(\lceil \mathbf{x} \rceil^{(t)} \oplus \lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lceil \mathbf{y} \rceil^{(t)} \oplus \lfloor \boldsymbol{\beta} \rfloor^{(t)}) \end{cases}.$$

Proof. Let $\mathbf{z} = (\mathbf{x} \boxplus \mathbf{y}) \lll t$ and $\mathbf{z}' = ((\mathbf{x} \lll t) \oplus \boldsymbol{\alpha}) \boxplus ((\mathbf{y} \lll t) \oplus \boldsymbol{\beta})$. Then,

$$\begin{cases} \lceil \mathbf{z} \rceil^{(n-t)} = \lfloor \mathbf{x} \rfloor^{(n-t)} \boxplus \lfloor \mathbf{y} \rfloor^{(n-t)} \\ \lfloor \mathbf{z} \rfloor^{(t)} = \lceil \mathbf{x} \rceil^{(t)} \boxplus_u \lceil \mathbf{y} \rceil^{(t)} \end{cases},$$

where $u = \mathfrak{c}_0(\lfloor \mathbf{x} \rfloor^{(n-t)}, \lfloor \mathbf{y} \rfloor^{(n-t)})$. Similarly,

$$\begin{cases} \lceil \mathbf{z}' \rceil^{(n-t)} = (\lfloor \mathbf{x} \rfloor^{(n-t)} \oplus \lceil \boldsymbol{\alpha} \rceil^{(n-t)}) \boxplus_v (\lfloor \mathbf{y} \rfloor^{(n-t)} \oplus \lceil \boldsymbol{\beta} \rceil^{(n-t)}) \\ \lfloor \mathbf{z}' \rfloor^{(t)} = (\lceil \mathbf{x} \rceil^{(t)} \oplus \lfloor \boldsymbol{\alpha} \rfloor^{(t)}) \boxplus (\lceil \mathbf{y} \rceil^{(t)} \oplus \lfloor \boldsymbol{\beta} \rfloor^{(t)}) \end{cases},$$

where $v = \mathfrak{c}_0(\lceil \mathbf{x} \rceil^{(t)} \oplus \lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lceil \mathbf{y} \rceil^{(t)} \oplus \lfloor \boldsymbol{\beta} \rfloor^{(t)})$. Consequently,

$$\boldsymbol{\lambda} \cdot \boldsymbol{\Delta} = (\lceil \boldsymbol{\lambda} \rceil^{(n-t)} \cdot \lceil \boldsymbol{\Delta} \rceil^{(n-t)}) \oplus (\lfloor \boldsymbol{\lambda} \rfloor^{(t)} \cdot \lfloor \boldsymbol{\Delta} \rfloor^{(n-t)}).$$

Applying Lemma 1 to $(-1)^{\boldsymbol{\lambda} \cdot \boldsymbol{\Delta}}$ gives the proof. □

Lemma 8. *Let* $u = \mathfrak{c}_0(\lfloor \mathbf{x} \rfloor^{(n-t)}, \lfloor \mathbf{y} \rfloor^{(n-t)})$ *and* $v = \mathfrak{c}_0(\lceil \mathbf{x} \rceil^{(t)} \oplus \lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lceil \mathbf{y} \rceil^{(t)} \oplus \lfloor \boldsymbol{\beta} \rfloor^{(t)})$. *Then,*

$$\sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda} \cdot \boldsymbol{\Delta}} = \sum_{(u,v) \in \mathbb{F}_2^2} \Psi(u,v) \Phi(u,v),$$

where $\Psi(u,v)$ *equals to*

$$\sum_{b \in \mathbb{F}_2} \sum_{(\lceil \mathbf{x} \rceil^{(t)}, \lceil \mathbf{y} \rceil^{(t)}) \in \mathbb{D}^{(t)}_{\substack{b \blacktriangleleft u \\ v \blacktriangleleft 0}}(\lfloor \boldsymbol{\alpha} \rfloor^{(t)}, \lfloor \boldsymbol{\beta} \rfloor^{(t)})} (-1)^{\lfloor \boldsymbol{\lambda} \rfloor^{(t)} \cdot \lfloor \boldsymbol{\Delta} \rfloor^{(t)}},$$

and $\Phi(u,v)$ *equals to*

$$\sum_{a \in \mathbb{F}_2} \sum_{(\lfloor \mathbf{x} \rfloor^{(n-t)}, \lfloor \mathbf{y} \rfloor^{(n-t)}) \in \mathbb{D}^{(n-t)}_{\substack{u \blacktriangleleft 0 \\ a \blacktriangleleft v}}(\lceil \boldsymbol{\alpha} \rceil^{(n-t)}, \lceil \boldsymbol{\beta} \rceil^{(n-t)})} (-1)^{\lceil \boldsymbol{\lambda} \rceil^{(n-t)} \cdot \lceil \boldsymbol{\Delta} \rceil^{(n-t)}}.$$

Proof. See Supplementary Material B in the extended version [39]. □

Theorem 2. *The rotational differential-linear correlation of* $\boxplus$ *with rotational offset* t*, rotational difference* $(\boldsymbol{\alpha}, \boldsymbol{\beta})$*, and linear mask* $\boldsymbol{\lambda}$ *can be computed as*

$$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\boldsymbol{\lambda}} = \frac{1}{2^{2n}} (1,0,1,0)\, \mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{2^{2n}} (0,1,0,1)\, \mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

where

$$\mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}} = \prod_{i=0}^{t-1} \mathbf{M}_{\alpha_i,\beta_i,\lambda_i} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \prod_{j=t}^{n-1} \mathbf{M}_{\alpha_j,\beta_j,\lambda_j}.$$

Proof. Applying similar techniques used in the proof of Theorem 1, for $(u, v) = (0, 0)$, we can derive that

$$\begin{cases} \Psi(0,0) = (1,0,1,0)\mathbf{M}_{\alpha_{t-1},\beta_{t-1},\lambda_{t-1}} \cdots \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}(1,0,0,0)^T \\ \Phi(0,0) = (1,1,0,0)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}} \cdots \mathbf{M}_{\alpha_t,\beta_t,\lambda_t}(1,0,0,0)^T \end{cases}.$$

For $(u, v) = (0, 1)$, we have

$$\begin{cases} \Psi(0,1) = (0,1,0,1)\mathbf{M}_{\alpha_{t-1},\beta_{t-1},\lambda_{t-1}} \cdots \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}(1,0,0,0)^T \\ \Phi(0,1) = (1,1,0,0)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}} \cdots \mathbf{M}_{\alpha_t,\beta_t,\lambda_t}(0,1,0,0)^T \end{cases}.$$

For $(u, v) = (1, 0)$, we have

$$\begin{cases} \Psi(1,0) = (1,0,1,0)\mathbf{M}_{\alpha_{t-1},\beta_{t-1},\lambda_{t-1}} \cdots \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}(0,0,1,0)^T \\ \Phi(1,0) = (0,0,1,1)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}} \cdots \mathbf{M}_{\alpha_t,\beta_t,\lambda_t}(1,0,0,0)^T \end{cases}.$$

For $(u, v) = (1, 1)$, we have

$$\begin{cases} \Psi(1,1) = (0,1,0,1)\mathbf{M}_{\alpha_{t-1},\beta_{t-1},\lambda_{t-1}} \cdots \mathbf{M}_{\alpha_0,\beta_0,\lambda_0}(0,0,1,0)^T \\ \Phi(1,1) = (0,0,1,1)\mathbf{M}_{\alpha_{n-1},\beta_{n-1},\lambda_{n-1}} \cdots \mathbf{M}_{\alpha_t,\beta_t,\lambda_t}(0,1,0,0)^T \end{cases}.$$

According to Definition 3 and Lemma 8, $2^{2n}\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}} = \sum_{(u,v)\in\mathbb{F}_2^2} \Psi(u,v)\Phi(u,v)$ can be computed as

$$\big(\Psi(0,0)\Phi(0,0) + \Psi(1,0)\Phi(1,0)\big) + \big(\Psi(0,1)\Phi(0,1) + \Psi(1,1)\Phi(1,1)\big),$$

which is equal to

$$\big(1,\,0,\,1,\,0\big)\,\mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(1,0,0,0)^T + \big(0,\,1,\,0,\,1\big)\,\mathbf{C}_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\lambda}}(0,1,0,0)^T.$$

□

Next, we give some concrete rotational differential-linear analysis of modulo additions over 32-bit integers, where $\mathbf{e}_i$ denotes the ith unit vector. The analysis of 64-bit and 128-bit addition can be found in Supplementary Material F in the extended version of this paper [39].

Example 6. Consider the 32-bit modulo addition. Let $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$ be the input difference such that

$$\begin{cases} \boldsymbol{\alpha} = (\texttt{10110000000100100101100000110010})_2 \\ \boldsymbol{\beta} = (\texttt{10100001011101110100110001110011})_2 \end{cases}.$$

Then, the rotational differential-linear correlations $\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$ with rotation offset $t = 30$ can be computed with Theorem 1, and the results are listed in Table 3.

Table 3: The correlations of example rotational differential-linear approximations of $\boxplus_0^{(32)}$ with rotation offset $t = 30$.

i	0	1	2	3	4	5	6	7
$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{5}{8}$	$\frac{3}{16}$	$-\frac{19}{32}$	0	$\frac{1}{2}$
i	8	9	10	11	12	13	14	15
$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	$\frac{3}{4}$	$-\frac{7}{8}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{3}{8}$
i	16	17	18	19	20	21	22	23
$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{1}{4}$	0	0	$-\frac{1}{2}$
i	24	25	26	27	28	29	30	
$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i\oplus\mathbf{e}_{i+1}}$	0	$\frac{1}{2}$	$\frac{3}{4}$	$-\frac{7}{8}$	0	$\frac{1}{4294967296}$	$\frac{1073741825}{2147483648}$	

Remark 2. Theorem 1 and Theorem 2 completely generalize the formulas presented in [32]. More importantly, the formulas presented in Theorem 1 and Theorem 2 efficiently compute the *exact* correlations of arbitrary rotational differential-linear distinguishers of modulo additions, while the formulas in [32] can only compute *approximations* of the correlations of rotational differential-linear distinguishers of modulo additions with output linear masks being unit vectors. The formulas given in [32] are not exact since they rely on certain statistical assumptions that may not hold perfectly in practice. This difference can be observed in Example 7.

Example 7. Consider the 32-bit modulo addition. Let $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$ be the input difference

$$\begin{cases} \boldsymbol{\alpha} = (\texttt{01100011101110001111101101010111})_2 \\ \boldsymbol{\beta} = (\texttt{01010011001111111110100111110011})_2 \end{cases}.$$

Then, the rotational differential-linear correlations $\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i}$ with rotation offset $t = 30$ can be computed with the formula presented in [32] or Theorem 1 in this work, and the results are listed in Table 4.

Table 4: The correlations $\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),\mathbf{e}_i}$ of example rotational differential-linear approximations of $\boxplus_0^{(32)}$ with rotation offset $t = 30$ and output masks being unit vectors

i	0	1	2	3	4	5	6	7
[32]	0	−0.5	−0.75	−0.875	−0.0625	0	0	0.5
This work	0.25	−0.375	−0.6875	−0.84375	−0.078125	0	0	0.5

5 Computing the (Rotational) Differential-Linear Correlation of Iterative ARX Primitives

Previous sections focus on the analysis of the *local* properties of modulo additions. In this section, we show how to efficiently compute the (rotational) differential-linear correlations of ARX primitives based on the theories developed in previous sections and Morawiecki's technique [34].

To extend Morawiecki's technique to evaluate the correlations of arbitrary rotational differential-linear approximations of a cipher, one must be able to compute the rotational differential-linear correlation (with an arbitrary output linear mask) of each building block $F : \mathbb{F}_2^m \to \mathbb{F}_2^n$ of the cipher being analyzed with the knowledge of $\Pr[x_{i-t} \oplus x'_i]$ for all $0 \leq i < m$ and some integer t. In the following, we provide the formulas for accomplishing this task.

Lemma 9. *Let $F : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a vectorial Boolean function. Assume that the input pair $(\mathbf{x}, \mathbf{x}') \in \mathbb{F}_2^m \times \mathbb{F}_2^m$ satisfies $\Pr[x_i \oplus x'_i = 1] = \Pr[x_i \neq x'_i] = p_i$ for $0 \leq i < m$, and the events $x_i \neq x'_i$ and $x_j \neq x'_j$ for different i and j are mutually independent. Then, for $\boldsymbol{\lambda} \in \mathbb{F}_2^n$, the differential-linear correlation of F can be computed as*

$$\begin{aligned}\mathcal{C}_{\boldsymbol{\lambda}}^{\mathrm{DL}} &= \Pr[\boldsymbol{\lambda} \cdot (F(\mathbf{x}) \oplus F(\mathbf{x}')) = 0] - \Pr[\boldsymbol{\lambda} \cdot (F(\mathbf{x}) \oplus F(\mathbf{x}')) = 1] \\ &= \sum_{\mathbf{u} \in \mathbb{F}_2^m} \frac{1}{2^m} \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\boldsymbol{\lambda} \cdot (F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \mathbf{u}))} \prod_{i=0}^{m-1} ((1-u_i) - (-1)^{u_i} p_i).\end{aligned}$$

Proof. Let $\mathcal{S}_{\mathbf{u}} = \{(\mathbf{x}, \mathbf{x}') \in \mathbb{F}_2^m \times \mathbb{F}_2^m : x \oplus x' = \mathbf{u}\}$ with $\#\mathcal{S}_{\mathbf{u}} = 2^m$. Then

$$\begin{aligned}\mathcal{C}_{\boldsymbol{\lambda}}^{\mathrm{DL}} &= \sum_{\mathbf{v} \in \boldsymbol{\lambda}^{\perp}} \Pr[F(\mathbf{x}) \oplus F(\mathbf{x}') = \mathbf{v}] - \sum_{\mathbf{v} \in \mathbb{F}_2^n \setminus \boldsymbol{\lambda}^{\perp}} \Pr[F(\mathbf{x}) \oplus F(\mathbf{x}') = \mathbf{v}] \\ &= \sum_{\mathbf{v} \in \mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda} \cdot \mathbf{v}} \Pr[F(\mathbf{x}) \oplus F(\mathbf{x}') = \mathbf{v}] \\ &= \sum_{\mathbf{v} \in \mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda} \cdot \mathbf{v}} \sum_{\mathbf{u} \in \mathbb{F}_2^m} \Pr[F(\mathbf{x}) \oplus F(\mathbf{x}') = \mathbf{v} | (\mathbf{x}, \mathbf{x}') \in \mathcal{S}_{\mathbf{u}}] \Pr[(\mathbf{x}, \mathbf{x}') \in \mathcal{S}_{\mathbf{u}}] \\ &= \sum_{\mathbf{u} \in \mathbb{F}_2^m} \sum_{\mathbf{v} \in \mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda} \cdot \mathbf{v}} \Pr[F(\mathbf{x}) \oplus F(\mathbf{x}') = \mathbf{v} | (\mathbf{x}, \mathbf{x}') \in \mathcal{S}_{\mathbf{u}}] \Pr[(\mathbf{x}, \mathbf{x}') \in \mathcal{S}_{\mathbf{u}}] \\ &= \sum_{\mathbf{u} \in \mathbb{F}_2^m} \frac{1}{2^m} \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\boldsymbol{\lambda} \cdot (F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \mathbf{u}))} \prod_{i=0}^{m-1} ((1-u_i) - (-1)^{u_i} p_i).\end{aligned}$$

□

Theorem 3. *Let $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{y}$, and $\mathbf{y}'$ be random n-bit strings such that $\Pr[x_i \oplus x'_i = 1] = p_i$ and $\Pr[y_i \oplus y'_i = 1] = q_i$ for $0 \leq i < n$. In addition, the events $x_i \oplus x'_i = 1$ and $y_j \oplus y'_j = 1$ for $0 \leq i, j < n$ are mutually independent. For $\boldsymbol{\lambda} \in \mathbb{F}_2^n$, the differential-linear correlation of $F(\mathbf{x}, \mathbf{y}) = \mathbf{x} \boxplus \mathbf{y}$ can be computed as*

$$\mathcal{C}_{\boldsymbol{\lambda}}^{\mathrm{DL}} = \frac{1}{2^{2n}} (1,1,1,1) \prod_{i=0}^{n-1} \mathbf{H}_{\lambda_i}^{p_i, q_i} (1,0,0,0)^T,$$

where $\mathbf{H}_{\lambda_i}^{p_i,q_i}$ *is a* 4×4 *matrix and is defined as*

$$\mathbf{H}_{\lambda_i}^{p_i,q_i} = \sum_{a,b\in\mathbb{F}_2} ((1-a)-(-1)^a p_i)((1-b)-(-1)^b q_i)\mathbf{M}_{a,b,\lambda_i}.$$

Proof. Let $\hat{p}_i(\alpha_i) = ((1-\alpha_i)-(-1)^{\alpha_i}p_i)$ and $\hat{q}_i(\beta_i) = ((1-\beta_i)-(-1)^{\beta_i}q_i)$. According to Lemma 9 and Theorem 1, $2^{2n}\mathcal{C}_{\boldsymbol{\lambda}}^{\mathrm{DL}}$ can be computed as

$$\begin{aligned}
&\sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathbb{F}_2^{2n}} \sum_{(\mathbf{x},\mathbf{y})\in\mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda}\cdot(S(\mathbf{x}\oplus\boldsymbol{\alpha},\mathbf{y}\oplus\boldsymbol{\beta})\oplus S(\mathbf{x},\mathbf{y}))} \prod_{i=0}^{n-1} \hat{p}_i(\alpha_i)\hat{q}_i(\beta_i) \\
&= \sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathbb{F}_2^{2n}} (1,1,1,1) \prod_{i=0}^{n-1} \mathbf{M}_{\alpha_i,\beta_i,\lambda_i} (1,0,0,0)^T \prod_{i=0}^{n-1} \hat{p}_i(\alpha_i)\hat{q}_i(\beta_i) \\
&= (1,1,1,1) \sum_{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathbb{F}_2^{2n}} \prod_{i=0}^{n-1} \hat{p}_i(\alpha_i)\hat{q}_i(\beta_i)\mathbf{M}_{\alpha_i,\beta_i,\lambda_i} (1,0,0,0)^T \\
&= (1,1,1,1) \prod_{i=0}^{n-1} \sum_{(a,b)\in\mathbb{F}_2^2} \hat{p}_i(a)\hat{q}_i(b)\mathbf{M}_{a,b,\lambda_i} (1,0,0,0)^T \\
&= (1,1,1,1) \prod_{i=0}^{n-1} \mathbf{H}_{\lambda_i}^{p_i,q_i} (1,0,0,0)^T.
\end{aligned}$$

□

Lemma 10. *Let* $F:\mathbb{F}_2^m\to\mathbb{F}_2^n$ *be a vectorial Boolean function and* $0\le t\le m-1$ *be an integer. Assume that the input pair* $(\mathbf{x},\mathbf{x}')\in\mathbb{F}_2^m\times\mathbb{F}_2^m$ *satisfies* $\Pr[x_{i-t}\oplus x'_i=1]=p_i$ *for* $0\le i<m$*, and the events* $x_{i-t}\neq x'_i$ *and* $x_{j-t}\neq x'_j$ *for different* i *and* j *are mutually independent. Then, for* $\boldsymbol{\lambda}\in\mathbb{F}_2^n$ *and rotation offset* t*, the rotational differential-linear correlation of* F *can be computed as*

$$\begin{aligned}
\mathcal{C}_{\boldsymbol{\lambda}}^{\text{R-DL}} &= \Pr[\boldsymbol{\lambda}\cdot(\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}'))=0] - \Pr[\boldsymbol{\lambda}\cdot(\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}'))=1] \\
&= \sum_{\mathbf{u}\in\mathbb{F}_2^m} \frac{1}{2^m} \sum_{\mathbf{x}\in\mathbb{F}_2^m} (-1)^{\boldsymbol{\lambda}\cdot(\overleftarrow{F}(\mathbf{x})\oplus F(\overleftarrow{\mathbf{x}}\oplus\mathbf{u}))} \prod_{i=0}^{m-1} ((1-u_i)-(-1)^{u_i}p_i).
\end{aligned}$$

Proof. Let $\mathcal{S}_{\mathbf{u}} = \{(\mathbf{x},\mathbf{x}')\in\mathbb{F}_2^m\times\mathbb{F}_2^m : \overleftarrow{\mathbf{x}}\oplus\mathbf{x}'=\mathbf{u}\}$ with $\#\mathcal{S}_{\mathbf{u}}=2^m$. Then,

$$\begin{aligned}
\mathcal{C}_{\boldsymbol{\lambda}}^{\text{R-DL}} &= \sum_{\mathbf{v}\in\boldsymbol{\lambda}^\perp} \Pr[\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}')=\mathbf{v}] - \sum_{\mathbf{v}\in\mathbb{F}_2^n\setminus\boldsymbol{\lambda}^\perp} \Pr[\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}')=\mathbf{v}] \\
&= \sum_{\mathbf{v}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot\mathbf{v}} \Pr[\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}')=\mathbf{v}] \\
&= \sum_{\mathbf{v}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot\mathbf{v}} \sum_{\mathbf{u}\in\mathbb{F}_2^m} \Pr[\overleftarrow{F}(\mathbf{x})\oplus F(\mathbf{x}')=\mathbf{v}|(\mathbf{x},\mathbf{x}')\in\mathcal{S}_{\mathbf{u}}]\Pr[(\mathbf{x},\mathbf{x}')\in\mathcal{S}_{\mathbf{u}}] \\
&= \sum_{\mathbf{u}\in\mathbb{F}_2^m}\sum_{\mathbf{v}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot\mathbf{v}} \Pr[F(\mathbf{x})\oplus F(\mathbf{x}')=\mathbf{v}|(\mathbf{x},\mathbf{x}')\in\mathcal{S}_{\mathbf{u}}]\Pr[(\mathbf{x},\mathbf{x}')\in\mathcal{S}_{\mathbf{u}}] \\
&= \sum_{\mathbf{u}\in\mathbb{F}_2^m} \frac{1}{2^m} \sum_{\mathbf{x}\in\mathbb{F}_2^m} (-1)^{\boldsymbol{\lambda}\cdot(\overleftarrow{F}(\mathbf{x})\oplus F(\overleftarrow{\mathbf{x}}\oplus\mathbf{u}))} \prod_{i=0}^{m-1} ((1-u_i)-(-1)^{u_i}p_i).
\end{aligned}$$

□

Theorem 4. *We use* $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{y}$, *and* $\mathbf{y}'$ *to represent random* n*-bit strings such that* $\Pr[x_{i-t} \oplus x'_i = 1] = p_i$ *and* $\Pr[y_{i-t} \oplus y'_i = 1] = q_i$ *for* $0 \le i < n$. *In addition, the events* $x_{i-t} \oplus x'_i = 1$ *and* $y_{j-t} \oplus y'_j = 1$ *for* $0 \le i, j < n$ *are mutually statistical independent. Let* $S(\mathbf{x}, \mathbf{y}) = \mathbf{x} \boxplus \mathbf{y}$ *and* $\mathbf{W}$ *be*

$$\prod_{i=0}^{t-1} \left(\sum_{(c,d)\in\mathbb{F}_2^2} \zeta(c,d,p_i,q_i)\mathbf{M}_{c,d,\lambda_i} \right) \begin{pmatrix} 1&1&0&0\\0&0&0&0\\0&0&1&1\\0&0&0&0 \end{pmatrix} \prod_{i=t}^{n-1} \left(\sum_{(a,b)\in\mathbb{F}_2^2} \zeta(a,b,p_i,q_i)\mathbf{M}_{a,b,\lambda_i} \right),$$

where $\zeta(a,b,p,q) = ((1-a)-(-1)^a p)((1-b)-(-1)^b q)$. *Then, for* $\boldsymbol{\lambda} \in \mathbb{F}_2^n$ *and rotation offset* t, *the rotational differential-linear correlation of* $S(\mathbf{x}, \mathbf{y})$ *can be computed as*

$$\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\lambda}} = (1,0,1,0)\mathbf{W}(1,0,0,0)^T + (0,1,0,1)\mathbf{W}(0,1,0,0)^T.$$

Proof. See Supplementary Material D in the extended version [39]. □

The above theorems only consider *standalone* modulo additions. In practice, we may tweak these theorems to make them suitable for use in specific applications. In what follows, we demonstrate a case where linear and nonlinear operations are considered as a whole. Note that in the remainder of this section, $\mathbf{x} \in \mathbb{F}_2^n$ and $F(\mathbf{x})$ for some vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ are regarded as *column* vectors. Therefore, a linear transformation of $\mathbf{y} \in \mathbb{F}_2^n$ can be written as $L\mathbf{y}$, where L is an $n \times n$ binary matrix.

Lemma 11. *Let* $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *be a vectorial Boolean function mapping* $\mathbf{x} \in \mathbb{F}_2^n$ *to* $L \circ S(\mathbf{x}) \oplus \mathbf{c}$, *where* $\mathbf{c} \in \mathbb{F}_2^n$ *is a constant,* $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ *is a nonlinear permutation, and* L *is an* $n \times n$ *binary matrix such that* $L(\mathbf{y} \lll t) = (L\mathbf{y}) \lll t$ *for all* $\mathbf{y} \in \mathbb{F}_2^n$ *and integer* t. *Then, the correlation of the rotational differential-linear approximation of* F *with rotation offset* t, *RX-difference* $\boldsymbol{\Delta}$, *and output linear mask* $\boldsymbol{\lambda} \in \mathbb{F}_2^n$ *can be computed as*

$$\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\Delta},\boldsymbol{\lambda}}(F) = (-1)^{\boldsymbol{\lambda}\cdot(\mathbf{c}\oplus\overleftarrow{\mathbf{c}})} \mathcal{C}^{\text{R-DL}}_{\boldsymbol{\Delta},L^T\boldsymbol{\lambda}}(S).$$

Proof. According to the definition of $\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\Delta},\boldsymbol{\lambda}}(F)$, we have

$$\begin{aligned}
\mathcal{C}^{\text{R-DL}}_{\boldsymbol{\Delta},\boldsymbol{\lambda}}(F) &= \frac{1}{2^{2n}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot(\overleftarrow{F}(\mathbf{x})\oplus F(\overleftarrow{\mathbf{x}}\oplus\boldsymbol{\Delta}))} \\
&= \frac{1}{2^{2n}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot(L(\overleftarrow{S}(\mathbf{x})\oplus S(\overleftarrow{\mathbf{x}}\oplus\boldsymbol{\Delta}))\oplus\overleftarrow{\mathbf{c}}\oplus\mathbf{c})} \\
&= \frac{1}{2^{2n}} \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{\boldsymbol{\lambda}\cdot(\overleftarrow{\mathbf{c}}\oplus\mathbf{c})\oplus\boldsymbol{\lambda}\cdot(L(\overleftarrow{S}(\mathbf{x})\oplus S(\overleftarrow{\mathbf{x}}\oplus\boldsymbol{\Delta})))} \\
&= \frac{1}{2^{2n}} (-1)^{\boldsymbol{\lambda}\cdot(\overleftarrow{\mathbf{c}}\oplus\mathbf{c})} \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{(L^T\boldsymbol{\lambda})\cdot(\overleftarrow{S}(\mathbf{x})\oplus S(\overleftarrow{\mathbf{x}}\oplus\boldsymbol{\Delta}))} \\
&= (-1)^{\boldsymbol{\lambda}\cdot(\mathbf{c}\oplus\overleftarrow{\mathbf{c}})} \mathcal{C}^{\text{R-DL}}_{\boldsymbol{\Delta},L^T\boldsymbol{\lambda}}(S).
\end{aligned}$$

□

In the analysis of `Alzette`, `SipHash`, `ChaCha`, and `SPECK`, we will instantiate the nonlinear permutation S in Lemma 11 with $S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$, while for `ChaCha`, we will consider $S(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y}, \mathbf{z}, \mathbf{w})$. Next, we only consider $S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$, and the generalization to the latter case is trivial.

Lemma 12. *The correlation of the differential-linear approximation of* $S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$ *with* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{F}_2^n$*, input difference* $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, and output linear mask* $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ *can be computed as*

$$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})}(S) = \frac{1}{2^{2n}}(1,1,1,1) \prod_{i=0}^{n-1} (-1)^{\gamma_i \beta_i} \mathbf{M}_{\alpha_i,\beta_i,\lambda_i} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Proof. Let $\mathbf{x}' = \mathbf{x} \oplus \boldsymbol{\alpha}$ and $\mathbf{y}' = \mathbf{y} \oplus \boldsymbol{\beta}$. Then,

$$\begin{aligned} \mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})}(S) &= \frac{1}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^{2n}} (-1)^{(\boldsymbol{\lambda},\boldsymbol{\gamma}) \cdot (S(\mathbf{x},\mathbf{y}) \oplus S(\mathbf{x}',\mathbf{y}'))} \\ &= \frac{(-1)^{\boldsymbol{\gamma} \cdot \boldsymbol{\beta}}}{2^{2n}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda} \cdot ((\mathbf{x} \boxplus \mathbf{y}) \oplus (\mathbf{x}' \boxplus \mathbf{y}'))}. \end{aligned}$$

Applying Theorem 1 to $\sum_{(\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^{2n}} (-1)^{\boldsymbol{\lambda} \cdot ((\mathbf{x} \boxplus \mathbf{y}) \oplus (\mathbf{x}' \boxplus \mathbf{y}'))}$ gives the proof. □

Similarly, based on Theorem 2, we can derive the following Lemma.

Lemma 13. *The correlation* $\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})}(S)$ *of the rotational differential-linear approximation of* $S(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$ *with rotational offset* t*, input RX-difference* $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, and output linear mask* $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ *can be computed as*

$$\big(1,0,1,0\big)\, \mathbf{C}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \big(0,1,0,1\big)\, \mathbf{C}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

where

$$\mathbf{C}_{(\boldsymbol{\alpha},\boldsymbol{\beta}),(\boldsymbol{\lambda},\boldsymbol{\gamma})} = 2^{-2n} \prod_{i=0}^{t-1} (-1)^{\gamma_i \beta_i} \mathbf{M}_{\alpha_i,\beta_i,\lambda_i} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \prod_{j=t}^{n-1} (-1)^{\gamma_j \beta_j} \mathbf{M}_{\alpha_j,\beta_j,\lambda_j}.$$

Lemma 12 and Lemma 13 lead to the following generalizations of Theorem 3 and Theorem 4.

Corollary 3. *Let* $\mathbf{x}$*,* $\mathbf{x}'$*,* $\mathbf{y}$*, and* $\mathbf{y}'$ *be random* n*-bit strings such that* $\Pr[x_i \oplus x'_i = 1] = p_i$ *and* $\Pr[y_i \oplus y'_i = 1] = q_i$ *for* $0 \leq i < n$*. In addition, the events* $x_i \oplus x'_i = 1$ *and* $y_j \oplus y'_j = 1$ *for* $0 \leq i, j < n$ *are mutually independent. For output linear*

mask $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, the differential-linear correlation of* $F(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$ *can be computed as*

$$\mathcal{C}^{\mathrm{DL}}_{(\boldsymbol{\lambda},\boldsymbol{\gamma})} = \frac{1}{2^{2n}}(1,1,1,1) \prod_{i=0}^{n-1} \mathbf{H}^{p_i,q_i}_{\lambda_i,\gamma_i}(1,0,0,0)^T,$$

where $\mathbf{H}^{p_i,q_i}_{\lambda_i,\gamma_i}$ *is a* 4×4 *matrix defined as*

$$\mathbf{H}^{p_i,q_i}_{\lambda_i,\gamma_i} = \sum_{a,b \in \mathbb{F}_2} (-1)^{\gamma_i b}((1-a) - (-1)^a p_i)((1-b) - (-1)^b q_i)\mathbf{M}_{a,b,\lambda_i}.$$

Corollary 4. *We use* $\mathbf{x}$*,* $\mathbf{x}'$*,* $\mathbf{y}$*, and* $\mathbf{y}'$ *to represent random* n*-bit strings such that* $\Pr[x_{i-t} \oplus x'_i = 1] = p_i$ *and* $\Pr[y_{i-t} \oplus y'_i = 1] = q_i$ *for* $0 \le i < n$*. In addition, the events* $x_{i-t} \oplus x'_i = 1$ *and* $y_{j-t} \oplus y'_j = 1$ *for* $0 \le i, j < n$ *are mutually statistical independent. Then, for output linear mask* $(\boldsymbol{\lambda}, \boldsymbol{\gamma}) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$*, rotational offset* t*, the rotational differential-linear correlation of* $F(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \boxplus \mathbf{y}, \mathbf{y})$ *can be computed as*

$$\mathcal{C}^{\text{R-DL}}_{(\boldsymbol{\lambda},\boldsymbol{\gamma})}(S) = (1,0,1,0)\mathbf{W}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + (0,1,0,1)\mathbf{W}\begin{pmatrix}0\\1\\0\\0\end{pmatrix},$$

where $\zeta(a,b,p,q) = ((1-a) - (-1)^a p)((1-b) - (-1)^b q)$*, and* $\mathbf{W}$ *is*

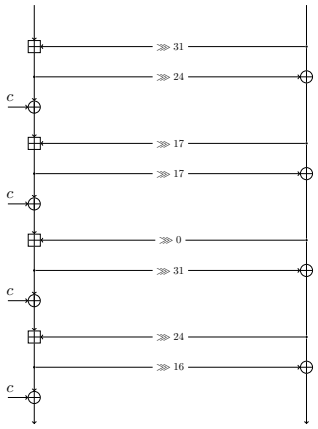
$$\prod_{i=0}^{t-1}\left(\sum_{(c,d)\in\mathbb{F}_2^2} (-1)^{\gamma_i d}\zeta(c,d,p_i,q_i)\mathbf{M}_{c,d,\lambda_i}\right)\begin{pmatrix}1&1&0&0\\0&0&0&0\\0&0&1&1\\0&0&0&0\end{pmatrix}\prod_{i=t}^{n-1}\left(\sum_{(a,b)\in\mathbb{F}_2^2} (-1)^{\gamma_i b}\zeta(a,b,p_i,q_i)\mathbf{M}_{a,b,\lambda_i}\right).$$

6 Applications to ARX Primitives

In this section, we apply the new technique for (rotational) differential-linear cryptanalysis proposed in Section 5 to the ARX primitives `Alzette`, `SipHash`, `SPECK`, and `ChaCha`. The source code for experimental verification is available at https://github.com/ZhongfengNiu/rot-differential-linear.

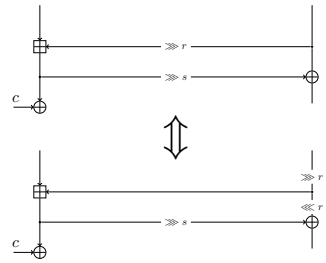
6.1 Cryptanalysis of `Alzette`

`Alzette` [7] is a 64-bit ARX-based S-box designed by Beierle et al., which is the main building block of the `Sparkle`-suite [8], a collection of lightweight symmetric cryptographic algorithms (AEADs and hash functions) currently in the final round of the NIST lightweight cryptography standardization effort. An instance of `Alzette` with an input $(x, y) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$ is depicted in Figure 1. To apply Corollary 3 and Corollary 4 developed in Section 5, we convert the `Alzette` round function into its equivalent form illustrated in Figure 2.



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

Fig. 1: The `Alzette` instance with $c =$ `B7E15162`



$$x \leftarrow x \boxplus (y \ggg r), y \leftarrow y \boxplus (x \ggg s), x \leftarrow x \oplus c$$

Fig. 2: An equivalent transformation of the round function of `Alzette`

Rotational differential-linear distinguishers. We use the same input RX-difference (`7fffffc`, `3fffffff`) employed in [32]. Then, we evaluate the correlations of the 4-round rotational differential-linear approximations of `Alzette` for all possible output linear masks with Hamming weight 2. The best distinguisher has a theoretical correlation of $2^{-5.57}$ (see the first row of Table 5), whose experimental correlation with 2^{26} random input pairs with the predefined input RX-difference is about $2^{-3.14}$.

Differential-linear distinguishers. According to Corollary 2, we choose (`80000000`, `00000000`) to be the input difference. Then, we evaluate the correlations of the differential-linear approximations of 4-, 5- and 6-round `Alzette` for all possible output linear masks with Hamming weight 2.

For 4-round `Alzette`, we identify a *deterministic* differential-linear approximation (see the second row of Table 5). For 5-, and 6-round `Alzette`, the best differential-linear distinguishers have a theoretical correlation of $-2^{-0.33}$ and $2^{-4.95}$, respectively. The experimental correlations given in Table 5 are obtained with 2^{26} random input pairs with the predefined input differences.

Table 5: Rotational differential-linear distinguishers for round-reduced `Alzette`, where the constants used are `0xB7E15162` and `0x38B4DA56`.

#Rnd	$\lll \gamma$	Input Difference	Output Mask	Correlation	
				Theory	Experiment
4	30	(7fffffc, 3fffffff)	(00004000, 40000000)	$2^{-5.57}$	$2^{-3.14}$
4	0	(80000000, 00000000)	(00080000, 00000008)	1	1
5	0	(80000000, 00000000)	(00000080, 00008000)	$-2^{-0.33}$	$-2^{-0.13}$
6	0	(80000000, 00000000)	(00000040, 00200000)	$2^{-4.95}$	$2^{-1.45}$
8	0	(80020100, 00010080)	(80000000, 00008000)	$-2^{-8.24}$	$-2^{-5.50}$

Extending the distinguisher. For 4-round `Alzette`, we can identify two optimal differential trails with a common input difference (80020100, 00010080) whose probabilities are 2^{-6}:

$$(80020100, 00010080) \to (01010000, 00030101),$$
$$(80020100, 00010080) \to (03010000, 00030301).$$

Moreover, we find two 4-round differential-linear approximations sharing a common output linear mask (80000000, 00008000) whose input differences are the two output differences of the above two differential trails respectively:

$$(01010000, 00030101) \to (80000000, 00008000),$$
$$(03010000, 00030301) \to (80000000, 00008000).$$

The theoretical correlations of the differential-linear approximations are $-2^{-2.90}$ and $-2^{-3.69}$, respectively. Combining the 4-round differential trials with the 4-round differential-linear approximations leads to an 8-round differential-linear distinguisher with theoretical correlation $2^{-6} \cdot (-2^{-2.90} - 2^{-3.69}) \approx -2^{-8.24}$ (see the last row of Table 5). The experimental correlation with 2^{26} random input pairs with the predefined input difference is $2^{-5.50}$.

6.2 Cryptanalysis of `SipHash`

`SipHash` [3] is a family of ARX-based pseudorandom functions optimized for short inputs. As mentioned in the introduction, instances of `SipHash` are widely deployed in real-world applications. The round function of `SipHash` is illustrated in Figure 3.

Aumasson and Bernstein proposed two specific instances for use, which are `SipHash`-2-4 and `SipHash`-4-8. Here we focus on the finalization process of `SipHash`-2-4, where four rounds are applied and the output branches are XORed together. In [20], Dobraunig, Mendel and Schläffer found a 4-round differential distinguisher for the finalization (see the last row of Table 6). According to Corollary 2, we choose 8000000000000000 to be the input difference of branch a. Then, we evaluate the correlations of all 3-round differential-linear approximations of `SipHash` for all possible output linear masks with Hamming weight 2. We

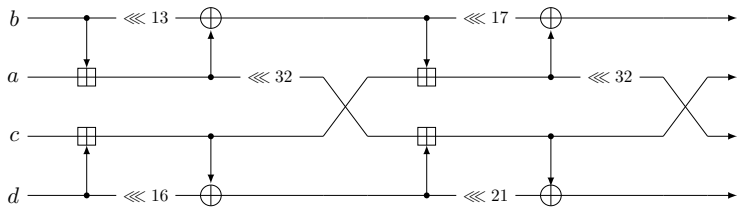


Fig. 3: The round function of `SipHash`

Table 6: Differential-linear distinguishers for the finalization of `SipHash`. Note that the 4-round distinguisher from [20] is a differential one.

#Rnd	Input Difference	Mask / Difference	Correlation / probability		Ref.
			Theory	Experiment	
3	0000000000000000 8000000000000000 0000000000000000 0000000000000000	0000400000004000	$2^{-2.19}$	$2^{-0.78}$	Sect. 6.2
4	0000000000040000 0000000080040000 0000000000000000 0000000000000000	2000000020000000	$2^{-12.45}$	$2^{-6.03}$	Sect. 6.2
4	0014002020010000 8010042000010000 0402200000000002 0402200000000000	2011421120010200	2^{-35}	–	[20]

find a 3-round distinguisher given in the first row of Table 6 with theoretical correlation $2^{-2.19}$ and experimental correlation $2^{-0.78}$.

If we choose `8000000000000000` to be the input difference of branch a and branch d, we find the following 3-round differential-linear distinguisher

$$\begin{matrix}0000000000000000\\8000000000000000\\0000000000000000\\8000000000000000\end{matrix} \rightarrow \quad 2000000020000000$$

with theoretical correlation $2^{-10.45}$. Extending this distinguisher backwards with the following differential

$$\begin{matrix}0000000000040000\\0000000080040000\\0000000000000000\\0000000000000000\end{matrix} \rightarrow \begin{matrix}0000000000000000\\8000000000000000\\0000000000000000\\8000000000000000\end{matrix}$$

with probability 2^{-2}, we obtain a 4-round differential-linear distinguisher for the finalization of SipHash-2-4 with theoretical correlation $2^{-12.45}$ and experimental correlation $2^{-6.03}$.

6.3 Cryptanalysis of SPECK

SPECK [6] is a family of ARX block ciphers designed by the U.S. National Security Agency (NSA). In this work, we focus on the version with a 32-bit block size, whose round function is depicted in Figure 4.

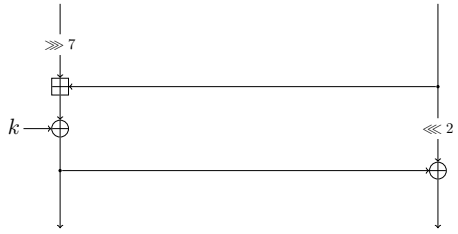


Fig. 4: The SPECK instance

As discussed in [32], it is difficult to apply rotational differential-linear attacks with nonzero rotation offset on keyed primitives due to the peculiarities of the RX-difference, and thus for SPECK we only consider ordinary differential-linear attacks (rotational differential-linear attacks with zero rotation offset). For SPECK32, we start with a 4-round differential trail:

$$(\texttt{0211}, \texttt{0a04}) \rightarrow (\texttt{8100}, \texttt{8102})$$

with probability 2^{-7}. Then, setting the input difference to $(\texttt{8100}, \texttt{8102})$, we evaluate the correlations of the differential-linear approximations of 4-round SPECK32 for all possible output linear masks with Hamming weight 2, and find one: $(\texttt{8100}, \texttt{8102}) \rightarrow (\texttt{0008}, \texttt{0008})$ with correlation $2^{-1.23}$. At this point, we obtain an 8-round differential-linear distinguisher for SPECK32 with theoretical correlation $2^{-8.23}$. By extending this distinguisher forward by a 1-round linear approximation $(\texttt{0008}, \texttt{0008}) \rightarrow (\texttt{5820}, \texttt{4020})$ with correlation 2^{-1} and backwards by a differential trail

$$(\texttt{0a20}, \texttt{4205}) \rightarrow (\texttt{0211}, \texttt{0a04})$$

with probability 2^{-5} we get a 10-round differential-linear distinguisher for SPECK32 with a theoretical correlation of $2^{-15.23}$, while previous best 10-round distinguisher for SPECK32 is a differential one with probability $2^{-31.01}$ [43] (too close to 2^{-32} to be valid in practice). Moreover, experimental results indicate that the actual correlation of our distinguisher is higher than expected. We random chose 100 master keys. For each key, we compute the experimental correlation of the distinguisher by going through the full plaintext space. The average correlation over the 100 keys is about $2^{-13.90}$.

Table 7: Differential-linear distinguishers for round-reduced SPECK32

#Rnd	$\lll \gamma$	Input Difference	Output Mask	Correlation	
				Theory	Experiment
8	0	(0211, 0a04)	(0008, 0008)	$2^{-8.23}$	$2^{-6.87}$
9	0	(0211, 0a04)	(5820, 4020)	$2^{-10.23}$	$2^{-8.93}$
10	0	(0a20, 4205)	(5820, 4020)	$2^{-15.23}$	$2^{-13.90}$

6.4 Cryptanalysis of ChaCha

As the default replacement for RC4 in the TLS protocol, ChaCha [10] is one of the most important ARX primitives. ChaCha operates on a 4×4 matrix of sixteen 32-bit words written as

$$\begin{pmatrix} \boldsymbol{x}_0 & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 \\ \boldsymbol{x}_4 & \boldsymbol{x}_5 & \boldsymbol{x}_6 & \boldsymbol{x}_7 \\ \boldsymbol{x}_8 & \boldsymbol{x}_9 & \boldsymbol{x}_{10} & \boldsymbol{x}_{11} \\ \boldsymbol{x}_{12} & \boldsymbol{x}_{13} & \boldsymbol{x}_{14} & \boldsymbol{x}_{15} \end{pmatrix}.$$

In each round, four parallel applications of a nonlinear transformation depicted in Figure 5 are performed on four 128-bit tuples formed by the words of the state matrix. Specifically, in odd-numbered rounds, the nonlinear transformation is applied to columns $(\boldsymbol{x}_0, \boldsymbol{x}_4, \boldsymbol{x}_8, \boldsymbol{x}_{12})$, $(\boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_9, \boldsymbol{x}_{13})$, $(\boldsymbol{x}_2, \boldsymbol{x}_6, \boldsymbol{x}_{10}, \boldsymbol{x}_{14})$, $(\boldsymbol{x}_3, \boldsymbol{x}_7, \boldsymbol{x}_{11}, \boldsymbol{x}_{15})$. In even-numbered rounds, the nonlinear operation is applied to $(\boldsymbol{x}_0, \boldsymbol{x}_5, \boldsymbol{x}_{10}, \boldsymbol{x}_{15})$, $(\boldsymbol{x}_1, \boldsymbol{x}_6, \boldsymbol{x}_{11}, \boldsymbol{x}_{12})$, $(\boldsymbol{x}_2, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{x}_{13})$, $(\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_9, \boldsymbol{x}_{14})$.

Rotational differential-linear distinguishers. In [45], Xu et al. presented a 1-round rotational differential-linear distinguisher for ChaCha (see the first row of Table 13 in Supplementary Material G in the extended version of this paper [39]) with an experimental correlation $2^{-0.01}$. The lower bound of the correlation of this differential-linear distinguisher is estimated to be 2^{-2} in [45]. We re-evaluate the correlation of this distinguisher with the method proposed in Section 5, and the obtained theoretical correlation is $2^{-0.01}$, perfectly matching the experimental result.

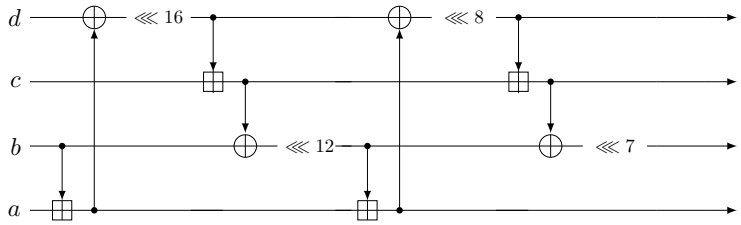


Fig. 5: The quartered round function of ChaCha

Differential-linear distinguishers. At CRYPTO 2020, Beierle et al. employed a series of 2.5-round differential-linear distinguishers (starting form even round) to perform key-recovery attacks on `ChaCha` [9] (see Table 13 in Supplementary Material G in the extended version of this paper [39]) whose experimental correlations are $2^{-8.3}$. With the method presented in Section 5, the predicted correlations are $2^{-12.14}$. Concerning this result, we would like to mention that the result obtained by Dey, Dey, Sarkar, and Meier [17] is better than ours. However, if the readers take a look at [17], it is easy to find that our method is more generic. Also, we evaluate the correlation of the 3-round differential-linear approximation used in FSE 2008 [4] and FSE 2016 [14] with an experimental correlation $2^{-5.25}$, and the obtained theoretical correlation is $2^{-9.88}$ (see the last row of Table 13 in Supplementary Material G in the extended version of this paper [39]).

Moreover, for 2.5-round `ChaCha`, we find a differential-linear distinguisher

$$\begin{matrix}\texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{20000000}\end{matrix} \rightarrow \begin{matrix}\texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000001}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\end{matrix}$$

whose theoretical correlation is $2^{-0.02}$. Since

$$\begin{cases} x_{10}^{2.5}[0] &= x_{10}^{3}[0] \oplus x_{14}^{3}[0] \\ x_{10}^{3}[0] &= x_{0}^{4}[0] \oplus x_{10}^{4}[0] \oplus x_{15}^{4}[0] \oplus x_{15}^{4}[8] \\ x_{14}^{3}[0] &= x_{3}^{4}[0] \oplus x_{3}^{4}[16] \oplus x_{4}^{4}[7] \oplus x_{9}^{4}[0] \oplus x_{14}^{4}[24] \end{cases},$$

extending the 2.5-round distinguisher gives a 4-round differential-linear distinguisher

$$\begin{matrix}\texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{00000000}&\texttt{20000000}\end{matrix} \rightarrow \begin{matrix}\texttt{00000001}&\texttt{00000000}&\texttt{00000000}&\texttt{00010001}\\ \texttt{00000080}&\texttt{00000000}&\texttt{00000000}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000001}&\texttt{00000001}&\texttt{00000000}\\ \texttt{00000000}&\texttt{00000000}&\texttt{01000000}&\texttt{00000101}\end{matrix}$$

with a theoretical correlation of $2^{-0.02}$ and experimental correlation of $2^{-0.98}$.

7 Conclusion, Discussion, and Open Problems

We present a method for evaluating the rotational differential-linear correlations of ARX ciphers for arbitrary output linear masks, partially solve the open problem proposed by Liu et al. at EUROCRYPT 2021. We apply the method to some ARX ciphers and obtain significantly improved results. Finally, we would like to give some open problems deserving further investigations.

Firstly, it seems that the formulas presented in this paper involving a chain of matrix multiplications cannot be translated into a compact finite automaton to be modeled with the MILP methodology. Therefore, we feel that the major pain spot of the current development is that there is no effective tool that can automatically search for good (rotational) differential-linear approximations, and

thus in practice the search space is severely limited to low Hamming weight output masks. Secondly, can we weaken or avoid the independence assumptions used in the method for evaluating the rotational differential-linear correlations? Remembering that we still have difficulties in explaining the experimental distinguishers listed in Supplementary Material H in the extended version of this paper [39], a solution to the independence problem may completely solve this problem.

Acknowledgment. We thank the reviewers for their valuable comments. This work is supported by the National Key Research and Development Program of China (2018YFA0704704), the Natural Science Foundation of China (62032014), and the Fundamental Research Funds for the Central Universities.

References

1. Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential cryptanalysis of round-reduced SIMON and SPECK. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption - 21st International Workshop, FSE 2014, March 3-5, 2014.*, volume 8540 of *Lecture Notes in Computer Science*, pages 525–545. Springer, 2014.
2. Tomer Ashur and Yunwen Liu. Rotational cryptanalysis in the presence of constants. *IACR Trans. Symmetric Cryptol.*, 2016(1):57–70, 2016.
3. Jean-Philippe Aumasson and Daniel J. Bernstein. SipHash: A fast short-input PRF. In *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, December 9-12, 2012. Proceedings*, pages 489–508, 2012.
4. Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier, and Christian Rechberger. New features of latin dances: Analysis of Salsa, ChaCha, and Rumba. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, February 10-13, 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 470–488. Springer, 2008.
5. Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST, 2010.
6. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK families of lightweight block ciphers. *IACR Cryptol. ePrint Arch.*, page 404, 2013.
7. 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 - (Feat. CRAX and TRAX). In *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, August 17-21, 2020, Proceedings, Part III*, pages 419–448, 2020.
8. 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 Trans. Symmetric Cryptol.*, 2020(S1):208–261, 2020.
9. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to ARX ciphers. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, August 17-21, 2020, Proceedings,*

Part III, volume 12172 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

10. Daniel J. Bernstein. Chacha, a variant of Salsa20. In *Workshop record of SASC*, volume 8, pages 3–5, 2008.
11. Daniel J. Bernstein. The Salsa20 family of stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.
12. Alex Biryukov and Vesselin Velichkov. Automatic search for differential trails in ARX ciphers. In *Topics in Cryptology - CT-RSA 2014, February 25-28, 2014. Proceedings*, volume 8366 of *Lecture Notes in Computer Science*, pages 227–250. Springer, 2014.
13. Alex Biryukov, Vesselin Velichkov, and Yann Le Corre. Automatic search for the best trails in ARX: application to block cipher SPECK. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, March 20-23, 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 289–310. Springer, 2016.
14. Arka Rai Choudhuri and Subhamoy Maitra. Significantly improved multi-bit differentials for reduced round Salsa and ChaCha. *IACR Trans. Symmetric Cryptol.*, 2016(2):261–287, 2016.
15. Murilo Coutinho and Tertuliano C. Souza Neto. Improved linear approximations to ARX ciphers and attacks against ChaCha. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 711–740. Springer, 2021.
16. David J. Wheeler and Roger M. Needham. TEA, a tiny encryption algorithm. In *International workshop on fast software encryption*, pages 363–366. Springer, 1994.
17. Sabyasachi Dey, Chandan Dey, Santanu Sarkar, and Willi Meier. Revisiting cryptanalysis on ChaCha from Crypto 2020 and Eurocrypt 2021. *IACR Cryptol. ePrint Arch.*, page 1059, 2021.
18. Daniel Dinu, Yann Le Corre, Dmitry Khovratovich, Léo Perrin, Johann Großschädl, and Alex Biryukov. Triathlon of lightweight block ciphers for the internet of things. *J. Cryptogr. Eng.*, 9(3):283–302, 2019.
19. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design strategies for ARX with provable bounds: SPARX and LAX. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 484–513, 2016.
20. Christoph Dobraunig, Florian Mendel, and Martin Schläffer. Differential cryptanalysis of SipHash. In *Selected Areas in Cryptography - SAC 2014 - 21st International Conference, August 14-15, 2014*, volume 8781 of *Lecture Notes in Computer Science*, pages 165–182. Springer, 2014.
21. Muhammad ElSheikh, Ahmed Abdelkhalek, and Amr M. Youssef. On MILP-based automatic search for differential trails through modular additions with application to Bel-T. In *Progress in Cryptology - AFRICACRYPT 2019 - 11th International Conference on Cryptology in Africa, July 9-11, 2019, Proceedings*, volume 11627 of *Lecture Notes in Computer Science*, pages 273–296. Springer, 2019.

22. Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. Submission to NIST, 2010.
23. Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. MILP-based automatic search algorithms for differential and linear trails for SPECK. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, March 20-23, 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 268–288. Springer, 2016.
24. Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Donggeon Lee. LEA: A 128-bit block cipher for fast encryption on common processors. In *Information Security Applications - 14th International Workshop, WISA 2013, August 19-21, 2013*, volume 8267 of *Lecture Notes in Computer Science*, pages 3–27. Springer, 2013.
25. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A new block cipher suitable for low-resource device. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, October 10-13, 2006, Proceedings*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.
26. Dongyeong Kim, Dawoon Kwon, and Junghwan Song. Efficient computation of boomerang connection probability for ARX-based block ciphers with application to SPECK and LEA. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 103-A(4):677–685, 2020.
27. Gaëtan Leurent. https://who.paris.inria.fr/Gaetan.Leurent/arxtools.html.
28. Gaëtan Leurent. Analysis of differential attacks in ARX constructions. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 226–243. Springer, 2012.
29. Gaëtan Leurent. Construction of differential characteristics in ARX designs application to Skein. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 241–258. Springer, 2013.
30. Gaëtan Leurent. Improved differential-linear cryptanalysis of 7-round chaskey with partitioning. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2016.
31. Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In *Fast Software Encryption, 8th International Workshop, FSE 2001, April 2-4, 2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 336–350. Springer, 2001.
32. Yunwen Liu, Siwei Sun, and Chao Li. Rotational cryptanalysis from a differential-linear perspective - practical distinguishers for round-reduced FRIET, Xoodoo, and Alzette. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 741–770. Springer, 2021.

33. Yunwen Liu, Glenn De Witte, Adrián Ranea, and Tomer Ashur. Rotational-XOR cryptanalysis of reduced-round SPECK. *IACR Trans. Symmetric Cryptol.*, 2017(3):24–36, 2017.
34. Pawel Morawiecki, Josef Pieprzyk, and Marian Srebrny. Rotational cryptanalysis of round-reduced Keccak. In Shiho Moriai, editor, *Fast Software Encryption 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 241–262. Springer, 2013.
35. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In Antoine Joux and Amr M. Youssef, editors, *Selected Areas in Cryptography - SAC 2014 - 21st International Conference, August 14-15, 2014*, volume 8781 of *Lecture Notes in Computer Science*, pages 306–323. Springer, 2014.
36. Nicky Mouha and Bart Preneel. Towards finding optimal differential characteristics for ARX: Application to Salsa20. Cryptology ePrint Archive, Report 2013/328, 2013. https://ia.cr/2013/328.
37. Nicky Mouha, Vesselin Velichkov, Christophe De Cannière, and Bart Preneel. The differential analysis of s-functions. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, August 12-13, 2010*, volume 6544 of *Lecture Notes in Computer Science*, pages 36–56. Springer, 2010.
38. National Institute of Standards and Technology. Preliminary state standard of republic of belarus (stbp 34.101.312011), 2011. http://apmi.bsu.by/assets/files/std/belt-spec27.pdf.
39. Zhongfeng Niu, Siwei Sun, Yunwen Liu, and Chao Li. Rotational differential-linear distinguishers of ARX ciphers with arbitrary output linear masks, 2022. https://eprint.iacr.org/2022/765.
40. Kaisa Nyberg and Johan Wallén. Improved linear distinguishers for SNOW 2.0. In *Fast Software Encryption, 13th International Workshop, FSE 2006, March 15-17, 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 144–162. Springer, 2006.
41. Roger M. Needham and David J. Wheeler. TEA extensions. *Report, Cambridge University*, 1997.
42. Akihiro Shimizu and Shoji Miyaguchi. Fast data encipherment algorithm FEAL. In *Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of of Cryptographic Techniques, April 13-15, 1987, Proceedings*, volume 304 of *Lecture Notes in Computer Science*, pages 267–278. Springer, 1987.
43. Ling Song, Zhangjie Huang, and Qianqian Yang. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, July 4-6, 2016, Proceedings, Part II*, volume 9723 of *Lecture Notes in Computer Science*, pages 379–394. Springer, 2016.
44. Johan Wallén. Linear approximations of addition modulo 2^n. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 261–273. Springer, 2003.
45. Yaqi Xu, Baofeng Wu, and Dongdai Lin. Rotational-linear attack: A new framework of cryptanalysis on ARX ciphers with applications to chaskey. In *Information and Communications Security - 23rd International Conference, ICICS 2021, November 19-21, 2021, Proceedings, Part II*, volume 12919 of *Lecture Notes in Computer Science*, pages 192–209. Springer, 2021.