Rotational Cryptanalysis From a Differential-linear Perspective

Practical Distinguishers for Round-reduced FRIET, Xoodoo, and Alzette

Yunwen Liu[1,2,3], Siwei Sun[2,3⋆], Chao Li[1]

[1] College of Liberal arts and Science, National University of Defense Technology, China univerlyw@hotmail.com
[2] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, China
siweisun.isaac@gmail.com

Abstract. The differential-linear attack, combining the power of the two most effective techniques for symmetric-key cryptanalysis, was proposed by Langford and Hellman at CRYPTO 1994. From the exact formula for evaluating the bias of a differential-linear distinguisher (JoC 2017), to the differential-linear connectivity table (DLCT) technique for dealing with the dependencies in the switch between the differential and linear parts (EUROCRYPT 2019), and to the improvements in the context of cryptanalysis of ARX primitives (CRYPTO 2020), we have seen significant development of the differential-linear attack during the last four years. In this work, we further extend this framework by replacing the differential part of the attack by rotational-xor differentials. Along the way, we establish the theoretical link between the rotational-xor differential and linear approximations, revealing that it is nontrivial to directly apply the closed formula for the bias of ordinary differential-linear attack to rotational differential-linear cryptanalysis. We then revisit the rotational cryptanalysis from the perspective of differential-linear cryptanalysis and generalize Morawiecki et al.'s technique for analyzing Keccak, which leads to a practical method for estimating the bias of a (rotational) differential-linear distinguisher in the special case where the output linear mask is a unit vector. Finally, we apply the rotational differential-linear technique to the permutations involved in FRIET, Xoodoo, Alzette, and SipHash. This gives significant improvements over existing cryptanalytic results, or offers explanations for previous experimental distinguishers without a theoretical foundation. To confirm the validity of our analysis, all distinguishers with practical complexities are verified experimentally.

Keywords: Differential-linear Cryptanalysis · Rotational Cryptanalysis · ARX · FRIET· Xoodoo· Alzette· SipHash

[⋆] Corresponding author

1 Introduction

The practical security of a symmetric-key primitive is determined by evaluating its resistance against an almost exhaustive list of known cryptanalytic techniques. Therefore, it is of essential importance to generalize existing cryptanalytic methods or develop new techniques. Sometimes the boundary between the two can be quite blurred. For example, the development of the invariant attacks [LAAZ11,LMR15,TLS19], ploytopic cryptanalysis [Tie16], division properties [Tod15,TM16], rotational cryptanalysis [KN10,AL16], etc. in recent years belongs to these two approaches.

Another approach is to employ known techniques in combination to enhance the effectiveness of the individual attacks. The boomerang [Wag99] and differential-linear cryptanalysis are the best examples. In particular, during the past four years, we have seen significant advancements in the development of the differential-linear cryptanalysis introduced by Langford and Hellman at CRYPTO 1994 [LH94], which combines the power of the two most important techniques (differential and linear attacks) for symmetric-key cryptanalysis. Our work starts with an attempt to further extend the differential-linear framework by replacing the differential part of this cryptanalytic technique with rotational-xor differentials.

Rotational and Rotational-xor Cryptanalysis. Rotational cryptanalysis was first formally introduced in [KN10] by Khovratovich and Nikolic, where the evolution of the so-called rotational pair $(x, x \lll t)$ through a target cipher was analyzed. The rotational properties of the building blocks of ARX primitives were then applied to the rotational rebound attack on the hash function Skein [KNR10], and later were refined to consider a chain of modular additions [KNP$^+$15]. Recently, cryptanalytic results of ARX-based permutations Chaskey and Chacha with respect to rotational cryptanalysis were reported [KAR20,BBM20]. Apart from the ARX constructions, permutations built with logical operations without modular additions, also known as AND-RX or LRX [AJN14] primitives, are particularly interesting with respect to rotational attacks. In 2010, Morawiecki *et al.* applied this technique to distinguish the round-reduced Keccak-f[1600] permutation by feeding in rotational pairs and observing the bias of the XOR of the $(i+t)$-th and i-th bits of the corresponding outputs, where t is the rotation offset and the addition should be taken modulo the size of the rotated word [MPS13]. We will come back to Morawiecki *et al.*'s technique and show that it has an intimate relationship with the so-called rotational differential-linear cryptanalysis we proposed in Section 3. To thwart rotational attacks, constants which are not rotation-invariant can be injected into the data path. Still, in certain cases, it is possible to overcome this countermeasure with some ad-hoc techniques.

Later, Ashur and Liu [AL16] generalized the concept of rotational pair by considering the propagation of a data pair (x, x') that is related by the so-called rotational-xor (RX) difference $(x \lll t) \oplus x' = \delta$. The cryptanalytic technique based on RX-difference was named as rotational-xor cryptanalysis. Note that when the RX-difference of the pair (x, x') is zero, it degenerates to a rotational pair. RX cryptanalysis integrates the effect of constants into the analysis and it has been successfully applied to many ARX or AND-RX designs [LWRA17,LLA$^+$20]. Hereafter, we refer both rotational and rotational-xor cryptanalysis as rotational cryptanalysis, or in a general sense, rotational cryptanalysis contains all the statistical attacks requiring chosen data (e.g., plaintexts) with certain rotational relationships.

Differential-linear Cryptanalysis. Given an encryption function E, we divide it into two consecutive subparts E_0 and E_1. Let $\delta \to \varDelta$ be a differential for E_0 with probability p, and $\varGamma \to \gamma$ be a linear approximation for E_1 with bias $\epsilon_{\varGamma,\gamma} = \Pr[\varGamma \cdot y \oplus \gamma \cdot E_1(y) = 0] - \frac{1}{2}$. Then, the overall bias $\mathcal{E}_{\delta,\gamma}$ of the differential-linear distinguisher can be estimated with the piling-up lemma [Mat93] as

$$\mathcal{E}_{\delta,\gamma} = \Pr[\gamma \cdot (E(x) \oplus E(x \oplus \delta)) = 0] - \frac{1}{2} = (-1)^{\varGamma \cdot \varDelta} \cdot 2p\epsilon^2_{\varGamma,\gamma}, \tag{1}$$

since $\gamma \cdot (E(x) \oplus E(x \oplus \delta))$ can be decomposed into the XOR sum of the following three terms

$$\begin{cases} \varGamma \cdot (E_0(x) \oplus E_0(x \oplus \delta)), \\ \varGamma \cdot E_0(x \oplus \delta) \oplus \gamma \cdot E(x \oplus \delta), \\ \varGamma \cdot E_0(x) \oplus \gamma \cdot E(x). \end{cases}$$

The derivation of Equation (1) not only relies on the independence of E_0 and E_1, but also the assumption

$$\Pr[\varGamma \cdot (E_0(x) \oplus E_0(x \oplus \delta)) = 0 \mid E_0(x) \oplus E_0(x \oplus \delta) \neq \varDelta] = \frac{1}{2}, \tag{2}$$

under which we have $\Pr[\varGamma \cdot (E_0(x) \oplus E_0(x \oplus \delta)) = 0] = \frac{1}{2} + \frac{(-1)^{\varGamma \cdot \varDelta}}{2} p$.

However, it has long been observed that Equation (2) may fail in many cases and multiple linear approximations have to be taken into account to make the estimates more accurate [LH94,LGZL09,Lu15]. In [BLN17], Blondeau, Leander, and Nyberg presented a closed formula for the overall bias $\mathcal{E}_{\delta,\gamma}$ based on the link between differential and linear attacks [CV94] under the sole assumption that E_0 and E_1 are independent. However, this closed formula is generally not applicable in practice even if E_0 and E_1 are independent, since it requires the computation of the exact bias $\epsilon_{\delta,v} = \Pr[v \cdot (E_0(x) \oplus E_0(x \oplus \delta)) = 0] - \frac{1}{2}$ for all v. [1] Moreover, in some cases the dependency between E_0 and E_1 can be significant. Inspired by the boomerang-connectivity table (BCT) and its successful applications in the context of boomerang attacks [CHP$^+$18], Bar-On, Dunkelman, Keller, and Weizman introduced the differential-linear connectivity table (DLCT) [BDKW19], where the target cipher is decomposed as $E = E_1 \circ E_m \circ E_0$ and the actual differential-linear probability of the middle part E_m is determined by experiments, fully

[1] Unlike the estimation of the probability of a differential with a large number of characteristics, a partial evaluation of the differential-linear distinguisher without the full enumeration of intermediate masks can be inaccurate, since both positive and negative biases occur.

addressing the issue of dependency in the switch between E_0 and E_1 (The effect of multiple characteristics and approximations still has to be handled by the framework of Blondeau *et al.* [BLN17]). Most recently, Beierle, Leander, and Todo presented several improvements to the framework of differential-linear attacks with a special focus on ARX ciphers at CRYPTO 2020 [BLT20].

Our Contribution. We start from the natural idea to extend the framework of differential-linear attacks by replacing the differential part with rotational-xor differentials. Specifically, given a pair of data with RX-difference $\delta = (x \lll t) \oplus x'$ and a linear mask γ, a *rotational differential-linear* distinguisher of a cipher E exploits the bias of $\gamma \cdot (\texttt{rot}(E(x)) \oplus E(\texttt{rot}(x) \oplus \delta))$, where $\texttt{rot}(\cdot)$ is some rotation-like operation.

We then present an informal formula similar to Equation (1) to estimate the bias of a rotational differential-linear distinguisher by the probability of the rotational-xor differential covering E_0 and the biases of the linear approximation and its rotated version covering E_1, where $E = E_1 \circ E_0$. This formula, as in the case of ordinary differential-linear cryptanalysis, requires certain assumptions that may not hold in practice.

Consequently, we try to derive a closed formula for computing the bias of a rotational differential-linear distinguisher, which we expect to be analogous to Blondeau *et al.*'s result [BLN17]. Although we failed to achieve this goal, we manage to establish a general link between the rotational-xor cryptanalysis and linear cryptanalysis as a by-product of this failed endeavour. From a practical point of view, we do not lose much due to the absence of a closed formula, since this kind of formula will inevitably involve the correlations of exponentially many trails which are hard to evaluate in most situations.

Then, we focus our attention on the special case of rotational differential-linear cryptanalysis where the output linear mask γ is a unit vector. In this case, the bias $\Pr[e_i \cdot (\texttt{rot}(f(x)) \oplus f(\texttt{rot}(x) \oplus \delta)) = 0] - \frac{1}{2}$ is

$$\Pr[(E(x))_j \oplus (E(x'))_i = 0] - \frac{1}{2} = \frac{1}{2} - \Pr[(E(x))_j \neq (E(x'))_i], \tag{3}$$

for some i and j, where $x' = \texttt{rot}(x) \oplus \delta$. With this formulation, we immediately realize that Morawiecki *et al.*'s approach [MPS13] gives rise to an efficient method for evaluating the biases of rotational differential-linear distinguishers, as well as ordinary differential-linear distinguishers whose output linear masks are unit vectors. We generalize some results from Morawiecki *et al.*'s work and arrive at formulas which are able to predict $\Pr[(f(x))_j \neq f(x')_i]$ based on the information $\Pr[x_j \neq x_i]$ for many common operations f appearing in ARX designs. In particular, we give the explicit formula for computing the differential-linear and rotational differential-linear probability for an n-bit modular addition with $O(n)$ operations, while a direct application of Bar-On *et al.*'s approach [BDKW19] based on the Fast Fourier Transformation (FFT) by treating the modular addition as an $2n \times n$ S-box would require a complexity of $\mathcal{O}(2^{2n})$. The probability evaluation can be iteratively applied for an ARX or AND-RX construction. Nevertheless, we note that the accuracy of the probability evaluation is affected by the dependency among the neighbour bits.

Finally, we apply the technique of rotational differential-linear cryptanalysis to the cryptographic permutations involved in FRIET, Xoodoo and Alzette. For FRIET, we find a 6-round rotational differential-linear distinguisher with a correlation $2^{-5.81}$, and it can be extended to a practical 8-round rotational differential-linear distinguisher with a correlation of $2^{-17.81}$. As a comparison, the correlation of the best known 8-round linear trail of FRIET is 2^{-40}. Moreover, our 6-round distinguisher for FRIET can be further extended to a 13-round one. For Xoodoo, we identify a 4-round rotational differential-linear distinguisher with a correlation 1, while previous best result for Xoodoo is a 3-round differential with a probability 2^{-36}. For Alzette, the 64-bit ARX-box, we find a 4-round differential-linear distinguisher with a correlation $2^{-0.27}$ and a 4-round rotational differential-linear distinguisher with a correlation $2^{-11.37}$. A summary of the results is shown in Table 1, where all distinguishers with practical complexities are experimentally verified.

Table 1: A summary of the results. R-DL = rotational differential-linear, DL = differential-linear, LC = linear characteristic, DC = differential characteristic. We show differentials with probabilities and LC/DL/R-DL with correlations.

Permutation	Type	# Round	Probability/Correlation		Ref.
			Theoretical	Experimental	
FRIET	R-DL	6	$2^{-5.81}$	$2^{-5.12}$	Sect. 5
	R-DL	7	$2^{-9.81}$	$2^{-9.12}$	Sect. 5
	LC	7	2^{-29}	–	[SBD+20]
	R-DL	8	$2^{-17.81}$	$2^{-17.2}$	Sect. 5
	LC	8	2^{-40}	–	[SBD+20]
	R-DL	13	$2^{-117.81}$	–	Sect. 5
Xoodoo	DC	3	2^{-36}	–	[DHAK18]
	R-DL	4	1	1	Sect. 5
Alzette	DC	4	2^{-6}	–	[BBdS+20]
	R-DL	4	$2^{-11.37}$	$2^{-7.35}$	Sect. 6
	DL	4	$2^{-0.27}$	$2^{-0.1}$	Sect. 6

Outline. Section 2 introduces the notations and preliminaries for rotational-xor and linear cryptanalysis. We propose the rotational differential-linear cryptanalysis and establish the theoretical link between the rotational-xor cryptanalysis and linear cryptanalysis in Section 3. This is followed by Section 4 where we explore the methods for evaluating the biases of rotational differential-linear distinguishers. In Section 5 and Section 6, we apply the techniques developed in previous sections to AND-RX and ARX primitives. Section 7 concludes the paper with some open problems.

2 Notations and Preliminaries

Let $\mathbb{F}_2 = \{0, 1\}$ be the field with two elements. We denote by x_i the i-th bit of a bit string $x \in \mathbb{F}_2^n$. For a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ with $y = F(x) \in \mathbb{F}_2^m$, its i-th output bit y_i is denoted by $(F(x))_i$. For an n-bit string x, we use the indexing scheme $x = (x_{n-1}, \cdots, x_1, x_0)$. In addition, concrete values in $\mathbb{F}_2^n$ are specified in hexadecimal notations. For example, we use `1111` to denote the binary string $(\texttt{0001 0001 0001 0001})_2$.

The XOR-difference and rotational-xor difference with offset t of two bit strings x and x' in $\mathbb{F}_2^n$ are defined as $x \oplus x'$ and $(x \lll t) \oplus x'$, respectively. For the rotational-xor difference $\delta = (x \lll t) \oplus x'$, we may omit the rotation offset and write $\delta = \overleftarrow{x} \oplus x'$ or $\delta = \texttt{rot}(x) \oplus x'$ to make the notation more compact when it is clear from the context. Moreover, by abusing the notation, $\overleftarrow{x}$ and $\texttt{rot}(x)$ may rotate the entire string x or rotate the substrings of x to the left separately with a common offset, depending on the context. For instance, in the analysis of `Keccak`-f, we rotate each lane of the state by certain amount [MPS13]. Correspondingly, $\overrightarrow{x}$ and $\texttt{rot}^{-1}(x)$ rotate x or its substrings to the right. Similar to differential cryptanalysis with XOR-difference, we can define the probability of an RX-differential as follows.

Definition 1 (RX-differential probability). *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial boolean function. Let α and β be n-bit words. Then, the RX-differential probability of the RX-differential $\alpha \to \beta$ for f is defined as*

$$\Pr[\alpha \to \beta] = 2^{-n}\#\{x \in \mathbb{F}_2^n : \texttt{rot}(f(x)) \oplus f(\texttt{rot}(x) \oplus \alpha) = \beta\}$$

Finally, the definitions of correlation, bias, and some lemmas concerning Boolean functions together with the piling-up lemma are needed.

Definition 2 ([Car06,Can16]). *The correlation of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as $\mathrm{cor}(f) = 2^{-n}(\#\{x \in \mathbb{F}_2^n : f(x) = 0\} - \#\{x \in \mathbb{F}_2^n : f(x) = 1\})$.*

Definition 3 ([Car06,Can16]). *The bias $\epsilon(f)$ of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as $2^{-n}\#\{x \in \mathbb{F}_2^n : f(x) = 0\} - \frac{1}{2}$.*

From Definition 2 and Definition 3 we can see that $\mathrm{cor}(f) = 2\epsilon(f)$.

Definition 4. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function. The Walsh-Hadamard transformation takes in f and produces a real-valued function $\hat{f} : \mathbb{F}_2^n \to \mathbb{R}$ such that*

$$\forall w \in \mathbb{F}_2^n, \quad \hat{f}(w) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot w}.$$

Definition 5. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and $g : \mathbb{F}_2^n \to \mathbb{F}_2$ be two Boolean functions. The convolutional product of f and g is a Boolean function defined as*

$$\forall y \in \mathbb{F}_2^n, \quad (f \star g)(y) = \sum_{x \in \mathbb{F}_2^n} g(x) f(x \oplus y).$$

Lemma 1 ([Car06], Corollary 2). *Let $\hat{f}$ be the Walsh-Hadamard transformation of f. Then the Walsh-Hadamard transformation of $\hat{f}$ is $2^n f$.*

Lemma 2 ([Car06], Proposition 6). $\widehat{(f \star g)}(z) = \hat{f}(z)\hat{g}(z)$ *and thus* $\widehat{(f \star f)} = (\hat{f})^2$.

Lemma 3 (Piling-up Lemma [Mat93]). *Let $Z_0, \cdots, Z_{m-1}$ be m independent binary random variables with $\Pr[Z_i = 0] = p_i$. Then we have that*

$$\Pr[Z_0 \oplus \cdots \oplus Z_{m-1} = 0] = \frac{1}{2} + 2^{m-1} \prod_{i=0}^{m-1} (p_i - \frac{1}{2}),$$

or alternatively, $2\Pr[Z_0 \oplus \cdots \oplus Z_{m-1} = 0] - 1 = \prod_{i=0}^{m-1}(2p_i - 1)$.

3 Rotational Differential-linear cryptanalysis

A natural extension of the differential-linear cryptanalysis is to replace the differential part of the attack by rotational-xor (RX) differentials. Let $E = E_1 \circ E_0$ be an encryption function. Assume that we have an RX-differential $\delta \to \Delta$ covering E_0 with $\Pr[\mathtt{rot}(E_0(x)) \oplus E_0(\mathtt{rot}(x) \oplus \delta) = \Delta] = p$ and a linear approximation $\Gamma \to \gamma$ of E_1 such that

$$\begin{cases} \epsilon_{\Gamma,\gamma} = \Pr[\Gamma \cdot y \oplus \gamma \cdot E_1(y) = 0] - \frac{1}{2}, \\ \epsilon_{\mathtt{rot}^{-1}(\Gamma),\mathtt{rot}^{-1}(\gamma)} = \Pr[\mathtt{rot}^{-1}(\Gamma) \cdot y \oplus \mathtt{rot}^{-1}(\gamma) \cdot E_1(y) = 0] - \frac{1}{2}. \end{cases}$$

Let $x' = \mathtt{rot}(x) \oplus \delta$. If the assumption

$$\Pr[\Gamma \cdot (\mathtt{rot}(E_0(x)) \oplus E_0(x')) = 0 \mid \mathtt{rot}(E_0(x)) \oplus E_0(x') \neq \Delta] = \frac{1}{2} \tag{4}$$

holds. We have

$$\Pr[\Gamma \cdot (\mathtt{rot}(E_0(x)) \oplus E_0(x')) = 0] = \frac{1}{2} + \frac{(-1)^{\Gamma \cdot \Delta}}{2} p.$$

Since

$$\begin{aligned} \gamma \cdot (\mathtt{rot}(E(x)) \oplus E(x')) &= \gamma \cdot \mathtt{rot}(E(x)) \oplus \Gamma \cdot \mathtt{rot}(E_0(x)) \\ &\oplus \Gamma \cdot (\mathtt{rot}(E_0(x)) \oplus E_0(x')) \\ &\oplus \Gamma \cdot E_0(x') \oplus \gamma \cdot E(x') \\ &= \mathtt{rot}(\mathtt{rot}^{-1}(\gamma) \cdot E(x) \oplus \mathtt{rot}^{-1}(\Gamma) \cdot E_0(x)) \\ &\oplus \Gamma \cdot (\mathtt{rot}(E_0(x)) \oplus E_0(x')) \\ &\oplus \Gamma \cdot E_0(x') \oplus \gamma \cdot E(x'), \end{aligned}$$

the bias of the rotational differential-linear distinguisher can be estimated by piling-up lemma as

$$\mathcal{E}^{\text{R-DL}}_{\delta,\gamma} = \Pr[\gamma \cdot (\overleftarrow{E}(x) \oplus E(x')) = 0] - \frac{1}{2} = (-1)^{\mathit{\Gamma}\cdot\mathit{\Delta}} \cdot 2p\epsilon_{\mathit{\Gamma},\gamma}\epsilon_{\mathtt{rot}^{-1}(\mathit{\Gamma}),\mathtt{rot}^{-1}(\gamma)}, \quad (5)$$

and the corresponding correlation of the distinguisher is

$$\mathcal{C}^{\text{R-DL}}_{\delta,\gamma} = 2\mathcal{E}^{\text{R-DL}}_{\delta,\gamma} = (-1)^{\mathit{\Gamma}\cdot\mathit{\Delta}} \cdot 4p\epsilon_{\mathit{\Gamma},\gamma}\epsilon_{\mathtt{rot}^{-1}(\mathit{\Gamma}),\mathtt{rot}^{-1}(\gamma)}. \quad (6)$$

We can distinguish E from random permutations if the absolute value of $\mathcal{E}^{\mathrm{R-DL}}_{\delta,\gamma}$ or $\mathcal{C}^{\text{R-DL}}_{\delta,\gamma}$ is sufficiently high. Note that if we set the rotation offset to zero, the rotational differential-linear attack is exactly the ordinary differential-linear cryptanalysis. Therefore, the rotational differential-linear attack is a strict generalization of the ordinary differential-linear cryptanalysis.

A rotational differential-linear distinguisher can be extended by appending linear approximations at the end. Given a rotational differential-linear distinguisher of a function f with a bias

$$\epsilon_{\delta,\gamma} = \Pr[\gamma \cdot (\mathtt{rot}(f(x)) \oplus f(\mathtt{rot}(x) \oplus \delta)) = 0] - \frac{1}{2},$$

and a linear approximation (γ, μ) over a function g with

$$\begin{cases} \epsilon_{\gamma,\mu} = \Pr[\gamma \cdot x \oplus \mu \cdot g(x) = 0] - \frac{1}{2}, \\ \epsilon_{\mathtt{rot}^{-1}(\gamma),\mathtt{rot}^{-1}(\mu)} = \Pr[\mathtt{rot}^{-1}(\gamma) \cdot x \oplus \mathtt{rot}^{-1}(\mu) \cdot g(x) = 0] - \frac{1}{2}, \end{cases}$$

we can compute the bias of the rotational differential-linear distinguisher of $h = g \circ f$ with input RX-difference δ and output linear mask μ by the piling-up lemma. Since

$$\begin{aligned} \mu \cdot (\mathtt{rot}(h(x)) \oplus h(\mathtt{rot}(x) \oplus \delta)) = {} & \gamma \cdot (\mathtt{rot}(f(x)) \oplus f(\mathtt{rot}(x) \oplus \delta)) \\ & \oplus \gamma \cdot \mathtt{rot}(f(x)) \oplus \mu \cdot \mathtt{rot}(h(x)) \\ & \oplus \gamma \cdot f(\mathtt{rot}(x) \oplus \delta) \oplus \mu \cdot h(\mathtt{rot}(x) \oplus \delta) \end{aligned} ,$$

the bias of the rotational differential-linear distinguisher can be estimated as

$$\Pr[\mu \cdot (\mathtt{rot}(h(x)) \oplus h(\mathtt{rot}(x) \oplus \delta)) = 0] - \frac{1}{2} = 4\epsilon_{\delta,\gamma}\epsilon_{\gamma,\mu}\epsilon_{\mathtt{rot}^{-1}(\gamma),\mathtt{rot}^{-1}(\mu)}. \quad (7)$$

However, as in ordinary differential-linear attacks, the assumption described by Equation (4) may not hold in practice, and we prefer a closed formula for the bias $\mathcal{E}^{\text{R-DL}}_{\delta,\gamma}$ without this assumption for much the same reasons leading to Blondeau *et al.*'s work [BLN17]. Also, we would like to emphasize that if Equation (5) and (7) are used to estimate the bias, we should verify the results experimentally whenever possible.

3.1 Towards a Closed Formula for The Bias of the Rotational Differential-linear Distinguisher

In [BLN17], Blondeau *et al.* proved the following theorem based on the general link between differential and linear cryptanalysis [CV94].

Theorem 1 ([BLN17]). *If E_0 and E_1 are independent, the bias of a differential-linear distinguisher with input difference δ and output linear mask γ can be computed as*

$$\mathcal{E}_{\delta,\gamma} = \sum_{v \in \mathbb{F}_2^n} \epsilon_{\delta,v} c_{v,\gamma}^2, \tag{8}$$

for all $\delta \neq 0$ and $\gamma \neq 0$, where

$$\begin{cases} \epsilon_{\delta,v} = \Pr[v \cdot (E_0(x) \oplus E_0(x \oplus \delta)) = 0] - \frac{1}{2} \\ c_{v,\gamma} = \mathrm{cor}(v \cdot y \oplus \gamma \cdot E_1(y)) \end{cases}.$$

To replay Blondeau *et al.*'s technique in an attempt to derive the rotational differential-linear counterpart of Equation (8), we have to first establish the relationship between rotational differential-linear cryptanalysis and linear cryptanalysis.

Link between RX-cryptanalysis and linear cryptanalysis. Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial Boolean function. The cardinality of the set

$$\{x \in \mathbb{F}_2^n : \overleftarrow{F}(x) \oplus F(\overleftarrow{x} \oplus a) = b\}$$

is denoted by $\xi_F(a,b)$, and the correlation of $u \cdot x \oplus v \cdot F(x)$ is $\mathrm{cor}(u \cdot x \oplus v \cdot F(x))$. Let $\underrightarrow{\overleftarrow{F}} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the vectorial Boolean function mapping x to $\overleftarrow{F}(\overrightarrow{x})$. It is easy to show that

$$\mathrm{cor}(u \cdot x \oplus v \cdot \underrightarrow{\overleftarrow{F}}(x)) = \mathrm{cor}(\overrightarrow{u} \cdot x \oplus \overrightarrow{v} \cdot F(x)).$$

In what follows, we are going to establish the relationship between

$$\xi_F(a,b), \quad \mathrm{cor}(u \cdot x \oplus v \cdot F(x)), \quad \text{and} \quad \mathrm{cor}(\overrightarrow{u} \cdot x \oplus \overrightarrow{v} \cdot F(x)).$$

Definition 6. *Given a vectorial Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, the Boolean function $\theta_F : \mathbb{F}_2^{2n} \to \mathbb{F}_2$ is defined as*

$$\theta_F(x,y) = \begin{cases} 1 & \text{if} \quad y = F(x), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Lemma 4. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial Boolean function. Then for any $(a,b) \in \mathbb{F}_2^{2n}$, we have $\xi_F(a,b) = (\theta_{\underrightarrow{\overleftarrow{F}}} \star \theta_F)(a,b)$.*

Proof. According to Definition 5, we have

$$\begin{aligned}(\theta_{\overleftarrow{\underrightarrow{F}}} \star \theta_F)(a,b) &= \sum_{x||y\in\mathbb{F}_2^{2n}} \theta_{\overleftarrow{\underrightarrow{F}}}(x,y)\theta_F(a\oplus x, b\oplus y)\\ &= \sum_{x\in\mathbb{F}_2^n}\sum_{y\in\mathbb{F}_2^n} \theta_{\overleftarrow{\underrightarrow{F}}}(x,y)\theta_F(a\oplus x, b\oplus y)\\ &= \sum_{x\in\mathbb{F}_2^n} \theta_{\overleftarrow{\underrightarrow{F}}}(x,\overleftarrow{\underrightarrow{F}}(x))\theta_F(a\oplus x, b\oplus \overleftarrow{\underrightarrow{F}}(x))\\ &= \sum_{x\in\mathbb{F}_2^n} \theta_F(a\oplus x, b\oplus \overleftarrow{\underrightarrow{F}}(x))\\ &= \#\{x\in\mathbb{F}_2^n : b\oplus \overleftarrow{\underrightarrow{F}}(x) = F(a\oplus x)\}\\ &= \xi_F(a,b).\end{aligned}$$

□

Lemma 5. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a vectorial Boolean function. Then for any $(a,b)\in\mathbb{F}_2^{2n}$, we have* $\mathrm{cor}(a\cdot x\oplus b\cdot F(x)) = 2^{-n}\hat{\theta}_F(a,b)$.

Proof. According to Definition 4, we have

$$\begin{aligned}\hat{\theta}_F(a,b) &= \sum_{x||y\in\mathbb{F}_2^{2n}} \theta_F(x,y)(-1)^{(x||y)\cdot(a||b)}\\ &= \sum_{x\in\mathbb{F}_2^n}\sum_{y\in\mathbb{F}_2^n} \theta_F(x,y)(-1)^{a\cdot x\oplus b\cdot y}\\ &= \sum_{x\in\mathbb{F}_2^n} (-1)^{a\cdot x\oplus b\cdot F(x)}\\ &= 2^n\mathrm{cor}(a\cdot x\oplus b\cdot F(x)).\end{aligned}$$

□

In addition, applying Lemma 5 to $\overleftarrow{\underrightarrow{F}}$ gives $\mathrm{cor}(a\cdot x\oplus b\cdot \overleftarrow{\underrightarrow{F}}(x)) = \frac{1}{2^n}\hat{\theta}_{\overleftarrow{\underrightarrow{F}}}(a,b)$.

Theorem 2. *The link between RX-differentials and linear approximations can be summarized as*

$$\xi_F(a,b) = \sum_{u\in\mathbb{F}_2^n}\sum_{v\in\mathbb{F}_2^n} (-1)^{u\cdot a\oplus v\cdot b}\mathrm{cor}(\overrightarrow{u}\cdot x\oplus \overrightarrow{v}\cdot F(x))\mathrm{cor}(u\cdot x\oplus v\cdot F(x)). \tag{10}$$

Proof. According to Lemma 4 and Lemma 2, we have

$$2^{2n}\xi_F(a,b) = (\widehat{\theta_{\overleftarrow{\underrightarrow{F}}}\star\theta_F})(a,b) = \widehat{\hat{\theta}_{\overleftarrow{\underrightarrow{F}}}\hat{\theta}_F}(a,b).$$

Since $\hat{\theta}_{\underset{\rightarrow}{\overleftarrow{F}}}\hat{\theta}_F = 2^{2n}\mathrm{cor}(u \cdot x \oplus v \cdot \underset{\rightarrow}{\overleftarrow{F}}(x))\mathrm{cor}(u \cdot x \oplus v \cdot F(x))$ due to Lemma 5,

$$\begin{aligned}
\widehat{\hat{\theta}_{\underset{\rightarrow}{\overleftarrow{F}}}\hat{\theta}_F}(a,b) &= 2^{2n} \sum_{u||v \in \mathbb{F}_2^{2n}} (-1)^{(u||v)\cdot(a||b)}\mathrm{cor}(u \cdot x \oplus v \cdot \underset{\rightarrow}{\overleftarrow{F}}(x))\mathrm{cor}(u \cdot x \oplus v \cdot F(x)) \\
&= 2^{2n} \sum_{u,v \in \mathbb{F}_2^n} (-1)^{u\cdot a \oplus v \cdot b}\mathrm{cor}(u \cdot x \oplus v \cdot \underset{\rightarrow}{\overleftarrow{F}}(x))\mathrm{cor}(u \cdot x \oplus v \cdot F(x)) \\
&= 2^{2n} \sum_{u,v \in \mathbb{F}_2^n} (-1)^{u\cdot a \oplus v \cdot b}\mathrm{cor}(\overrightarrow{u} \cdot x \oplus \overrightarrow{v} \cdot F(x))\mathrm{cor}(u \cdot x \oplus v \cdot F(x))
\end{aligned}$$

□

If the function F is rotation invariant, i.e., $\overleftarrow{F(x)} = F(\overleftarrow{x})$, then we have $\mathrm{cor}(\overrightarrow{u} \cdot x \oplus \overrightarrow{v} \cdot F(x)) = \mathrm{cor}(u \cdot x \oplus v \cdot F(x))$. As a result, the theoretical link between rotational-xor and linear cryptanalysis degenerates to the link between ordinary differential cryptanalysis and linear cryptanalysis. Moreover, based on the link between differential and linear cryptanalysis, Blondeau *et al.* derive a closed formula for the bias of an ordinary differential-linear distinguisher as shown in Equation (8). We try to mimic Blondeau *et al.*'s approach to obtain a closed formula for the biases of rotational differential-linear distinguishers. However, we failed in this attempt due to a fundamental difference between rotational-xor differentials and ordinary differentials: the output RX-difference is not necessarily zero when the input RX-difference $\mathtt{rot}(x) \oplus x'$ is zero. We leave it as an open problem to derive a closed formula for the bias of a rotational differential-linear distinguisher. From a practical point of view, we do not lose much due to the absence of a closed formula since this kind of formula will inevitably involve the correlations of exponentially many trails which are hard to evaluate in most situations.

3.2 Morawiecki *et al.*'s Technique Revisited

In [MPS13], Morawiecki *et al.* performed a rotational cryptanalysis on the Keccak-f permutation E. In this attack, the probability of

$$\Pr[(E(x))_{i-t} \neq (E(x \lll t))_i]$$

was exploited to distinguish the target. In what follows, we show that Morawiecki *et al.*'s technique can be regarded as a special case of the rotational differential-linear framework.

Eventually, what we exploit in a rotational differential-linear attack associated with an input RX-difference $\delta \in \mathbb{F}_2^n$ and an output linear mask $\gamma \in \mathbb{F}_2^n$ is the abnormally high absolute bias or correlation of the Boolean function

$$\gamma \cdot (\mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta)).$$

Following the notation of [BLN17], let $\mathrm{sp}(\gamma) \subseteq \mathbb{F}_2^n$ be the linear space spanned by γ, and $\mathrm{sp}(\gamma)^\perp = \{u \in \mathbb{F}_2^n : \forall v \in \mathrm{sp}(\gamma), u \cdot v = 0\}$ be the orthogonal space of $\mathrm{sp}(\gamma)$.

We then define two sets $\mathbb{D}_0$ and $\mathbb{D}_1$ which form a partition of $\mathbb{F}_2^n$:

$$\begin{cases} \mathbb{D}_0 = \{x \in \mathbb{F}_2^n : \mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta) \in \mathrm{sp}(\gamma)^{\perp}\} \\ \mathbb{D}_1 = \{x \in \mathbb{F}_2^n : \mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta) \in \mathbb{F}_2^n - \mathrm{sp}(\gamma)^{\perp}\} \end{cases}.$$

Under the above notations, for any $x \in \mathbb{D}_0$, $\gamma \cdot (\mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta)) = 0$ and for any $x \in \mathbb{D}_1$, $\gamma \cdot (\mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta)) = 1$.

Thus, the higher the absolute value of

$$\#\mathbb{D}_0 - \#\mathbb{D}_1 = 2^n \mathrm{cor}(\gamma \cdot (\mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta))),$$

the more effective the attack is.

If $\gamma = e_i$ is the i-th unit vector, we have $\mathrm{sp}(\gamma) = \{0, e_i\}$ and $\mathrm{sp}(\gamma)^{\perp}$ contains all vectors whose i-th bit is 0. In this case,

$$\begin{aligned} \#\mathbb{D}_0 - \#\mathbb{D}_1 &= 2^n - 2\#\mathbb{D}_1 \\ &= 2^n - 2^{n+1} \left(\Pr[e_i \cdot (\mathtt{rot}(E(x)) \oplus E(\mathtt{rot}(x) \oplus \delta)) = 1]\right) \\ &= 2^n - 2^{n+1} \left(\Pr[(E(x))_j \neq (E(\mathtt{rot}(x) \oplus \delta))_i]\right) \\ &= 2^n - 2^{n+1} \left(\Pr[(E(x))_j \neq (E(x')_i]\right). \end{aligned}$$

Therefore, the effectiveness of the rotational differential-linear attack can be completely characterized by $\Pr[(E(x))_{i-t} \neq (E(x'))_i]$. In the next section, we show how to compute this type of probabilities for the target cipher.

4 Evaluate the Bias of Rotational Differential-linear Distinguishers

According to the previous section, for a rotational differential-linear distinguisher with an input RX-difference δ and output linear mask e_i, the bias of the distinguisher can be completely determined by

$$\Pr[(E(x))_{i-t} \neq (E(x'))_i], \text{ where } x' = x \lll t \oplus \delta,$$

and we call it the rotational differential-linear probability or R-DL probability. Note that for a random pair $(x, x' = x \lll t \oplus \delta)$ with rotational-xor difference $\delta \in \mathbb{F}_2^n$, we have

$$\Pr[x_{i-t} \neq x'_i] = \frac{1 + (-1)^{1-\delta_i}}{2},$$

for $0 \leq i < n$. Therefore, what we need is a method to evaluate the probability

$$\Pr[(F(x))_{i-t} \neq (F(x'))_i]$$

for $0 \leq i < m - 1$, where $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is a vectorial Boolean function that represents a component of E. Then, with certain independence assumptions, we can iteratively determine the probability $\Pr[(E(x))_{i-t} \neq (E(x'))_i]$.

Observation 1 *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function. Assume that the input pair (x, x') satisfies $\Pr[x_{i-t} \neq x'_i] = p_i$ for $0 \le i < n$, where $x, x' \in \mathbb{F}_2^n$. For $u \in \mathbb{F}_2^n$, we define the set $\mathcal{S}_u = \{(x, x') \in \mathbb{F}_2^n \times \mathbb{F}_2^n : (x \lll t) \oplus x' = u\}$ with $\#\mathcal{S}_u = 2^n$. Let y_i and y'_i be the i-th bit of $F(x)$ and $F(x')$ respectively for $0 \le i < m$. Then we have*

$$\begin{aligned}\Pr[y_{i-t} \neq y_i] &= \sum_{u \in \mathbb{F}_2^n} \Pr[y_{i-t} \neq y_i | (x, x') \in \mathcal{S}_u] \Pr[(x, x') \in \mathcal{S}_u] \\ &= \sum_{u \in \mathbb{F}_2^n} \Pr[y_{i-t} \neq y_i | (x, x') \in \mathcal{S}_u] \prod_{i=0}^{n-1} ((1 - u_i) - (-1)^{u_i} p_i) \\ &= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} \#\{(x, x') \in \mathcal{S}_u : y_{i-t} \neq y_i\} \prod_{i=0}^{n-1} ((1 - u_i) - (-1)^{u_i} p_i).\end{aligned}$$

The observation is inspired by Morawiecki *et al.*'s work on rotational cryptanalysis [MPS13] where, given a rotational pair, the bias of the output pair being unequal at certain bit is calculated for one-bit AND, NOT and XOR. In the following, we reformulate and generalize their propagation rules in terms of rotational differential-linear probability. Note that all these rules can be derived from Observation 1.

Proposition 1 (AND-rule). *Let a, b, a', and b' be n-bit strings with $\Pr[a_{i-t} \neq a'_i] = p_i$ and $\Pr[b_{i-t} \neq b'_i] = q_i$. Then*

$$\Pr[(a \wedge b)_{i-t} \neq (a' \wedge b')_i] = \frac{1}{2}(p_i + q_i - p_i q_i).$$

Proposition 2 (XOR-rule). *Let a, b, a', and b' be n-bit strings with $\Pr[a_{i-t} \neq a'_i] = p_i$ and $\Pr[b_{i-t} \neq b'_i] = q_i$. Then*

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

Proposition 3 (NOT-rule). *Let a and b be n-bit strings with $\Pr[a_{i-t} \neq b_i] = p_i$. Then $\Pr[\bar{a}_{i-t} \neq \bar{b}_i] = p_i$.*

Next, we consider constant additions. Let $(x, x') \in \mathbb{F}_2^{2n}$ be a data pair with $\Pr[x_{i-t} \neq x'_i] = p_i$ for some integer t and $c \in \mathbb{F}_2^n$ be a constant. Then $\Pr[(x \oplus c)_{i-t} \neq (x' \oplus c)_i] = \Pr[x_{i-t} \oplus x'_i \neq c_{i-t} \oplus c_i]$. In [MPS13], only the cases where $c_{i-t} \oplus c_i = 1$ or $c_{i-t} = c_i = 0$ are considered. We generalize the rule for constant addition from [MPS13] to the following proposition with all possibilities taken into account.

Proposition 4 (Adjusted C-rule). *Let a and a' be n-bit strings with $\Pr[a_{i-t} \neq a'_i] = p_i$ and $c \in \mathbb{F}_2^n$ be a constant. Then we have*

$$\Pr[(a \oplus c)_{i-t} \neq (a' \oplus c)_i] = \begin{cases} 1 - p_i, & c_{i-t} \oplus c_i = 1 \\ p_i, & c_{i-t} \oplus c_i = 0 \end{cases}$$

4.1 Propagation of R-DL Probabilities in Arithmetic Operations

For functions with AND-RX or LRX construction, such as the permutation Keccak-f, the propagation of the R-DL probability can be evaluated by the propositions previously shown, under the independency assumptions on the neighbouring bits. However, when dependency takes over, even if a function can be expressed as a boolean circuit, a direct applications of the AND, XOR, NOT and adjusted C-rule may lead to errors that accumulated during the iterated evaluation. One such example is the modular addition. In the following, we will derive the propagation rules of the differential-linear (DL) probability and R-DL probability for an n-bit modular addition.

Lemma 6 (carry-rule). *Let $\varsigma : \mathbb{F}_2^3 \to \mathbb{F}_2$ be the carry function*

$$\varsigma(x_0, x_1, x_2) = x_0x_1 \oplus x_1x_2 \oplus x_0x_2.$$

Let a, b, c, a', b', and c' be binary random variables with

$$p_0 = \Pr[a \neq a'], p_1 = \Pr[b \neq b'], p_2 = \Pr[c \neq c'].$$

Then, we have that

$$\Pr[\varsigma(a,b,c) \neq \varsigma(a',b',c')] = p_0p_1p_2 - \frac{p_0p_1 + p_0p_2 + p_1p_2}{2} + \frac{p_0 + p_1 + p_2}{2}.$$

Proof. We prove the carry-rule with Observation 1 by enumerating $u \in \mathbb{F}_2^3$. For $u = (0,0,0)$, $\Pr[\varsigma(a,b,c) \neq \varsigma(a',b',c') | a = a', b = b', c = c'] = 0$. For $u = (0,0,1)$, $\Pr[\varsigma(a,b,c) \neq \varsigma(a',b',c') | a = a', b = b', c \neq c'] = \Pr[a \oplus b = 1] = 1/2$ and $\prod_{i=0}^{2}((1-u_i) + (-1)^{1-u_i}p_i) = (1-p_a)(1-p_b)p_c$.

Similarly, one can derive the expression for all $u \in \mathbb{F}_{2^3}$, and we omit the details.The overall probability of the event $ab \oplus ac \oplus bc \neq a'b' \oplus a'c' \oplus b'c'$ is $p_ap_bp_c - (p_ap_b + p_ap_c + p_bp_c)/2 + (p_a + p_b + p_c)/2$. □

Based on the carry-rule, we can immediately prove the following two theorems on the DL and R-DL probabilities for n-bit modulo additions.

Theorem 3 ($\boxplus$-rule for DL). *Let x, y and x', y' be n-bit string, such that $\Pr[x_i \neq x_i'] = p_i$ and $\Pr[y_i \neq y_i'] = q_i$. Then, the differential-linear probability for modular addition can be computed as*

$$\Pr[(x \boxplus y)_i \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i$$

where $s_0 = 0$ and

$$s_{i+1} = p_iq_is_i - \frac{p_iq_i + p_is_i + q_is_i}{2} + \frac{p_i + q_i + s_i}{2}, i \leq n-1$$

Proof. For inputs x and y, denote the carry by

$$c = (x \boxplus y) \oplus x \oplus y = (c_{n-1}, \cdots, c_1, c_0),$$

where $c_0 = 0, c_{i+1} = x_i y_i \oplus x_i c_i \oplus y_i c_i$. Similarly, for x' and y', denote the carry by $c' = (c'_{n-1}, \cdots, c'_1, c'_0)$. Let s_i denote the probability $\Pr[c_i \neq c'_i]$. Then, $s_0 = 0$ and for $i \geq 1$, the event $c_i \neq c'_i$ is equivalent to

$$x_{i-1}y_{i-1} \oplus x_{i-1}c_{i-1} \oplus y_{i-1}c_{i-1} \neq x'_{i-1}y'_{i-1} \oplus x'_{i-1}c'_{i-1} \oplus y'_{i-1}c'_{i-1}.$$

Therefore, s_i can be computed as

$$p_{i-1}q_{i-1}s_{i-1} - (p_{i-1}q_{i-1} + p_{i-1}q_{i-1} + q_{i-1}s_{i-1})/2 + (p_{i-1} + q_{i-1} + s_{i-1})/2$$

according to Lemma 6. Since $x \boxplus y = x \oplus y \oplus c$, and $x' \boxplus y' = x' \oplus y' \oplus c'$, with the XOR-rule, we have

$$\Pr[(x \boxplus y)_i \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i.$$

□

Example 1. Consider an 8-bit modular addition with input difference being $a = 7$ and $b = 7$. Then, we have for $0 \leq i \leq 7$,

$$p_i = \frac{1 + (-1)^{1-a_i}}{2}, q_i = \frac{1 + (-1)^{1-b_i}}{2},$$

so

$$\begin{aligned} p_0 = p_1 = p_2 = 1, p_3 = p_4 = p_5 = p_6 = p_7 = 0, \\ q_0 = q_1 = q_2 = 1, q_3 = q_4 = q_5 = q_6 = q_7 = 0. \end{aligned}$$

The $\boxplus$-rule gives the output DL-probabilities in Table 2. The probabilities predicted in the table are verified by running through the 16-bit input space. In addition, we verified the $\boxplus$-rule in DL with all input differences on an 8-bit modular addition. Under the precision level given in Table 2, the experiments match the theoretical prediction perfectly.

Table 2: The DL-probabilities of an 8-bit modular addition with input differences $a = b = 7$ by theoretical evaluation, which are confirmed by experiments.

i	0	1	2	3	4	5	6	7
p_i	0	2^{-1}	$2^{-0.415037}$	$2^{-0.192645}$	$2^{-1.19265}$	$2^{-2.19265}$	$2^{-3.19265}$	$2^{-4.19265}$

As for the rotational differential-linear cryptanalysis of an n-bit modular addition, a left rotation by t bits is applied to the operands. Firstly, we present the $\boxplus$-rule for RX-difference with a rotation offset $t = 1$.

Theorem 4 ($\boxplus$-rule for RL, $t = 1$). *Given random n-bit strings x, y and x', y' such that $x' = (x \lll 1) \oplus a, y' = (y \lll 1) \oplus b$, where $\Pr[x_{i-1} \neq x'_i] = p_i, \Pr[y_{i-1} \neq y'_i] = q_i$. Then, the rotational differential-linear probability of the modular addition can be computed as*

$$\Pr[(x \boxplus y)_{i-1} \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i,$$

where $s_0 \approx 1/2, s_1 = 1/4$,

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

Proof. Denote $x = (x_{n-1}, \cdots, x_1, x_0)$, $y = (y_{n-1}, \cdots, y_1, y_0)$. Then

$$x' = ((x'_{n-1}, \cdots, x'_1, x'_0) = (x_{n-2} \oplus a_{n-1}, \cdots, x_0 \oplus a_1, x_{n-1} \oplus a_0)$$

$$y' = ((y'_{n-1}, \cdots, y'_1, y'_0) = (y_{n-2} \oplus b_{n-1}, \cdots, y_0 \oplus b_1, y_{n-1} \oplus b_0)$$

Let $c = (c_{n-1}, \cdots, c_0) = (x \boxplus y) \oplus x \oplus y$ and $c' = (c'_{n-1}, \cdots, c'_0) = (x' \boxplus y') \oplus x' \oplus y'$ be the two carries.

Let s_i denote the probability $\Pr[c_{i-1} \neq c'_i]$. When $i = 0$, $s_0 = \Pr[c_{n-1} \neq c'_0] = \Pr[x_{n-2}y_{n-2} \oplus x_{n-2}c_{n-2} \oplus y_{n-2}c_{n-2} = 0] \approx 1/2$, because the LHS term is balanced for independent random variables x and y. For $i = 1$, $s_1 = \Pr[c_0 \neq c'_1] = \Pr[x'_0y'_0 \neq 0] = 1/4$. For $i > 1$, s_i is equal to

$$\begin{aligned}\Pr[c_{i-1} \neq c'_i] &= \Pr[x_{i-2}y_{i-2} \oplus x_{i-2}c_{i-2} \oplus y_{i-2}c_{i-2} \neq x'_{i-1}y'_{i-1} \oplus x'_{i-1}c'_{i-1} \oplus y'_{i-1}c'_{i-1}] \\ &= p_{i-1}q_{i-1}s_{i-1} - \frac{p_{i-1}q_{i-1} + p_{i-1}s_{i-1} + q_{i-1}s_{i-1}}{2} + \frac{p_{i-1} + q_{i-1} + s_{i-1}}{2}\end{aligned}$$

For $x \boxplus y$ and $x' \boxplus y'$, applying the XOR-rule on the inputs and the carry vector gives

$$\Pr[(x \boxplus y)_{i-1} \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i$$

□

Example 2. Consider an 8-bit modular addition with input RX-difference (left rotate by 1-bit) being $a = 7$ and $b = 7$, which implies that

$$p_0 = p_1 = p_2 = 1, p_3 = p_4 = p_5 = p_6 = p_7 = 0,$$
$$q_0 = q_1 = q_2 = 1, q_3 = q_4 = q_5 = q_6 = q_7 = 0.$$

The R-DL probability of the i-th output bit, $0 \leq i < 8$ is given in Table 3. The probabilities predicted for $i \geq 2$ are verified by running through the 16-bit input space, and the probability for $i = 0$ is $2^{-1.01132}$ by experiment.

The experiments on an 8-bit modular addition show that the theoretical estimation of the DL and R-DL probabilities match the experiments well, except that the approximation in R-DL probability for the least significant bit has a marginal error in precision.

With a similar deduction, we give the following theorem for computing the R-DL probability through a modular addition under the condition that $\texttt{rot}(x) = x \lll t$, for an integer $2 \leq t \leq n-1$.

Table 3: The RL-probabilities of an 8-bit modular addition with input differences $a, b = 7$. $\texttt{rot}(x) = x \lll 1$. The index i represents the position of the output bit.

i	0	1	2	3	4	5	6	7
p	2^{-1}	2^{-2}	$2^{-0.678072}$	$2^{-0.29956}$	$2^{-1.29956}$	$2^{-2.29956}$	$2^{-3.29956}$	$2^{-4.29956}$

Theorem 5 **($\boxplus$-rule for RL for arbitrary $t > 1$).** *Given random n-bit strings x, y and x', y' such that $x' = x \lll t \oplus a, y' = y \lll t \oplus b$, where $\Pr[x_{i-1} \neq x'_i] = p_i, \Pr[y_{i-1} \neq y'_i] = q_i$. Then, the rotational differential-linear probability of the modular addition for $i \geq 0$ can be computed as*

$$\Pr[(x \boxplus y)_{i-1} \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i,$$

where $s_0 \approx 1/2, s_t = 1/2$,

$$s_{i+1} = p_iq_is_i - \frac{p_iq_i + p_is_i + q_is_i}{2} + \frac{p_i + q_i + s_i}{2}, 1 \leq i \leq n-1, i \neq t$$

Proof. Denote $x = (x_{n-1}, \cdots, x_1, x_0)$, $y = (y_{n-1}, \cdots, y_1, y_0)$, then

$$x' = ((x'_{n-1}, \cdots, x'_1, x'_0) = (x_{n-1-t} \oplus a_{n-1}, \cdots, x_{n-t+1} \oplus a_1, x_{n-t} \oplus a_0)$$

$$y' = ((y'_{n-1}, \cdots, y'_1, y'_0) = (y_{n-1-t} \oplus b_{n-1}, \cdots, y_0 \oplus b_1, y_{n-1} \oplus b_0).$$

Let $c = (c_{n-1}, \cdots, c_1, c_0)$ and $c' = (c'_{n-1}, \cdots, c'_1, c'_0)$ be the carries. Let s_i denote the probability $\Pr[c_{i-t} \neq c'_i]$. When $i = 0$,

$$s_0 = \Pr[c_{n-t} \neq c'_0] = \Pr[x_{n-t-1}y_{n-t-1} \oplus x_{n-t-1}c_{n-t-1} \oplus y_{n-t-1}c_{n-t-1} \neq 0] \approx 1/2$$

When $i = t$, $s_t = \Pr[c_0 \neq c'_t] = \Pr[x'_{t-1}y'_{t-1} \oplus x'_{t-1}c'_{t-1} \oplus y'_{t-1}c'_{t-1} \neq 0] \approx 1/2$ For all i, $i \neq 0, t$,

$$\begin{aligned} s_i &= \Pr[c_{i-t} \neq c'_i] \\ &= \Pr[x'_{i-1}y'_{i-1} \oplus x'_{i-1}c'_{i-1} \oplus y'_{i-1}c'_{i-1} \\ &\qquad \neq x_{n-t+i-1}y_{n-t+i-1} \oplus x_{n-t+i-1}c_{n-t+i-1} \oplus c_{n-t+i-1}y_{n-t+i-1}] \\ &= p_{i-1}q_{i-1}s_{i-1} - \frac{p_{i-1}q_{i-1} + p_{i-1}s_{i-1} + q_{i-1}s_{i-1}}{2} + \frac{p_{i-1} + q_{i-1} + s_{i-1}}{2}. \end{aligned}$$

Then, we have

$$\Pr[(x \boxplus y)_{i-t} \neq (x' \boxplus y')_i] = p_i + q_i - 2p_iq_i - 2p_is_i - 2q_is_i + 4p_iq_is_i.$$

□

The $\boxplus$-rules for DL and R-DL allows us to compute the partial DLCT of an n-bit modular addition accurately and efficiently. A naive application of Bar-On *et al.*'s approach [BDKW19] based on the Fast Fourier Transformation (FFT) by treating the modular addition as an $2n \times n$ S-box would require a complexity of $\mathcal{O}(2^{2n})$, where it requires a complexity of $O(n2^{2n})$ to obtain the n rows of the DLCT whose output masks are the unit vectors. In contrast, with the $\boxplus$-rule for DL, given the input difference, the DL-probability for all output masks that are unit vectors can be evaluated in $\mathcal{O}(n)$ operations, which achieves an exponential speed-up.

4.2 Finding Input Differences for Local Optimization

According to Proposition 1 and Proposition 2, for x and y in $\mathbb{F}_2$, if $\Pr[x \neq x'] = p_1, \Pr[y \neq y'] = p_2$, we have

$$\Pr[xy \neq x'y'] = \frac{1}{2}(p_1 + p_2 - p_1p_2), \quad \Pr[x \oplus y \neq x' \oplus y'] = p_1 + p_2 - 2p_1p_2.$$

Obviously, $\Pr[xy \neq x'y']$ is in the interval $[0, 0.5]$ and $\Pr[x \oplus y \neq x' \oplus y']$ is in the interval $[0, 1]$. Moreover, a behaviour of $\Pr[x \oplus y \neq x' \oplus y']$ is that it collapses to $\frac{1}{2}$ (e.g., correlation zero) whenever one of p_1 and p_2 is $\frac{1}{2}$. This observation suggests that the input probabilities should be biased from $\frac{1}{2}$ as much as possible. Otherwise, the probabilities will rapidly collapse to $\frac{1}{2}$ for all one-bit output masks after a few iterative evaluations of the round function.

In order to find distinguishers that cover as many rounds of a function F as possible, our strategy is to look for an input RX-difference δ, such that the DL or R-DL probability after one or a few propagations still has a relatively large imbalance for all the output masks whose Hamming weights are one. Therefore, we can define the objective function to maximize the summation of the absolute biases:

$$\sum_i (|\Pr[e_i \cdot (\texttt{rot}(f(x)) \oplus f(\texttt{rot}(x) \oplus \delta)) = 0] - 1/2|). \tag{11}$$

For 8-bit modular additions, we observed that the absolute DL and R-DL bias are relatively large when the input RX-differences are either with a large Hamming weight or a small weight. For instance, with RX-difference $(x \lll 1) \oplus x'$, when the input differences are $a = \texttt{0}$ and $b = \texttt{1}$, the RL-probabilities are given as follows for $e_i, i = 0, 1, \ldots, 7$.

$$2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}.$$

Whereas for $a = \texttt{ff}$ and $b = \texttt{ff}$, the RL-probabilities are given as follows for $e_i, i = 0, 1, \ldots, 7$.

$$2^{-1}, 2^{-2}, 2^{-0.678072}, 2^{-0.29956}, 2^{-0.142019}, 2^{-0.0692627}, 2^{-0.0342157}, 2^{-0.0170064}.$$

When the size of the operands are large (e.g., $n = 32$), it is difficult to find the optimal input difference manually. Next, we show the optimal input RX-difference with respect to the objective function given by Equation (11) in a 32-bit modular addition. See Appendix A for the search of such differences.

Example 3. Consider the R-DL probability for a 32-bit modular addition with $\mathtt{rot}(x) = x \lll 1$. With input RX-differences

$$a = \mathtt{7ffffffc}, b = \mathtt{7ffffffe},$$

the objective function in Equation 11 is maximized, and the R-DL probabilities $\Pr[e_i \cdot (\mathtt{rot}(x \boxplus y) \oplus ((\mathtt{rot}(x) \oplus a) \boxplus (\mathtt{rot}(y) \oplus b))) = 1]$ for $0 \leq i \leq 31$ are shown as follows.

i	0	1	2	3	4	5	6	7
p_i	0.5	0.75	0.5	0.75	0.875	0.9375	0.96875	0.984375
i	8	9	10	11	12	13	14	15
p_i	0.992188	0.996094	0.998047	0.999023	0.999512	0.999756	0.999878	0.999939
i	16	17	18	19	20 – 31			
p_i	0.999969	0.999985	0.999992	0.999996	1			

5 Applications to AND-RX Primitives

In this section, we apply the rotational differential-linear technique to the AND-RX permutations involved in `FRIET` and `Xoodoo`, and significant improvements are obtained. To confirm the validity of the results, all distinguishers with practical complexities are experimentally verified, and the source code is available[2].

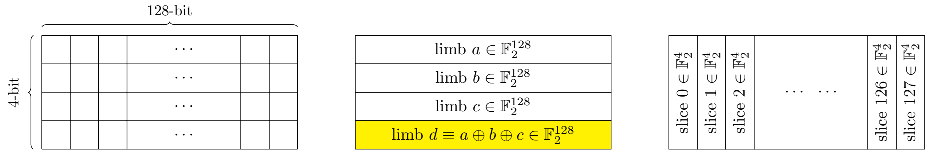
5.1 Distinguishers for Round-reduced `FRIET`

`FRIET` is an authenticated encryption scheme with built-in fault detection mechanisms proposed by Simon et al. at EUROCRYPT 2020 [SBD+20]. Its fault detection ability comes from its underlying permutation, which is designed based on the so-called *code embedding* approach.

The core permutation `FRIET`-P employed in `FRIET` operates on a $4 \times 128 = 512$-bit state arranged into a rectangular with 4 rows (called limbs) and 128 columns (called slices) as shown in Figure 1. The permutation `FRIET`-P is an iterative design with its round function g_{rc_i} visualized in Figure 2, where a, b, and $c \in \mathbb{F}_2^{128}$ are the four limbs (see Figure 1) of the input state and rc_i is the round constant for the i-th round.

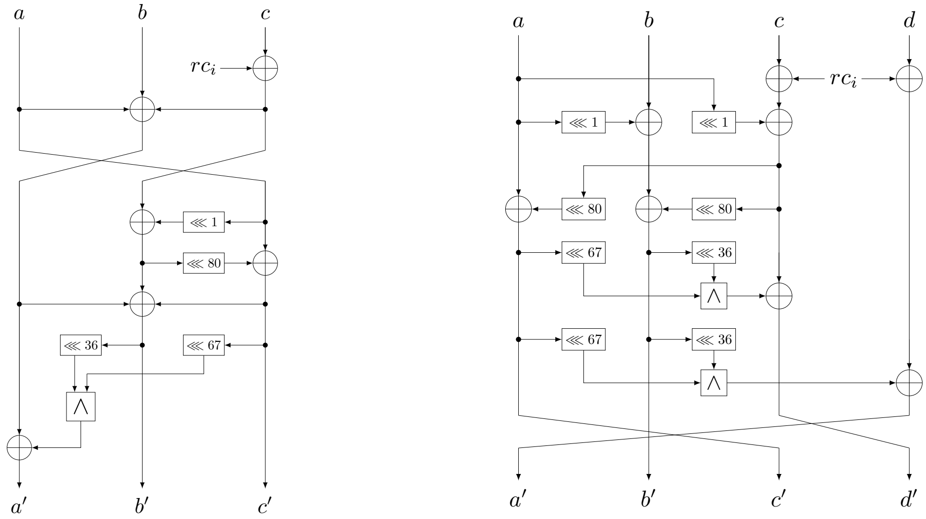
By design, the round function g_{rc_i} is slice-wise *code-abiding* for the parity code $[4, 3, 2]_{\mathbb{F}_2}$, meaning that every slice of the output state is a code word if every slice of the input state is a code word. Mathematically, it means that $a + b + c = d$ implies $a' + b' + c' = d'$. This slice-wise *code-abiding* property is inherited by the permutation `FRIET`-P $= g_{rc_{t-1}} \circ \cdots \circ g_{rc_1} \circ g_{rc_0}$. Consequently,

[2] `https://github.com/YunwenL/Rotational-cryptanalysis-from-a-differential-linear-perspective`


128-bit
4-bit
· · ·
limb $a \in \mathbb{F}_2^{128}$
limb $b \in \mathbb{F}_2^{128}$
limb $c \in \mathbb{F}_2^{128}$
limb $d \equiv a \oplus b \oplus c \in \mathbb{F}_2^{128}$
slice 0 $\in \mathbb{F}_2^4$
slice 1 $\in \mathbb{F}_2^4$
slice 2 $\in \mathbb{F}_2^4$
· · · · · ·
slice 126 $\in \mathbb{F}_2^4$
slice 127 $\in \mathbb{F}_2^4$

(a) The state (b) The limbs (c) The slices

Fig. 1: The view of the state



(a) FRIET-PC (b) FRIET-P

Fig. 2: The round functions of Friet-PC and Friet-P

faults will be detected if some output slice is not a code word when all of the slices of the input state are code words. Note that the behavior of the permutation FRIET-PC is identical to FRIET-P by design if we ignore the limb d.

Practical Distinguishers for FRIET-PC. Since a distinguisher for the permutation FRIET-PC directly translates to a distinguisher for FRIET-P, we focus on the permutation FRIET-PC. Let (a, b, c) and (a', b', c') in $\mathbb{F}_2^{128\times 3}$ be the input pair of the permutation with RX-differences

$$\Delta_a = (a \lll t) \oplus a', \quad \Delta_b = (b \lll t) \oplus b', \quad \Delta_c = (c \lll t) \oplus c'.$$

In our analysis, we only consider input RX-differences such that $wt(\Delta_a) + wt(\Delta_b) + wt(\Delta_c) \leq 1$.

According to the adjusted C-rule (see Proposition 4), the constant addition injects an RX-difference $c \oplus (c \lll t)$ to the state, and alters the R-DL-probabilities when the corresponding bits in $c \oplus (c \lll t)$ is nonzero. A rule-of-thumb for choosing the rotational amount is to minimize the weight of the

RX-difference introduced by the round constants, so that the effect of the constants on destroying the rotational propagation is presumably decreased. The first 6 round constants of `FRIET`-PC are (in Hexadecimal)

$$\texttt{1111}, \texttt{11100000}, \texttt{1101}, \texttt{10100000}, \texttt{101}, \texttt{10110000}.$$

To minimize the Hamming weight of the RX-differences from the round constants, one of the best rotational operations is to left rotate by 4 bits, such that the consecutive nonzero nibbles cancel themselves as many as possible. Then, the injected RX-differences due to the round constants are

$$\texttt{10001}, \texttt{100100000}, \texttt{10111}, \texttt{111100000}, \texttt{1111}, \texttt{111010000}.$$

With the AND-rule, XOR-rule and adjusted C-rule, the R-DL probability can be evaluated given the input RX-differences with $w_h(\varDelta_a)+w_h(\varDelta_b)+w_h(\varDelta_c) \leq 1$ and the output linear mask e_i. Table 4 shows the rotational differential-linear distinguishers with the largest absolute correlation we found in reduced-round `FRIET`-PC, where $\varDelta_a, \varDelta_b, \varDelta_c$ are the input RX-differences, and $\gamma_a, \gamma_b, \gamma_c$ are the output masks for the limbs a, b, c, respectively.

Table 4: Distinguishers for reduced-round `FRIET`-PC with rotation offset $t = 4$.

Round	$\varDelta_a$	$\varDelta_b$	$\varDelta_c$	γ_a	γ_b	γ_c	Correlation	
							Theoretical	Experimental
1	0	0	0	1	0	0	1	1
2	0	0	0	1	0	0	1	1
3	0	0	0	1	0	0	1	1
4	0	0	0	0	1	0	1	1
5	0	0	1	0	0	400000000000000000000	$2^{-0.96}$	$2^{-0.83}$
6	0	0	10000	0	0	40000	$2^{-5.81}$	$2^{-5.12}$

For `FRIET`-PC reduced to 4-round, an R-DL distinguisher with correlation 1 is detected, with input RX-differences $(0, 0, 0)$ and output masks $(0, 1, 0)$. For 5, 6-round `FRIET`-PC, we found practical rotational differential-linear distinguishers with correlation $2^{-0.96}$ and $2^{-5.81}$, respectively. All the distinguishers shown in Table 4 are verified experimentally with 2^{24} random plaintexts.

Extending the Practical Distinguishers. According to the discussion of Section 3, we can extend a rotational differential-linear distinguisher by appending a linear approximation $\gamma \to \mu$, and the bias of the extended distinguisher can be computed with Equation (7). Consequently, this extension is optimal when $\epsilon_{\gamma,\mu}$ and $\epsilon_{\mathtt{rot}^{-1}(\gamma),\mathtt{rot}^{-1}(\mu)}$ reach their largest possible absolute values simultaneously. For `FRIET`-PC, we always have $\epsilon_{\gamma,\mu} = \epsilon_{\mathtt{rot}^{-1}(\gamma),\mathtt{rot}^{-1}(\mu)}$, and thus we can focus on finding an optimal linear approximation $\gamma \to \mu$.

Here we take the 6-round R-DL distinguisher presented in Table 4 and append optimal linear approximations to extend it. The output linear mask of the 6-round distinguisher is $(\texttt{0}, \texttt{0}, \texttt{40000})$. In Table 5, we list the correlations of the optimal linear approximations for round-reduced FRIET-PC whose input masks are $(\texttt{0}, \texttt{0}, \texttt{40000})$, which are found with the SMT-based approach [KLT15].

Table 5: The correlation of optimal linear trails found in round-reduced FRIET-PC with the input masks $(\texttt{0}, \texttt{0}, \texttt{40000})$

# Round	1	2	3	4	5	6	7
Correlation	2^{-2}	2^{-6}	2^{-12}	2^{-20}	2^{-30}	2^{-42}	2^{-56}

The optimal 1-round linear trail we found has output masks

$$\begin{aligned}\mu_a &= \texttt{00000000000000020000000000040000}\\ \mu_b &= \texttt{00004000000000020000000000040000}\\ \mu_c &= \texttt{00000000000080020000000000060000}.\end{aligned}$$

Thus a 7-round distinguisher can be built by concatenating the 6-round distinguisher with a 1-round linear approximation, and the estimated correlation is $2^{-5.81} \times 2^{-2\times 2} = 2^{-9.81}$. With 2^{24} pairs of inputs satisfying the input RX-difference, the output difference under the specified mask are biased with a correlation approximately $2^{-9.12}$. Similarly, by appending a 2-round linear trail with output masks

$$\begin{aligned}\mu_a &= \texttt{00000000000000030000000000060000}\\ \mu_b &= \texttt{00006000000000010000000030020000}\\ \mu_c &= \texttt{600000000000c0010000000000030000}.\end{aligned}$$

at the end of the 6-round rotational differential-linear distinguisher, we get a 8-round RL-distinguisher with a correlation $2^{-17.81}$. And with 2^{40} pairs of inputs satisfying the input RX-difference, we find the experimental correlation of the 8-round distinguisher is $2^{-17.2}$. As a comparison, the 7-,8-round linear trails presented in the specification of FRIET-PC have correlation 2^{-29} and 2^{-40}, respectively. With the linear trails shown in Table 5, the concatenated distinguisher can reach up to 13 rounds, with an estimated correlation $2^{-117.81}$.

5.2 Distinguishers for Round-reduced Xoodoo

Xoodoo [DHAK18] is a 384-bit lightweight cryptographic permutation whose primary target application is in the Farfalle construction [BDH+17]. The state of Xoodoo is arranged into a $4 \times 3 \times 32$ cuboid and the bit at a specific position is accessed as $a[x][y][z]$. One round of Xoodoo consists of the following operations.

$$
\begin{aligned}
& a[x][y][z] = a[x][y][z] \oplus \sum_{y} a[x-1][y][z-5] \oplus \sum_{y} a[x-1][y][z-14] \\
& a[x][1][z] = a[x-1][1][z], a[x][2][z] = a[x][2][z-11] \\
& a[0][0] = a[0][0] \oplus RC_i \\
& a[x][y][z] = a[x][y][z] \oplus ((a[x][y+1][z] + 1) * (a[x][y+2][z])) \\
& a[x][1][z] = a[x][1][z-1], a[x][2][z] = a[x-1][2][z-8]
\end{aligned}
$$

The total number of rounds in Xoodoo is 12, and in some modes (Farfalle [BDH$^+$17] for instance), the core permutation calls a 6-round Xoodoo permutation. The round constants of Xoodoo are shown in the following, and for Xoodoo reduced to r rounds, the round constants are $c_{-(r-1)}, \cdots, c_0$.

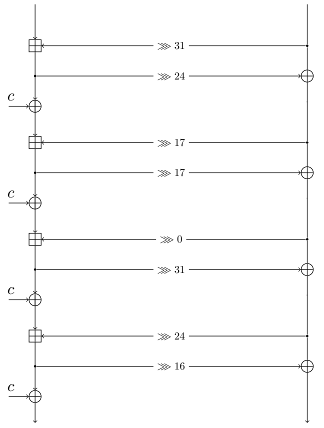
$$
\begin{array}{rrrr}
c_{-11} = \mathtt{00000058}, & c_{-8} = \mathtt{000000D0}, & c_{-5} = \mathtt{00000060}, & c_{-2} = \mathtt{000000F0} \\
c_{-10} = \mathtt{00000038}, & c_{-7} = \mathtt{00000120}, & c_{-4} = \mathtt{0000002C}, & c_{-1} = \mathtt{000001A0} \\
c_{-9} = \mathtt{000003C0}, & c_{-6} = \mathtt{00000014}, & c_{-3} = \mathtt{00000380}, & c_{0} = \mathtt{00000012}
\end{array}
$$

Given input difference being all-zero, *i.e.*, the input pair is exactly a rotational pair, let the rotation amount be left-rotate by 1-bit. We find that after 3 rounds of Xoodoo, there are still many output bits that are highly biased, with the largest correlation being 1 and the one-bit mask at position $(1, 0, 16)$. This suggests a nonzero mask 10000 at the lane $(1, 0)$. However, extending one extra round, we no longer see any significant correlation.

Noticing that the round constant is XORed into the state right after the first two linear operations, one can control the input RX-difference such that the difference is cancelled by the injection of the first-round constant. As a result, it gains one round free at the beginning, and we are able to construct a 4-round distinguishers for Xoodoo. When the left-rotational amount is set to 1-bit, the RX-difference of the first constant c_{-3} is 00000480. This suggests that if we take input RX-differences

$$
\begin{aligned}
& a[0][0] = \mathtt{484ccc80}; \mathrm{a}[0][1] = \mathtt{484cc800}; \mathrm{a}[0][2] = \mathtt{484cc800}; \\
& a[1][0] = \mathtt{3ab9821a}; \mathrm{a}[1][1] = \mathtt{3ab9821a}; \mathrm{a}[1][2] = \mathtt{3ab9821a}; \\
& a[2][0] = \mathtt{37b6cde9}; \mathrm{a}[2][1] = \mathtt{37b6cde9}; \mathrm{a}[2][2] = \mathtt{37b6cde9}; \\
& a[3][0] = \mathtt{45a3f0cb}; \mathrm{a}[3][1] = \mathtt{45a3f0cb}; \mathrm{a}[3][2] = \mathtt{45a3f0cb}.
\end{aligned}
$$

The RX-difference after the first round of Xoodoo will be all zero. Hence, we are able to find a 4-round distinguishers with significant correlations. We find a rotational differential-linear distinguishers with correlation 1 with the output mask being 10000 at lane $(1, 0)$ and zero for the rest lanes. Another two distinguishers with the same correlation are found with output mask 20000 at lane $(1, 1)$ and 1000000 at lane $(3, 2)$.



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

Fig. 3: The `Alzette` instance.

6 Applications to ARX Primitives

In this section, we apply the rotational differential-linear technique to the ARX permutations involved in `Alzette` and `SipHash`, and the source code for experimental verifications is available[3].

6.1 Application in the 64-bit ARX-box `Alzette`

At CRYPTO 2020, Beierle et al. presented a 64-bit ARX-box `Alzette` [BBdS$^+$20] that is efficient for software implementation. The design is along the same research line with a previous design called SPARX [DPU$^+$16] with a 32-bit ARX-box where a long trail argument was proposed for deriving a security bound in ARX ciphers. Figure 3 shows an instance of `Alzette` with an input $(x, y) \in \mathbb{F}_2^{32} \times \mathbb{F}_2^{32}$. The differential and linear properties of `Alzette` is comparable to the 8-bit S-box of AES. The optimal differential characteristic in `Alzette` has a probability of 2^{-6}. In addition, because of the modular additions in `Alzette` and the diffusion, the designers showed by division property that the `Alzette` may have full degree in all its coordinates.

In the following, we present the rotational differential-linear and differential-linear distinguishers of `Alzette` found with the techniques in Section 4. The constant $c =$ `B7E15162` (the first constant in SPARX-based design Sparkle-128) is considered for illustration.

Rotational differential-linear distinguisher. In Section 4.2, (`7ffffffc`, `7ffffffe`) is found to be optimal in 32-bit modular addition under the objective function considered in Example 3. Here, the difference can be used as the input difference of the first modular addition in `Alzette`. Because of the right rotation

[3] `https://github.com/YunwenL/Rotational-cryptanalysis-from-a-differential-linear-perspective`.

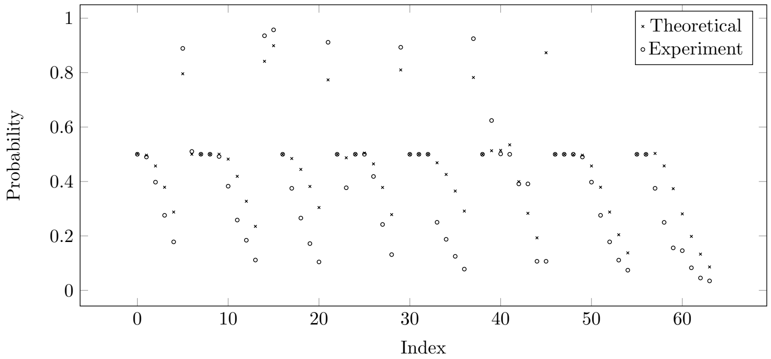


Fig. 4: A comparison between the differential-linear probability in `Alzette` by theoretical computation and by experiment. The index shows the index of the nonzero bit in the unit-vector output mask. For instance, when the index is 0, the output mask is (`0`,`1`), and when the index is 63, it is (`80000000`,`0`).

by 31 bits before the modular addition, the input RX-difference to `Alzette` is (`7ffffffc`, `3fffffff`). With an iterative evaluation on the steps in `Alzette`, we found that the second least significant bit is biased. Specifically, with an output mask $(2, 0)$, the RL-probability is 0.500189, that is a correlation $2^{-11.37}$. By taking 2^{28} pairs of random plaintexts, the experimental correlation of the distinguisher is $2^{-7.35}$. In addition, we checked all input RX-differences (a, b) with Hamming weight $wt(a) + wt(b) = 1$, but no rotational differential-linear distinguisher is found.

Differential-linear distinguisher. For all input differences with Hamming weight 1, we compute the differential-linear probability of `Alzette` with the technique in Section 4. The best found distinguisher has an input difference (`80000000`, `0`) and output mask (`80000000`, `0`), with a probability of 0.086, equivalently, a correlation of $2^{-0.27}$. By experiment verification with 2^{28} pairs of random plaintexts, the correlation is $2^{-0.1}$.

The following Figure 4 shows a comparison of the probability for an input difference (`80000000`, `0`) and output masks $(\mathtt{1} \lll t, \mathtt{0})$ (for all integer $t \in [0, 31]$), by our evaluation technique and the experiment with 2^{24} pairs of random plaintexts. The theoretical evaluation matches the experiment within a tolerable fluctuation.

Comparing with RL-distinguishers and DL-distinguisher found in `Alzette`, the latter is significantly stronger. Also, it is interesting to notice that input differences with low Hamming weight often lead to good differential-linear distinguishers in `Alzette`, whereas we didn't find any rotational differential-linear distinguisher with low-weight RX-differences when the rotational offset is greater than zero. The influence of the constants in RL-distinguishers may be the main cause.

6.2 Experimental Distinguishers for `SipHash` Explained

`SipHash` [AB12], designed by Aumasson and Bernstein, is a family of ARX-based pseudorandom functions optimized for short inputs. Instances of `SipHash` are widely deployed in practice. For example, `SipHash-2-4` is used in the dnscache instances of all `OpenDNS` resolvers and employed as `hash()` in `Python` for all major platforms (`https://131002.net/siphash/#us`).

In [HY19], from a perspective of differential cryptanalysis, a bias of the difference distribution of one particular output bit for 3-round `SipHash` is observed when the Hamming weight of the input difference is one. For instance, with input difference $a = 1$, He and Yu showed that the output difference is biased at the 27-th bit with a correlation 2^{-6} by experiments. This observation was obtained through extensive experiments and the theoretical reason behind these distinguishers is unclear as stated by He and Yu:

> "*... we are not concerned about why it shows a rotation property or why it reaches such a bias level. However, a great number of experiments can support those observations.* (see [HY19, Section 4.2, Page 11])"

According to the discussion of Section 3.2, the bias of $E(x) \oplus E(x \oplus \delta)$ observed in [HY19] is equivalent to the bias of

$$e_i \cdot (E(x) \oplus E(x \oplus \delta)).$$

It can be interpreted in the differential-linear framework and analyzed with the theoretical approach presented in Section 4. Here, we apply the rules for modular addition and XOR, and compute the DL-probability of the 3-round distinguisher found in SipHash. With our technique, we confirm that the 3-round differential-linear distinguisher with the aforementioned difference and mask, the predicted correlation is $2^{-6.6}$ which is close to He and Yu's experiments.

In addition, we can explain the observation on the rotation property with the $\boxplus$-rule in differential-linear. We will adopt the notations that are used in Theorem 3.

Because the input difference in their experiment has only one nonzero bit, we consider the DL-probability of an n-bit modular addition where the input difference is $(e_k, 0)$, for an integer k.

Then, for a pair of inputs (x, y) and (x', y'), the probability $p_k = \Pr[x_k \neq x'_k] = 1$. And for the remaining bits, $p_i = \Pr[x_i \neq x'_i], i \neq k$ and $q_i = \Pr[y_i, y'_i]$ are equal to zero.

Let $s_i = \Pr[\varsigma(x, y)_i \neq \varsigma(x', y')]$. We have $s_0, \cdots, s_k = 0, s_{k+t} = 2^{-t}, 1 \leq t \leq n - 1 - k$. As a result, the DL-probabilities through the modular addition at the i-th bit is given by $P_i = \Pr[(x \boxplus y)_i \neq (x' \boxplus y')_i], 0 \leq i \leq n - 1$, where

$$\Pr[(x \boxplus y)_i \neq (x' \boxplus y')_i] = \begin{cases} 0, & i \leq k \\ 2^{-i+k}, & \quad \text{otherwise} \end{cases} \tag{12}$$

By rotating the input difference $(\texttt{1} \lll k, 0)$ to the left by one bit, the differential-linear probability for the i-th bit of the output $\overleftarrow{P_i}$ is equal to 2^{-i+k+1} for $k + 1 < i \leq n - 1$, and to zero for $i \leq k + 1$.

It is obvious that the by rotating the differential-linear probability in Equation (12), we obtain the probabilities $\overleftarrow{P_i}$ for all but the least significant bit, where $\overleftarrow{P_0} = 0$ and $P_{n-1} = 2^{-n-1+k}$. Nevertheless, the error is negligible if $n - k$ is large, and it holds for large modular additions such as the 64-bit one adopted in SipHash.

For input differences with Hamming weight more than 1, a similar rotational property can be observed for the ⊞-rule in differential-linear. And it gives a straightforward intuition on the rotational property observed in the differential-linear distinguishers of SipHash.

7 Conclusion and Open Problems

We extend the differential-linear framework by using rotational-xor differentials in the differential part of the framework and we name the resulting cryptanalytic technique as rotational differential-linear cryptanalysis. We give an informal formula to estimate the bias of rotational differential-linear distinguisher under certain assumptions. In particular, we show Morawiecki et al.'s technique can be generalized to estimate the bias of a rotational differential-linear distinguisher whose output linear mask is a unit vector. We apply our method to the permutations involved in `FRIET`, `Xoodoo`, `Alzette`, and `SipHash`, which leads to significant improvements over existing cryptanalytic results or explanations for previous experimental distinguishers without a theoretical foundation. Finally, we would like to mention that we failed to derive a closed formula for the bias of a rotational differential-linear distinguisher under the sole assumption of the independence between the rotational-xor differential part and linear part. This is left open and the link between rotational-xor differential and linear cryptanalysis we presented in this work can be seen as a first step towards solving this problem.

A natural extension of rotational differential-linear cryptanalysis is to the SPN-type primitives, where one aims at finding a rotational relation that is preserved with a significant probability through the nonlinear Sbox layer. Especially, it is feasible to check all the rotational differences for their transition probabilities in a small-scale Sbox. Comparing to binary and arithmetic operations, our observation is that rotational relations are less likely to preserve in Sboxes, so it is challenging to find good distinguishers in Sbox-based designs. We leave it as an interesting future work.

Acknowledgement. We would like to thank the reviewers of Eurocrypt 2021 for their comments and suggestions to improve this paper. This work is supported by National Key R&D Program of China (2017YFB0802000, 2018YFA0704704), Natural Science Foundation of China (NSFC) under Grants 61902414, 61722213, 62032014, and 61772519, and the Chinese Major Program of National Cryptography Development Foundation (MMJJ20180102).

References

AB12. 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, Kolkata, India, December 9-12, 2012. Proceedings*, pages 489–508, 2012.

AJN14. Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. Analysis of NORX: investigating differential and rotational properties. In *Progress in Cryptology - LATINCRYPT 2014 - Third International Conference on Cryptology and Information Security in Latin America, Florianópolis, Brazil, September 17-19, 2014, Revised Selected Papers*, pages 306–324, 2014.

AL16. Tomer Ashur and Yunwen Liu. Rotational cryptanalysis in the presence of constants. *IACR Trans. Symmetric Cryptol.*, 2016(1):57–70, 2016.

BBdS$^+$20. Christof Beierle, Alex Biryukov, Luan Cardoso dos Santos, Johann Großschädl, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, and Qingju Wang. Alzette: A 64-bit arx-box - (feat. CRAX and TRAX). In *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, pages 419–448, 2020.

BBM20. Stefano Barbero, Emanuele Bellini, and Rusydi H. Makarim. Rotational analysis of ChaCha permutation. *CoRR*, abs/2008.13406, 2020.

BDH$^+$17. Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Farfalle: parallel permutation-based cryptography. *IACR Trans. Symmetric Cryptol.*, 2017(4):1–38, 2017.

BDKW19. Achiya Bar-On, Orr Dunkelman, Nathan Keller, and Ariel Weizman. DLCT: A new tool for differential-linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 313–342, 2019.

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

BLT20. Christof Beierle, Gregor Leander, and Yosuke Todo. Improved differential-linear attacks with applications to ARX ciphers. In *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, pages 329–358, 2020.

Can16. Anne Canteaut. Lecture notes on cryptographic boolean functions, 2016. `https://www.rocq.inria.fr/secret/Anne.Canteaut/`.

Car06. Claude Carlet. Boolean functions for cryptography and error correcting codes, 2006. `https://www.rocq.inria.fr/secret/Anne.Canteaut/`.

CHP$^+$18. Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 683–714, 2018.

CV94. Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 356–365, 1994.

DHAK18. Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. The design of Xoodoo and Xoofff. *IACR Trans. Symmetric Cryptol.*, 2018(4):1–38, 2018.

DPU^{+}16. 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, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pages 484–513, 2016.

HY19. Le He and Hongbo Yu. Cryptanalysis of reduced-round siphash. *IACR Cryptol. ePrint Arch. 2019/865*, 2019.

KAR20. Liliya Kraleva, Tomer Ashur, and Vincent Rijmen. Rotational cryptanalysis on MAC algorithm Chaskey. In *Applied Cryptography and Network Security - 18th International Conference, ACNS 2020, Rome, Italy, October 19-22, 2020, Proceedings, Part I*, pages 153–168, 2020.

KLT15. Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON block cipher family. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 161–185, 2015.

KN10. Dmitry Khovratovich and Ivica Nikolic. Rotational cryptanalysis of ARX. In *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, pages 333–346, 2010.

KNP^{+}15. Dmitry Khovratovich, Ivica Nikolic, Josef Pieprzyk, Przemyslaw Sokolowski, and Ron Steinfeld. Rotational cryptanalysis of ARX revisited. In *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, pages 519–536, 2015.

KNR10. Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational rebound attacks on reduced Skein. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 1–19, 2010.

LAAZ11. Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of PRINTcipher: The invariant subspace attack. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 206–221, 2011.

LGZL09. Zhiqiang Liu, Dawu Gu, Jing Zhang, and Wei Li. Differential-multiple linear cryptanalysis. In *Information Security and Cryptology - 5th International Conference, Inscrypt 2009, Beijing, China, December 12-15, 2009. Revised Selected Papers*, pages 35–49, 2009.

LH94. Susan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, pages 17–25, 1994.

LLA^{+}20. Jinyu Lu, Yunwen Liu, Tomer Ashur, Bing Sun, and Chao Li. Rotational-XOR cryptanalysis of Simon-like block ciphers. In *Information Security and Privacy - 25th Australasian Conference, ACISP 2020, Perth, WA, Australia, November 30 - December 2, 2020, Proceedings*, pages 105–124, 2020.

LMR15. Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of Robin, iSCREAM and Zorro. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 254–283, 2015.

Lu15. Jiqiang Lu. A methodology for differential-linear cryptanalysis and its applications. *Des. Codes Cryptogr.*, 77(1):11–48, 2015.

LWRA17. 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.

Mat93. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 386–397, 1993.

MPS13. 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.

SBD$^+$20. Thierry Simon, Lejla Batina, Joan Daemen, Vincent Grosso, Pedro Maat Costa Massolino, Kostas Papagiannopoulos, Francesco Regazzoni, and Niels Samwel. Friet: An authenticated encryption scheme with built-in fault detection. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, pages 581–611, 2020.

Tie16. Tyge Tiessen. Polytopic cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 214–239, 2016.

TLS19. Yosuke Todo, Gregor Leander, and Yu Sasaki. Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM, and Midori64. *J. Cryptol.*, 32(4):1383–1422, 2019.

TM16. Yosuke Todo and Masakatu Morii. Bit-based division property and application to Simon family. In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 357–377, 2016.

Tod15. Yosuke Todo. Structural evaluation by generalized integral property. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 287–314, 2015.

Wag99. David A. Wagner. The boomerang attack. In *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, pages 156–170, 1999.

Supplementary Material

A Finding Input Differences for Local Optimization with the Gurobi optimizer

In Section 4, we presented a rotational differential-linear distinguisher for the 32-bit modular addition, such that the function $\sum_{i=0}^{n-1}(|\Pr[e_i \cdot (\texttt{rot}(f(x)) \oplus f(\texttt{rot}(x) \oplus \delta)) = 0] - 1/2|)$ is maximized. This solution can be found with the Gurobi optimizer by converting the problem into a quadratic constraint programming problem.

The problem we consider here is to find the input RX-differences a, b, such that the value of the following objective function is maximized:

$$\sum_{i=0}^{n-1}(|\Pr[e_i \cdot (\texttt{rot}(x \boxplus y) \oplus ((\texttt{rot}(x) \oplus a) \boxplus (\texttt{rot}(y) \oplus b))) = 0] - 1/2|). \tag{13}$$

We assume that the input difference is some fixed value. Thus, the initial R-DL probabilities are zero or one. The constraints are all nonlinear, quadratic for AND-rule and XOR-rule, and cubic in $\boxplus$-rule.

Quadratic constraint programming(QCP) is a class of programming problems that optimize an objective function (quadratic or linear) given a set of quadratic constraints. The constraints can be inequalities or equations, and when it is the second case, the problem is called non-convex. The optimizer Gurobi can solve some QCP problems, convex or non-convex, and returns one or many solutions for the optimization. When the problem is non-convex, the optimizer solves it with a mixed-interger programming (MIP) strategy. In addition, the constraints in AND-rule and XOR-rule involves quadratic terms that are the cross-product of variables, that is to say, there is no terms with the form a^2, such constraints are called bilinear constraints.

To call Gurobi optimizer for QCP solving with Python, we need to set the following parameters for the model.

```
import gurobipy as gp
from gurobipy import GRB
from gurobipy import abs_
m = gp.Model("qcp")
m.params.NonConvex = 2
```

The intermediate probabilities during the evaluation are allocated as variables between 0 and 1, particularly the initial probabilities are integers.

```
a = m.addVar(0.0,1.0,0.0,name="a")
z = m.addVar(0.0,1.0,0.0,GRB.INTEGER,name="z")
```

To add a constraint, for instance, the XOR-rule $a + b - 2ab = p$, the clause to add is

```
m.addConstr(a + b - 2*a*b == p,"p")
```

After setting all constraints, we call `m.optimize()` to solve the model.