Functional Cryptanalysis

Application to reduced-round Xoodoo

Emanuele Bellini and Rusydi H. Makarim

Cryptography Research Centre,
Technology Innovation Institute (TII),
Abu Dhabi, United Arab Emirates
emanuele.bellini@tii.ae,rusydi.makarim@tii.ae

Abstract. This paper proposes *functional cryptanalysis*, a flexible and versatile approach to analyse symmetric-key primitives with two primary features. Firstly, it is a generalization of multiple attacks including (but not limited to) differential, rotational and rotational-xor cryptanalysis. Secondly, it is a theoretical framework that unifies all of the aforementioned cryptanalysis techniques and at the same time opens up possibilities for the development of new cryptanalytic approaches. The main idea of functional cryptanalysis is the usage of binary relations in the form of *functions*, hence the name *functional*, instead of binary operations like in a classical settings of "differential"-like cryptanalysis. We establish the theoretical foundations of functional cryptanalysis from standard terminologies. This work also presents an interpretation of functional cryptanalysis from the point of view of commutative algebra. In particular, we exhibit an algorithm to compute the functional probability (hence differential, rotational, and rotational-xor probability) using Gröbner bases. We demonstrate the applicability of functional cryptanalysis against reduced-round Xoodoo and compare it against the best differential. To avoid dealing with invalid differential trails, we propose a method to construct a valid differential trail using Satisfiability Modulo Theory (SMT). To the best of our knowledge, this is the first time the SMT model is used to construct a valid differential while previous approaches rely on Mixed-Integer Linear Programming (MILP) model. Lastly, we remark that the use of non-translation functionals shares analogous advantages and limitations with the use of nonlinear approximations in linear cryptanalysis.

Keywords: Functional Cryptanalysis · Differential Cryptanalysis · Rotational Cryptanalysis · Rotational-XOR Cryptanalysis · Xoodoo · SMT · Gröbner bases

1 Introduction

Nowadays, the security of a symmetric cipher is established by extensive analysis from the research community, by first applying standard cryptanalytic techniques and then more tailored approaches depending on the specific design of the cipher. It is often the case that new designs require a variation of a standard technique to be developed in order to mount an attack. For example, the introduction of ARX (Addition-Rotation-XOR) constructions pushed the rise of differential cryptanalysis where the difference is taken with respect to either modular arithmetic or rotation and XOR combined together.

Intuitively, for the case of differential cryptanalysis, the difference is usually chosen with respect to the binary operation that is used to inject the key into the cipher, so to be able to cancel the effect of the key during the propagation of differences. In fact, especially in the related-key scenario, this choice is not as trivial as it might seem. For example, as in the case of Speck [LWRA17], rotational-xor differences yield lower weight trails

than standard XOR differences. One might wonder if even other type of non-standard differences could be exploited, and one of the goal of this work is to explore this direction.

One other interesting problem is to unify the description of several differential attacks under a common framework, with the hope that by abstracting the problem, one could derive new forms of cryptanalysis. While elegant generalizations might be of theoretical interests per se, not all of them are suited to improve existing results in practice. In this work, we focus on the theory of first order differential cryptanalysis, and we try to achieve both goals of an elegant generalization and a practical application of our theory.

The most challenging part to mount a differential attack on a symmetric-key primitive is to find high probability differential trails. One approach is to construct it manually by hand which requires extensive analytical effort [WY05]. In a separate trend, there have been multiple works that utilizes constraint-based solving such as Mixed-Integer Linear Programming (MILP) [MWGP11], SAT [MP13], and Satisfiability Modulo Theory (SMT) [AJN14], to find differential trails. However, most of these works only models the propagation of differentials which are treated independently for each round. The drawback of this approach is that the trail may turn out to be invalid due to contradictions in the set of conditions implied from the differential trail [LIM20, SRB20].

1.1 Related Work

Differential cryptanalysis [BS93] is arguably one of the most fundamental techniques to analyze the security of symmetric-key primitives. Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function. Given a group operation $+$ over $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$, differential cryptanalysis studies the propagation of a *difference* $\alpha = x' - x$ in the input that leads to a difference $\beta = F(x') - F(x)$ in the output with nonzero probability.[1] The goal of an adversary is to find α, β yielding to a high probability, ideally the highest possible.

Since its first introduction, differential cryptanalysis has been extended and generalized in multiple directions. One direction is *truncated differential* [Knu94], where several differentials with many output differences that forms a linear space are clustered together. *Multiple differential cryptanalysis* [BG11] follows the same line of idea without imposing any restriction on the set of the output differences. Another direction is *higher-order differential* [Knu94, Lai94] which considers the propagation of a vector space of input differences, hence requiring more than a pair of inputs. It is also possible to consider other types of differences, such as *mod n cryptanalysis* [KSW99] or *rotational-xor cryptanalysis* [AL16], or cryptanalysis based on differences derived from *ad-hoc* operations [CBS19]. In [HO99], Hawkes and O'Connor showed that the differential probability is maximized with high probability when differences are defined with respect to the XOR operation. Finally, researchers have combined different forms of differential cryptanalysis with linear cryptanalysis as suggested for the first time by Langford and Hellman [CV94], who introduced the *differential-linear cryptanalysis* technique, or more recently by Liu et al. [LSL21] for the case of rotational-xor cryptanalysis.

There have been several attempts in the past aiming at expressing cryptanalysis techniques under a unified framework. A notable example, building on the frameworks of Vaudenay's *chi-squared cryptanalysis* [Vau96] and Harpes and Massey's *partitioning cryptanalysis* [HM97], is due to Wagner, who introduced the concept of *commutative diagram cryptanalysis* [Wag04]. Under this framework, the following attacks can be described: linear cryptanalysis, differential cryptanalysis, differential-linear cryptanalysis, mod n attacks, truncated differential cryptanalysis, impossible differential cryptanalysis, higher-order differential cryptanalysis, and interpolation attacks. Thanks to this framework, Wagner was able to generalize truncated differential cryptanalysis to *generalized* truncated differential cryptanalysis and interpolation attacks to *bivariate* interpolation attacks. So

[1] An adversary can also consider other binary operations.

far, we are not aware of any cryptanalytic result that has been improved by the application of these two attacks. The main idea of Wagner is to use commutative diagrams to model local properties of the round functions of a cipher, and to connect these local properties to determine a global property for the full cipher.

In the numerous works mentioned above, differential cryptanalysis and its generalizations tend to rely on defining a difference with a chosen *binary operation* in both input and output of the function. In this paper, we introduce a new paradigm to generalize differential cryptanalysis using a binary relation in a form of functions, hence the name *functional cryptanalysis*. We also remark that our work is not limited only to a generalization of differential cryptanalysis, but it is also a theoretical framework to describe multiple "differential-like" cryptanalysis technique (e.g. related-key differential [KSW96], rotational [KN10], and rotational-xor cryptanalysis [AL16]) in a unified manner. Most importantly, we show that functional cryptanalysis can be used to improve existing cryptanalysis results.

1.2 Contributions

The primary contributions of this work are fourfold which are summarized in each of the following points:

1. This paper proposes and establishes the theoretical foundation of *functional cryptanalysis*, a flexible and versatile approach for the cryptanalysis of symmetric-key primitives. Its primary feature is it generalizes multiple existing cryptanalysis techniques including, but not limited to, (related-key) differential, rotational and rotational-xor cryptanalysis. The versatility of functional cryptanalysis does not only serve as a generalization of all the aforementioned cryptanalysis techniques, but it also opens up other possible new constructions of cryptanalysis approaches that can not be captured by existing techniques in the literature. The key idea of functional cryptanalysis is the use of binary relations in the form of *functions*, hence the name *functional*, in order to express the relations of two inputs/outputs instead of binary operations. This idea allows the unification of cryptanalysis techniques that rely on the notion of "difference" (such as differential and rotational-xor cryptanalysis) with other techniques where the notion of "difference" can not be defined (such as rotational cryptanalysis).

2. In addition to the theoretical foundation, this work also provides a way to look at functional cryptanalysis (hence differential, rotational, and rotational-xor cryptanalysis) from an algebraic point of view. On this part, the main result is the following.

   Theorem 1. *Consider a function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and let $\mathcal{R}$ be a multivariate polynomial ring over $\mathbb{F}_2$ with $2(n+m)$ variables. For any $\sigma : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ and $\varphi : \mathbb{F}_2^m \mapsto \mathbb{F}_2^m$, there exists an ideal $I \subseteq \mathcal{R}$ such that the functional probability[2] of (σ, φ) on F is equal to*

   $$2^{-n} \cdot \dim_{\mathbb{F}_2}(\mathcal{R}/I)$$

   where $\dim_{\mathbb{F}_2}(\mathcal{R}/I)$ is the dimension of the factor ring $\mathcal{R}/I$ as an $\mathbb{F}_2$-vector space.

   The result above may reveal non-trivial algebraic implications of functional cryptanalysis by studying the structure of the ideal I. A follow-up result on this is an algorithm to compute the functional probability using tools from computational commutative algebra, mainly Gröbner bases algorithms.

[2] See Definition 11 for the definition of functional probability.

3. An advantage of functional cryptanalysis is demonstrated in this paper against the best known differential for a particular cipher[3]. To avoid dealing with invalid differential trails, such as in [LIM20], we introduce a way to automatically construct and verify differential trail using Satisfiability Modulo Theory (SMT). The main contribution on this part is an SMT model that returns a differential characteristic for a given probability (assuming it exists) and a pair of inputs that satisfy the characteristic. To the best of our knowledge, this is the first time an SMT model is used to verify the validity of a differential trail for a cipher. We have implemented this model for Xoodoo [DHAK18b] and verified its correctness. Thus, the proposed SMT model also serves as an alternative computational proof, in addition to trail-search approach [DHAK18a], on the bound of the best differential of Xoodoo.
4. This work also explains the strategy to build a functional distinguisher from existing differential trail. The resulting distinguisher improves upon the best differential, often by a significant factor. We experimentally verify it against round-reduced Xoodoo and show that a statistical assumption on the probability of functionals also holds in practice.[4]

1.3 Outline

Section 2 describes the notations and terminologies used throughout this paper. Section 3 introduces functional cryptanalysis. It starts by recalling differential, rotational, and rotational-xor cryptanalysis followed by the formalization of functional cryptanalysis. Then we discuss the propagation and the representation of the notion of functionals. Section 4 describes functional cryptanalysis from an algebraic point of view, where it provides the proof of Theorem 1. The algorithm to compute the functional probability using Gröbner bases and its complexity is also discussed. Section 5 presents the application of functional cryptanalysis on Xoodoo. It begins with an overview of Xoodoo, followed by the description of the SMT model to construct a valid differential characteristic for Xoodoo. We compare in practice the distinguishing attacks on round-reduced Xoodoo using functional and differential distinguisher. We discuss the limitation of functional cryptanalysis in Section 6 followed by some remark on its relation with nonlinear cryptanalysis. Finally, Section 7 concludes this paper by pointing to some possible future developments.

2 Preliminaries and Notations

For any set S, the notation $|S|$ denotes its cardinality. The notations $\mathbb{F}_2, \mathbb{F}_2^n$ denote the binary field and the n-dimensional vector space over $\mathbb{F}_2$ respectively. Addition in $\mathbb{F}_2^n$ is denoted by $+$ whereas addition modulo 2^n is denoted by $\boxplus$. For any $\alpha \in \mathbb{F}_2^n$, we define T_α to be the translation $T_\alpha(x) = x + \alpha$. We index elements starting from 1. The set of all polynomials in variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{F}_2$ is denoted by $\mathbb{F}_2[x_1, \ldots, x_n]$. For any $v \in \mathbb{F}_2^n$ and $w \in \mathbb{F}_2^m$ we define by $v \parallel w \in \mathbb{F}_2^{n+m}$ the vector constructed by concatenating v and w.

Definition 1 (Concatenated Function)**.** Let s, t, n be positive integers. For any function F,

$$\begin{aligned} F : (\mathbb{F}_2^n)^s &\mapsto (\mathbb{F}_2^n)^t \\ (x_1, \ldots, x_s) &\mapsto (y_1, \ldots, y_t) \end{aligned} \tag{1}$$

[3]The reason the comparison is focused on differential cryptanalysis rather than rotational-(xor) since the former is generically applicable on any ciphers and the latter is a dedicated technique towards ARX (Addition-Rotation-XOR) based ciphers.

[4]The implementation is available under an open-source license in https://github.com/Crypto-TII/functional_cryptanalysis/.

where $y_j = F_j(x_1, \ldots, x_s), j = 1, \ldots, t$ with $F_j : (\mathbb{F}_2^n)^s \mapsto \mathbb{F}_2^n$ be the coordinate function of F, we define the *concatenated function* $C[F] : \mathbb{F}_2^{ns} \mapsto \mathbb{F}_2^{nt}$ of F as $C[F](x_1 \parallel \cdots \parallel x_s) = (y_1 \parallel \cdots \parallel y_t)$.

For any binary operation $*$ on $\mathbb{F}_2^n$ we often write $*(x, y) = x * y$ for any $x, y \in \mathbb{F}_2^n$. We also define the following to denote bitwise rotation.

Definition 2. For any positive integer n and $0 \leq r < n$, we define the function $\rho_r : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ to be the following

$$\rho_r(x_1, \ldots, x_n) = (x_{r+1}, x_{r+2}, \ldots, x_n, x_1, x_2, \ldots, x_r). \tag{2}$$

We remark that the generalization of all results in this paper, which are discussed over $\mathbb{F}_2$, to other finite fields $\mathbb{F}_q$ is immediate to derive.

3 Overview of Functional Cryptanalysis

3.1 Differential Cryptanalysis

Differential cryptanalysis exploits a non-random probabilistic occurrence of a difference in the input and in the output of $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$. Concretely, it tries to find an input difference α and output difference β such that for any input pair $x', x \in \mathbb{F}_2^n$ to F where $x' - x = \alpha$, the output difference $F(x') - F(x) = \beta$ occurs with high probability.

Definition 3. A *differential* on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is a pair $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ and its *probability* is defined as

$$\mathsf{DP}_F(\alpha, \beta) = \Pr_{\mathbf{X}}[F(\mathbf{X} + \alpha) = F(\mathbf{X}) + \beta] \tag{3}$$

where the probability is taken from the distribution of $\mathbf{X}$. This work assumes that $\mathbf{X}$ is uniformly distributed in $\mathbb{F}_2^n$ in which case

$$\mathsf{DP}_F(\alpha, \beta) = 2^{-n} \cdot |\{x \in \mathbb{F}_2^n \mid F(x + \alpha) = F(x) + \beta\}|. \tag{4}$$

In a setting where n, m are small (such as substitution boxes), a straighforward approach to obtain high probability differentials is to compute the probability of all differentials on F. A convenient tool that achieves this is the *difference distribution table*.

Definition 4 (Difference Distribution Table (DDT))**.** For any function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, the *difference distribution table* $\mathsf{DDT}_F(\alpha, \beta)$ of F is a $2^n \times 2^m$ table where the entry at row $\alpha \in \mathbb{F}_2^n$ and column $\beta \in \mathbb{F}_2^m$ is defined as

$$\mathsf{DDT}_F(\alpha, \beta) = |\{x \in \mathbb{F}_2^n \mid F(x + \alpha) = F(x) + \beta\}|.$$

Clearly, finding high probability differentials on F by computing its DDT quickly becomes infeasible as n and m grows. However, in practice $F = F_r \circ F_{r-1} \circ \cdots \circ F_1$ is an r-round iterative permutation[5], denoted by F^r, constructed with finite iteration of the *round functions* $F_1, \ldots, F_r$. In such setting, the notion of *trail* is introduced which will be used to approximate the probability of an r-round differential, i.e. differential on an r-round iterative permutation.

Definition 5 (Differential Trail)**.** Let $F = F_r \circ F_{r-1} \circ \cdots \circ F_1$ be an r-round iterative cipher. An r-round *differential trail* on F is a sequence $\alpha_1, \alpha_2, \ldots, \alpha_{r+1} \in \mathbb{F}_2^n$ where (α_i, α_{i+1}) is a differential of F_i for $i \in \{1, 2, \ldots, r\}$ and its probability is defined as

$$\mathsf{DTP}_F(\alpha_1, \alpha_2, \ldots, \alpha_{r+1}) = \prod_{i=1}^{r} \mathsf{DP}_{F_i}(\alpha_i, \alpha_{i+1}). \tag{5}$$

[5]Such as a block cipher with a fixed-key or the permutation used in the sponge construction.

3.2 Rotational Cryptanalysis

Rotational cryptanalysis is a probabilistic attack introduced in [KN10] targeting mainly ARX ciphers. The goal of the attack is to fix a positive integer k such that for any pair of input $(x_1, \ldots, x_s), (x'_1, \ldots, x'_s)$ to $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^t$ where $x'_i = \rho_k(x_i), i = 1, \ldots, s$, the condition $y'_j = \rho_k(y_j), j = 1, \ldots, t$ on their respective output $(y_1, \ldots, y_t)$, $(y'_1, \ldots, y'_t)$ also holds with high probability. In the following, we introduce the notion of *rotational* and formalize rotational cryptanalysis in a more generic setting than the one in [KN10].

Definition 6. A *rotational* on $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^t$ is a pair $((k_i)_{i=1}^s, (k'_j)_{j=1}^t)$ where $0 \leq k_i, k'_j < n$ and its probability, denoted by $\mathsf{RP}_F((k_i)_{i=1}^s, (k'_j)_{j=1}^t)$, is defined as

$$\Pr_{\mathbf{X_1}, \ldots, \mathbf{X_s}}[F((\rho_{k_i}(\mathbf{X_i}))_{i=1}^s) = (\rho_{k'_j}(F_j(\mathbf{X_1}, \ldots, \mathbf{X_s})))_{j=1}^t].$$

where the probability is taken from the distribution of $\mathbf{X_i}, i = 1, \ldots, s$, which are uniformly distributed in $\mathbb{F}_2^n$.

Definition 7. Let $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$ be an r-round iterative permutation where $F = F_r \circ F_{r-1} \circ \cdots \circ F_1$ and $F_i : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$. An r-round *rotational trail* on F is a sequence $(k_{1,j})_{j=1}^s, (k_{2,j})_{j=1}^s, \ldots, (k_{r+1,j})_{j=1}^s$ where $((k_{i,j})_{j=1}^s, (k_{i+1,j})_{j=1}^s)$ is a rotational on F_i, $i = 1, \ldots, r$, and its probability is defined as

$$\prod_{i=1}^{r} \mathsf{RP}_{F_i}((k_{i,j})_{j=1}^s, (k_{i+1,j})_{j=1}^s).$$

The existing results on the rotational probability consider the values k_i, k'_j in a rotational $((k_i)_{i=1}^s, (k'_j)_{j=1}^t)$ to be equal to a fix value, say k. Such rotational has probability 1 for addition in $\mathbb{F}_2^n$ (bitwise XOR) since ρ_k is $\mathbb{F}_2$-linear,

$$\rho_k(+(x, y)) = +(\rho_k(x), \rho_k(y)). \tag{6}$$

The same probability also holds for the rotational $(k, k+k' \pmod n)$ on the bitwise rotation $\rho_{k'}$ since the composition of rotation is commutative, i.e. $\rho_{k'}(\rho_k(x)) = \rho_k(\rho_{k'}(x)) = \rho_{(k+k') \pmod n}(x)$. On the other hand, the rotational probability for addition modulo 2^n varies depending on the rotation amount k, which is equal to $2^{-2} \cdot (1 + 2^{k-n} + 2^{-k} + 2^{-n})$ [Dau05, Chapter 4]. The lower and the upper bound are reached with $k = n/2$ and $k = 1$ respectively. We remark that rotational cryptanalysis also works if an adversary defines ρ_r in the opposite orientation than the one in (2).

In order for a rotational to propagate deterministically through a translation, such as key addition or constant addition, it is necessary that the following holds

$$\rho_r(x) + c = \rho_r(x + c) \iff c = \rho_r(c). \tag{7}$$

This implies that if F is a keyed-permutation (e.g. a block cipher) then rotational cryptanalysis works in a related-key settings. Thus, the overall probability should also take into account the propagation of rotationals in the key schedule.

3.3 Rotational-XOR Cryptanalysis

A particular issue that was not extensively discussed in [KN10] and its subsequent works is the propagation of rotational through the translation (i.e., the injection of constants) where the condition in (7) does not hold. This is often used to assert the security of a cipher against rotational cryptanalysis, such as in SEA [SPGQ06] and ChaCha [Ber].

To overcome the above countermeasure, in [AL16], Ashur and Liu introduced *Rotational-XOR* (RX) cryptanalysis. The technique can be seen as a generalization of both rotational

and differential cryptanalysis. The main idea is to compose a bitwise rotation followed by a translation to construct a pair of input. This allows the notion of "difference" to be introduced for the case of rotational-xor.

Definition 8 (RX-difference)**.** The *RX-difference* $\Delta_\gamma(x, x')$ of $(x, x') \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ w.r.t. γ is defined as

$$\Delta_\gamma(x, x') = x + \rho_\gamma(x').$$

Definition 9. A *rotational-xor* (RX) on a function $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^t$ is a pair $((k_i, \alpha_i)_{i=1}^s, (k'_j, \alpha'_j)_{j=1}^t)$ where $0 \leq k_i, k'_j < n$ and $\alpha_i, \alpha'_j \in \mathbb{F}_2^n$ and its probability, denoted by $\mathsf{RXP}_F((k_i, \alpha_i)_{i=1}^s, (k'_j, \alpha'_j)_{j=1}^t)$, is defined as

$$\Pr_{\mathbf{X_1},\ldots,\mathbf{X_s}}[F((\rho_{k_i}(\mathbf{X_i}) + \alpha_i)_{i=1}^s) = (\rho_{k'_j}(F_j(\mathbf{X_1}, \ldots, \mathbf{X_s})) + \alpha'_j)_{j=1}^t].$$

Definition 10. Let $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$ be an r-round iterative permutation where $F = F_r \circ F_{r-1} \circ \cdots \circ F_1$ and $F_i : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$. An r-round *rotational-xor (RX) trail* on F is a sequence $(k_{1,j}, \alpha_{1,j})_{j=1}^s, (k_{2,j}, \alpha_{2,j})_{j=1}^s, \ldots, (k_{r+1,j}, \alpha_{r+1,j})_{j=1}^s$ where $((k_{i,j}, \alpha_{i,j})_{j=1}^s, (k_{i+1,j}, \alpha_{i+1,j})_{j=1}^s)$ is an RX on F_i, $i = 1, \ldots, r$, and its probability is defined as

$$\prod_{i=1}^{r} \mathsf{RXP}_{F_i}((k_{i,j}, \alpha_{i,j})_{j=1}^s, (k_{i+1,j}, \alpha_{i+1,j})_{j=1}^s).$$

RX cryptanalysis is the study of the propagation of RX-differences throughout different operations of a cipher. Let $z = x + y$ and $z' = x' + y'$, then

$$\Delta_\gamma(z, z') = z + \rho_\gamma(z') = (x + y) + \rho_\gamma(x' + y') = \Delta_\gamma(x, x') + \Delta_\gamma(y, y')$$

which shows that RX-difference propagates deterministically through addition in $\mathbb{F}_2^n$. Similarly, if we let $y = \rho_r(x)$ and $y' = \rho_r(x')$, then

$$\Delta_\gamma(y, y') = \rho_r(x) + \rho_\gamma(\rho_r(x')) = \rho_r(x) + \rho_r(\rho_\gamma(x')) = \rho_r(\Delta_\gamma(x, x'))$$

which proves that RX-difference also propagates with probability 1 through bitwise rotation. The only probabilistic propagation of RX-difference in an ARX cipher is via modular addition. In [AL16], the authors provided the formula to compute the probability of propagation of RX-difference with $\gamma = 1$.

3.4 Functional Cryptanalysis

At this point we have seen how the terminologies in differential, rotational, and rotational-xor cryptanalysis are interrelated, except for one case: there is no counterpart for the notion of "difference" in rotational cryptanalysis. This is one of the primary motivations for the introduction of functional cryptanalysis, i.e. to work in a cryptanalysis setting without having the need to define a "difference" between two inputs/outputs.

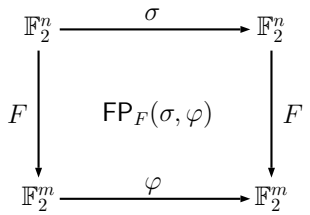

Figure 1: Probabilistic commutative diagram of a functional (σ, φ).

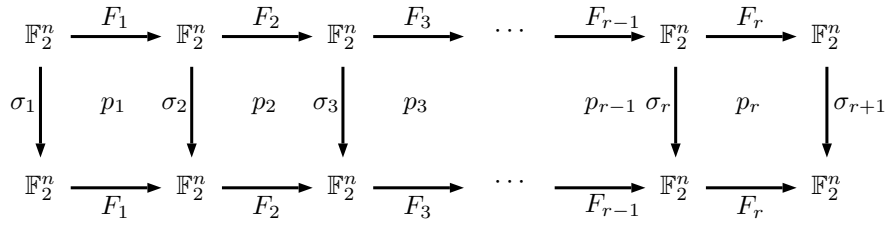

Figure 2: Probabilistic commutative diagram of a functional trail on r-round iterative permutation with $\mathsf{FP}_{F_i}(\sigma_i, \sigma_{i+1}) = p_i$ for $i = 1, 2, \ldots, r$.

Definition 11 (Functional and Functional Probability)**.** A *functional* on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is a pair (σ, φ) where $\sigma : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ and $\varphi : \mathbb{F}_2^m \mapsto \mathbb{F}_2^m$ and its probability is defined as

$$\mathsf{FP}_F(\sigma, \varphi) = \Pr_{\mathbf{X}}[F(\sigma(\mathbf{X})) = \varphi(F(\mathbf{X}))]$$

where $\mathbf{X}$ is uniformly distributed in $\mathbb{F}_2^n$ in which case

$$\mathsf{FP}_F(\sigma, \varphi) = 2^{-n} \cdot |\{x \in \mathbb{F}_2^n \mid F(\sigma(x)) = \varphi(F(x))\}|.$$

We refer to σ and φ as the input and output functions to F respectively.

Figure 1 illustrates the concept of a functional using a probabilistic commutative diagram. For the rest of this paper, we are interested to find a high probability functional on an r-round iterative permutation. Our approach to find a high probability functional for an iterative permutation is done by constructing a high probability functional on each round function and connecting them together. This approach is similar to other techniques such as linear and differential cryptanalysis on an iterative permutation. It is then natural to define the following notion.

Definition 12 (Functional Trail)**.** Let $F^r = F_r \circ F_{r-1} \circ \cdots \circ F_1$ be an r-round iterative permutation. An r-round *functional trail* of F is a sequence of mapping $\sigma_1, \sigma_2, \ldots, \sigma_{r+1}$ where (σ_i, σ_{i+1}) is a functional on F_i with $\mathsf{FP}_{F_i}(\sigma_i, \sigma_{i+1}) = p_i$ for $i = 1, 2, \ldots, r$. We often write it as

$$\sigma_1 \xrightarrow[p_1]{F_1} \sigma_2 \xrightarrow[p_2]{F_2} \cdots \xrightarrow[p_r]{F_r} \sigma_{r+1}$$

and the probability is defined as

$$\prod_{i=1}^{r} \mathsf{FP}_{F_i}(\sigma_i, \sigma_{i+1}).$$

Figure 2 gives a probabilistic commutative diagram of a functional trail on an iterative cipher. We will now discuss how differential, rotational, and rotational-xor cryptanalysis are described in terms of functional cryptanalysis.

Differential Cryptanalysis as Functional Cryptanalysis It is immediate to see that differential cryptanalysis is a special case of functional cryptanalysis. Indeed, a differential (α, β) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is a functional (T_α, T_β) on F and hence $\mathsf{DP}_F(\alpha, \beta) = \mathsf{FP}_F(T_\alpha, T_\beta)$. If $F^r = F_r \circ F_{r-1} \circ \cdots \circ F_1$ is an r-round iterative cipher, then any differential trail $\alpha_1, \alpha_2, \ldots, \alpha_{r+1}$ with $p_i = \mathsf{DP}_{F_i}(\alpha_i, \alpha_{i+1})$ for $i = 1, 2, \ldots, r$ is equal to the following functional trail

$$T_{\alpha_1} \xrightarrow[p_1]{F_1} T_{\alpha_2} \xrightarrow[p_2]{F_2} \cdots \xrightarrow[p_r]{F_r} T_{\alpha_{r+1}}.$$

In order to put differential cryptanalysis in the framework of functional cryptanalysis, we define the following class of functionals.

Definition 13 (Translation-Functional)**.** A functional (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is a *translation-functional* if both σ and φ are translations in $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ respectively.

Rotational Cryptanalysis as Functional Cryptanalysis For a positive integer s and for $k_1, \ldots, k_s$ we define the following function

$$\begin{aligned} \mathrm{P}_{(k_i)_{i=1}^s} : (\mathbb{F}_2^n)^s &\mapsto (\mathbb{F}_2^n)^s \\ (x_i)_{i=1}^s &\mapsto (\rho_{k_i}(x_i))_{i=1}^s. \end{aligned}$$

A rotational $((k_i)_{i=1}^s, (k'_j)_{j=1}^t)$ on $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^t$ corresponds to the functional $(C[\mathrm{P}_{(k_i)_{i=1}^s}], (C[\mathrm{P}_{(k'_j)_{j=1}^t}]))$ on the concatenated function $C[F]$ of F. Similarly $\mathsf{RP}_F((k_i)_{i=1}^s, (k'_j)_{j=1}^t) = \mathsf{FP}_{C[F]}(C[\mathrm{P}_{(k_i)_{i=1}^s}], C[\mathrm{P}_{(k'_j)_{j=1}^t}])$. Also for an r-round iterative cipher $F = F^r : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$ where $F^r = F_r \circ \cdots \circ F_1$, a rotational trail $((k_{1,i})_{i=1}^s, \ldots, (k_{r+1,i})_{i=1}^s)$ is equivalent to the following functional trail

$$C[\mathrm{P}_{(k_{1,i})_{i=1}^s}] \xrightarrow[p_1]{C[F_1]} C[\mathrm{P}_{(k_{2,i})_{i=1}^s}] \xrightarrow[p_2]{C[F_2]} \cdots \xrightarrow[p_r]{C[F_r]} C[\mathrm{P}_{(k_{r+1,i})_{i=1}^s}]$$

on the concatenated function $C[F^r]$ of F^r.

Note that since ρ_k is $\mathbb{F}_2$-linear for any $k \in \{0, 1, \ldots, n-1\}$, then $C[\mathrm{P}_{(k_i)_{i=1}^s}]$ is also $\mathbb{F}_2$-linear. Thus, we define the following particular class of functionals, which generalizes rotationals.

Definition 14 (Linear-Functional)**.** A functional (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is said to be a *linear-functional* if both σ and φ are $\mathbb{F}_2$-linear.

Rotational-XOR Cryptanalysis as Functional Cryptanalysis For a positive integer s and $(k_1, \alpha_1), \ldots, (k_s, \alpha_s) \in \mathbb{Z}_{\geq 0} \times \mathbb{F}_2^n$, we define the following function

$$\begin{aligned} \mathrm{X}_{(k_i,\alpha_i)_{i=1}^s} : (\mathbb{F}_2^n)^s &\mapsto (\mathbb{F}_2^n)^s \\ (x_i)_{i=1}^s &\mapsto (\rho_{k_i}(x_i) + \alpha_i)_{i=1}^s. \end{aligned}$$

It is immediate to see that a rotational-xor $((k_i, \alpha_i)_{i=1}^s, (k'_j, \alpha'_j)_{j=1}^t)$ on $F : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^t$ is a functional $(C[\mathrm{X}_{(k_i,\alpha_i)_{i=1}^s}], C[\mathrm{X}_{(k'_j,\alpha'_j)_{j=1}^t}])$ on the concatenated function $C[F]$ of F. Also the rotational-xor probability can be expressed in terms of functional probability as $\mathsf{RXP}_F((k_i, \alpha_i)_{i=1}^s, (k'_j, \alpha'_j)_{j=1}^t) = \mathsf{FP}_{C[F]}(C[\mathrm{X}_{(k_i,\alpha_i)_{i=1}^s}], C[\mathrm{X}_{(k'_j,\alpha'_j)_{j=1}^t}])$. Moreover, for the iterative cipher $F = F^r : (\mathbb{F}_2^n)^s \mapsto (\mathbb{F}_2^n)^s$, where $F^r = F_r \circ F_{r-1} \circ \cdots \circ F_1$, a rotational-xor trail $((k_{1,i}, \alpha_{1,i})_{i=1}^s, \ldots, (k_{r+1,i}, \alpha_{r+1,i})_{i=1}^s)$ is equivalent to the following functional trail

$$C[\mathrm{X}_{(k_{1,i},\alpha_{1,i})_{i=1}^s}] \xrightarrow[p_1]{C[F_1]} C[\mathrm{X}_{(k_{2,i},\alpha_{2,i})_{i=1}^s}] \xrightarrow[p_2]{C[F_2]} \cdots \xrightarrow[p_r]{C[F_r]} C[\mathrm{X}_{(k_{r+1,i},\alpha_{r+1,i})_{i=1}^s}]$$

on the concatenated function $C[F^r]$ of F^r.

Note that since the function $\rho_r(x) + \alpha$ is $\mathbb{F}_2$-affine for any $r \in \{0, 1, \ldots, n-1\}$ and $\alpha \in \mathbb{F}_2^n$, this implies that $C[\mathrm{X}_{(k_i,\alpha_i)_{i=1}^s}]$ is also $\mathbb{F}_2$-affine. This observation motivates the following definition of a class of functionals.

Definition 15 (Affine-Functional)**.** A functional (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is said to be an *affine-functional* if both σ and φ are $\mathbb{F}_2$-affine.

In Figure 3 we give an overall classification of the set of all functionals together with other techniques that can be expressed in terms of functionals. Note that an intersection among the set of all translation-functionals (differentials), rotationals, rotational-xor, and linear functionals contains only a functional (σ, φ) where both σ, φ are the identity mapping.

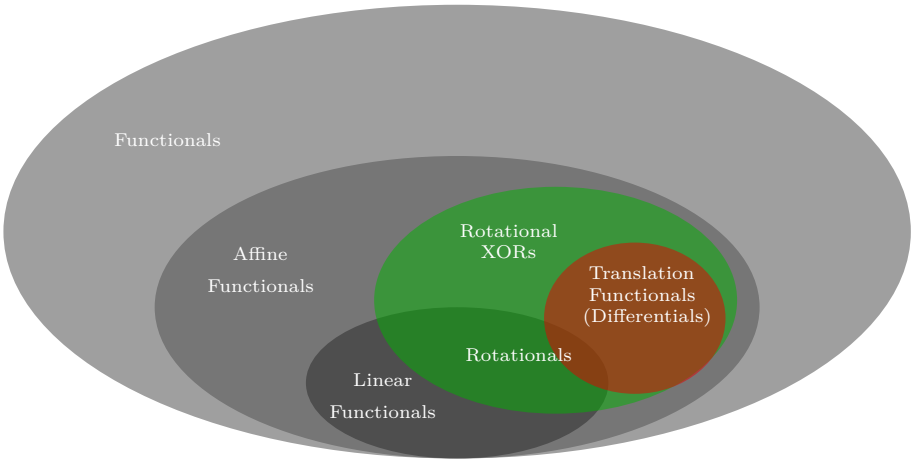

Figure 3: Classification of cryptanalysis techniques that belong to functional cryptanalysis.

3.5 Propagation of Functionals

In this subsection, we discuss the propagation property of functionals in the context of iterative permutations. We shall work in a setting where F, σ, φ are permutations in $\mathbb{F}_2^n$.

While cryptanalytic attacks deal with probabilistic propagation, an adversary is ideally interested to have a deterministic propagation. Given an input function σ to F, one immediately has a deterministic functional (σ, φ) on F where $\varphi = F \circ \sigma \circ F^{-1}$. It is interesting to note that if σ and F are elements of a group under a composition, then φ is also an element of the same group. We have observed this phenomena in the propagation of rotational through bitwise rotation as well as differential (translation-functional) via a translation.

For non-deterministic propagation of functionals, it is not immediately clear how to derive the set of all possible output functions corresponding to an input function σ and their functional probabilities. However, for relatively small n we introduce the *functional distribution table*.

Definition 16 (Functional Distribution Table)**.** The *functional distribution table* (FDT) of $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is a $2^{n \cdot 2^n} \cdot 2^{m \cdot 2^m}$ table $\mathsf{FDT}_F(\sigma, \varphi)$ where each row σ and each column φ of $\mathsf{FDT}_F(\sigma, \varphi)$ is defined as

$$\mathsf{FDT}_F(\sigma, \varphi) = |\{x \in \mathbb{F}_2^n \mid F(\sigma(x)) = \varphi(F(x))\}|. \tag{8}$$

FDT is a generalization of DDT since $\mathsf{DDT}_F(\alpha, \beta) = \mathsf{FDT}_F(T_\alpha, T_\beta)$ for any $(\alpha, \beta) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ and we also have $\mathsf{FP}_F(\sigma, \varphi) = 2^{-n} \cdot \mathsf{FDT}_F(\sigma, \varphi)$. However, it is defined for all mappings σ, φ only for the sake of completeness. In practice, FDT is not intended to be a tool that exhaustively describes the probability of all functionals on F since its size and the complexity to compute it quickly become infeasible even for $n = m = 4$. Rather, one can compute the FDT of a function F only for a particular subset of functionals.

3.6 Representation of Functionals and Functional Trails

Previously we saw in the definition of differential, rotational, and rotational-xor that they intrinsically provide representations for those notions. This, however, is not the case for functionals, in the sense that Definition 11 gives no immediate clue on an efficient representation. This section will address this issue and discuss several ideas to represent functional and functional trail. We shall begin by discussing generic representations that work for any functional (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$.

The first observation to make is that each function σ, φ in a functional (σ, φ) on F can be seen as a vectorial Boolean function. One can then represent σ and φ as lookup table. However, this is useful only if n, m are small such as when F is a substitution box.

For large n, m we propose to use system of multivariate polynomials to represent σ and φ. Suppose that $x_1, \ldots, x_n$ and $x'_1, \ldots, x'_n$ be variables that represent two inputs to F and let $y_1, \ldots, y_m$ and $y'_1, \ldots, y'_m$ be their corresponding output variables. Then σ and φ are represented by system of multivariate polynomials in variables $x_1, \ldots, x_n, x'_1, \ldots, x'_n$ and $y_1, \ldots, y_m, y'_1, \ldots, y'_m$ respectively with coefficients in $\mathbb{F}_2$. Concretely, σ is a system with $2n$ variables and n polynomials with input variables $x_1, \ldots, x_n$ and output variables $x'_1, \ldots, x'_n$ (or vice versa). Similarly φ is a system with $2m$ variables and m polynomials with input variables $y_1, \ldots, y_m$ and output variables $y'_1, \ldots, y'_m$ (or vice versa). This is naturally extended for the case of an r-round functional trail, which is represented by a sequence of r systems of multivariate polynomials with coefficients in $\mathbb{F}_2$.

Remark 1. From this point onwards, there are no distinction between the functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ and their representations as system of multivariate polynomials.

While such representation are useful for general cases, possible improvements can be done for specific functionals. For instance, a linear-functional (σ, φ) where σ and φ are bijective can be represented using $n \times n$ and $m \times m$ matrix with coefficients in $\mathbb{F}_2^n$ respectively. Also, a translation-functional can be represented like a differential (using an element of $\mathbb{F}_2^n \times \mathbb{F}_2^m$) and similarly for the case of functionals that correspond to rotationals and rotational-xors. Hence, *functional cryptanalysis views the notion of differential, rotational, and rotational-xor as representations for their corresponding functionals.*

4 Algebraic Perspective of Functional Cryptanalysis

This section presents functional cryptanalysis from an algebraic point of view. The central subjects here are the polynomial ring $\mathcal{R} = \mathbb{F}[x_1, \ldots, x_n]$ over a field $\mathbb{F}$ and ideals in $\mathcal{R}$. This turns to be a natural approach, due to the representation of functionals discussed in Subsection 3.6. Note that the generic nature of functional cryptanalysis also means that the results from this section are applicable for differential, rotational, and rotational-xor cryptanalysis. We shall begin by introducing notations used specifically in this section.

The ideal generated by $f_1, \ldots, f_m \in \mathcal{R}$ will be denoted by $\langle f_1, \ldots, f_m \rangle = \{\sum_{i=1}^m h_i f_i \mid h_i \in \mathcal{R}\}$. For any $F \subseteq \mathcal{R}$ and $f \in \mathcal{R}$ we allow a mixed notation and write $\langle F, f \rangle$ for the ideal generated by the elements in $F \cup f$. Whenever $\mathcal{R}$ is defined, we adopt a monomial ordering $<$ on the set $\mathsf{M}(x_1, \ldots, x_n)$ of all monomials[6] in variables $x_1, \ldots, x_n$. Examples of such orderings include *lexicographic* (lex), *degree lexicographic* (deglex) and *degree-reverse lexicographic* (degrevlex). Interested readers are referred to [CLO15, §2, Chapter 2] for their formal definitions. However, the results on this section are independent from the choice of variable ordering and the monomial ordering of $\mathcal{R}$. For any $0 \neq f \in \mathcal{R}$ we denote by $\mathsf{LM}(f)$ the leading monomial of f (i.e. the largest monomial in f according to the term ordering $<$) and similarly for any subset $F \subseteq \mathcal{R}$ we define $\mathsf{LM}(F) = \{\mathsf{LM}(f) \mid 0 \neq f \in F\}$. The residue class ring of $\mathcal{R}$ modulo an ideal I is denoted by $\mathcal{R}/I$, which is also an $\mathbb{F}$-vector space and its dimension is denoted by $\dim_{\mathbb{F}}(\mathcal{R}/I)$.

We now recall some relevant notions from computational commutative algebra.

Proposition 1 (Zero-Dimensional Ideal)**.** *Let $\overline{\mathbb{F}}$ be the algebraic closure of the field $\mathbb{F}$ and let $I = \langle f_1, \ldots, f_m \rangle$ be an ideal of $\mathcal{R}$. The following conditions are equivalent.*

1. *The system of equations $\{f_1 = 0, \ldots, f_m = 0\}$ has only finitely many solutions in $\overline{\mathbb{F}}^n$.*
2. *For $i = 1, 2, \ldots, n$ we have $I \cap \mathbb{F}[x_i] \neq \{0\}$.*

[6] By monomial we mean a polynomial which has only one term.

3. *The $\mathbb{F}$-vector space $\mathcal{R}/I$ is finite-dimensional.*

4. *The set $\mathsf{M}(x_1, \ldots, x_n) \setminus \mathsf{LM}(I)$ is finite.*

5. *For every $i \in \{1, \ldots, n\}$, there exists a non-negative integer a_i such that $x_i^{a_i} \in \mathsf{LM}(I)$.*

An ideal of $\mathcal{R}$ that satisfies one of the conditions above is called a zero-dimensional ideal.

Proof. See the proof of Proposition 3.7.1 in Section §3.7 of [KR00, pg. 243]. □

Definition 17 (Radical Ideal)**.** An ideal I of $\mathcal{R}$ is radical if $f^m \in I$ for some positive integer m implies that $f \in I$.

Proposition 2 (Seidenberg's Lemma)**.** *Let I be a zero-dimensional ideal of $\mathcal{R}$. Suppose that for every $i \in \{1, \ldots, n\}$ there exists a nonzero univariate polynomial $g_i \in I \cap \mathbb{F}[x_i]$ such that the greatest common divisor of g_i and its derivative is equal to 1. Then I is radical ideal*

Proof. See the proof of Proposition 3.7.15 in Section §3.7 of [KR00, pg. 250]. □

Definition 18 (Gröbner basis)**.** A finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal $\{0\} \neq I \subseteq \mathcal{R}$ is a Gröbner basis of I if $\langle \mathsf{LM}(g_1), \ldots, \mathsf{LM}(g_t) \rangle = \langle \mathsf{LM}(I) \rangle$.

Note that a Gröbner basis G generates its ideal, i.e., it is a basis of it.

Proposition 3. *For any ideal $\{0\} \neq I \subseteq \mathcal{R}$ we define $\mathsf{RM}(I) = \mathsf{M}(x_1, \ldots, x_n) \setminus \mathsf{LM}(I)$ the set of reduced monomials w.r.t. I. Then*

$$\begin{aligned} \mathsf{RM}(I) &= \{\mathsf{m} \in \mathsf{M}(x_1, \ldots, x_n) \mid \mathsf{m}' \nmid \mathsf{m}, \forall \mathsf{m}' \in \mathsf{LM}(I)\} \\ &= \{\mathsf{m} \in \mathsf{M}(x_1, \ldots, x_n) \mid \mathsf{m}' \nmid \mathsf{m}, \forall \mathsf{m}' \in \mathsf{LM}(G)\} \end{aligned}$$

where G is a Gröbner basis of I.

Proof. See the proof of Lemma 6.51 in [BWK93, pg. 272]. □

Proposition 4. *Let $\{0\} \neq I \subseteq \mathcal{R}$ be an ideal of $\mathcal{R}$. The set $\{\mathsf{m} + I \mid \mathsf{m} \in \mathsf{RM}(I)\}$ is a basis of the $\mathbb{F}$-vector space $\mathcal{R}/I$*

Proof. See the proof of Proposition 6.52 in [BWK93, pg. 273]. □

Proposition 5. *Let I be a zero-dimensional ideal of $\mathcal{R}$ and let $\overline{\mathbb{F}}$ be the algebraic closure of $\mathbb{F}$. Then the number of zeroes of I in $\overline{\mathbb{F}}^n$ is less than or equal to the $\dim_{\mathbb{F}}(\mathcal{R}/I)$. If $\mathbb{F}$ has characteristic zero or is a finite field and I is a radical ideal, then the equality holds.*

Proof. See the proof of Proposition 8.32 in [BWK93, pg. 348]. □

Definition 19. For any polynomial ring $\mathcal{R} = \mathbb{F}_q[x_1, \ldots, x_n]$ defined over a finite field $\mathbb{F}_q$, we define the *field polynomials* of $\mathcal{R}$ as

$$\mathcal{P}(\mathcal{R}) = \{x_i^q - x_i \mid \forall i = 1, \ldots, n\}.$$

4.1 Functional Ideal

The primary aim of having an algebraic interpretation of a functional on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is to help understanding the algebraic implication of the functional itself. From the setting of a polynomial ring and the representation of a functional proposed in Subsection 3.6, this is achieved by having an ideal in the polynomial ring that corresponds to a functional on F. For that we will work on the following ring

$$\mathcal{R} = \mathbb{F}_2[x_1, \ldots, x_n, y_1, \ldots, y_m, x'_1, \ldots, x'_n, y'_1, \ldots, y'_m]. \tag{9}$$

The variables that represent two inputs to F and their respective outputs as well as a functional (σ, φ) are the same as described in Subsection 3.6. We will now formally introduce the *functional ideal*. Note that, with abuse of notation and to avoid heavier notation, we indicate with F both a function and a set of polynomials representing the function.

Definition 20 (Functional Ideal)**.** Let $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and let $\mathcal{R}$ be a polynomial ring defined in (9). For any functional (σ, φ) on F we define the *functional ideal* $\mathsf{FI}_F(\sigma, \varphi) \subseteq \mathcal{R}$ of F w.r.t (σ, φ) as

$$\mathsf{FI}_F(\sigma, \varphi) = \langle F, F', \sigma, \varphi, \mathcal{P}(\mathcal{R}) \rangle$$

where F' is a set of polynomials representing F in variables $x'_1, \ldots, x'_n, y'_1, \ldots, y'_m$.

One can also define other ideals for linear and differential-linear cryptanalysis. We give their definitions in Appendix E for additional reference.

In principle the set $F \cup F' \cup \sigma \cup \varphi$ is sufficient to algebraically describe a functional (σ, φ) on F. However, the corresponding system of equations would have solutions that lie strictly in the algebraic closure of $\mathbb{F}_2$ but not in $\mathbb{F}_2$. Hence it is necessary to include $\mathcal{P}(\mathcal{R})$ in the basis of the ideal to remove such superfluous solutions. However, the field polynomials $\mathcal{P}(\mathcal{R})$ also serves other purposes which will be explained in the next subsection.

4.2 Computing Functional Probability using Gröbner Bases

Apart from removing the zeros of $F \cup F' \cup \sigma \cup \varphi$ that lie in a proper extension of $\mathbb{F}_2$, the inclusion of the field polynomials in the functional ideal has two more implications: it makes the ideal zero-dimensional (by Proposition 1) and radical (by Proposition 2). Hence, Proposition 5 implies that the dimension of $\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)$ determines the probability of the functional (σ, φ) on F. This proves Theorem 1 where $I = \mathsf{FI}_F(\sigma, \varphi)$. We formally restate it in the following theorem by giving the explicit construction of the ideal.

Theorem 2. *Let $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ and let $\mathcal{R}$ be a polynomial ring defined in (9). The probability of a functional (σ, φ) on F is equal to*

$$\mathsf{FP}_F(\sigma, \varphi) = 2^{-n} \cdot \dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)).$$

The construction of the functional ideal inherently does not give any information about its structure, including the dimension of $\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)$. One way to overcome this is to compute a Gröbner basis of $\mathsf{FI}_F(\sigma, \varphi)$. Some of the algorithms to compute a Gröbner basis of an ideal in $\mathcal{R}$ include the Buchberger [Buc06], F_4 [Fau99], F_5 [Fau02], and M4GB algorithm [MS17]. By Proposition 3 and Proposition 4, a Gröbner basis of $\mathsf{FI}_F(\sigma, \varphi)$ enables the construction of a basis of $\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)$. This implies that one can compute a functional probability (hence differential, rotational, and rotational-xor probability) using a Gröbner basis algorithm.

Theorem 3. *Given a functional (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, Algorithm 4.1 computes the functional probability of (σ, φ).*

Input: A function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$
Input: A functional (σ, φ) on F
Output: The functional probability $\mathsf{FP}_F(\sigma, \varphi)$
1 $\mathcal{R} \leftarrow \mathbb{F}_2[x_1, \ldots, x_n, y_1, \ldots, y_m, x'_1, \ldots, x'_n, y'_1, \ldots, y'_m]$
2 $G \leftarrow$ GröbnerBasis($\mathsf{FI}_F(\sigma, \varphi)$)
3 $\mathsf{RM}(\mathsf{FI}_F(\sigma, \varphi)) \leftarrow$ ReducedMonomials(G)
4 **return** $2^{-n} \cdot |\mathsf{RM}(\mathsf{FI}_F(\sigma, \varphi))|$

Algorithm 4.1: Algorithm to compute functional probability using a Gröbner basis.

Proof. Termination: The termination Algorithm 4.1 is obvious since the computation of Gröbner basis in line 2 and ReducedMonomials (see the description in Appendix F) in line 3 end after finite number of steps[7].
Correctness: Proposition 4 states that the set $\{\mathsf{m} + \mathsf{FI}_F(\sigma, \varphi) \mid \mathsf{m} \in \mathsf{RM}(\mathsf{FI}_F(\sigma, \varphi))\}$ is a basis of the $\mathbb{F}_2$-vector space $\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)$. Therefore by Theorem 2 we have $\mathsf{FP}_F(\sigma, \varphi) = 2^{-n} \cdot |\mathsf{RM}(\mathsf{FI}_F(\sigma, \varphi))|$. □

Example 1. In order to illustrate the result in Theorem 3, we will demonstrate it using the function $F : \mathbb{F}_2^3 \mapsto \mathbb{F}_2^3$ defined by the following polynomials

$$x_2x_3 + x_1 + x_3 + y_1,$$
$$x_1x_3 + x_1 + x_2 + y_2,$$
$$x_1x_2 + x_2 + x_3 + y_3.$$

Let (σ, φ) be the following functional on F

σ	φ
$x'_1 + x_1x_2 + x_1 + x_3 + 1$	$y'_1 + y_1 + 1,$
$x'_2 + x_1x_3 + x_2x_3 + x_2 + x_3 + 1,$	$y'_2 + y_2,$
$x'_3 + x_1x_3 + x_1 + x_2 + x_3 + 1,$	$y'_3 + y_3 + 1.$

The reduced Gröbner basis of $\mathsf{FI}_F(\sigma, \varphi)$ w.r.t. the degree-reverse lexicographic ordering consists of the following polynomials

$$\begin{array}{lll}
x_1'^2 + x'_1, & y'_1y'_2 + x'_3 + y'_2 + y'_3, & x_2 + y'_2 + y'_3, \\
x'_1x'_3 + x'_1 + y'_1, & y_2'^2 + y'_2, & x_3 + x'_3 + y'_2 + 1, \\
x_3'^2 + x'_3, & x'_1y'_3 + x'_1 + y'_1, & y_1 + y'_1 + 1, \\
x'_1y'_1 + y'_1, & x'_3y'_3 + x'_1 + y'_1, & y_2 + y'_2, \\
x'_3y'_1, & y'_1y'_3, & y_3 + y'_3 + 1, \\
y_1'^2 + y'_1, & y'_2y'_3 + x'_1 + y'_1 + y'_3, & x'_2 + y'_1 + y'_2, \\
x'_1y'_2 + x'_3 + y'_2 + y'_3, & y_3'^2 + y'_3, & \\
x'_3y'_2 + x'_1 + x'_3 + y'_1, & x_1 + x'_1 + y'_2. &
\end{array}$$

One can verify that $\dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{FI}_F(\sigma, \varphi)) = 6$ and therefore $\mathsf{FP}_F(\sigma, \varphi) = 0.75$.

The complexity of Algorithm 4.1 is clearly upper-bounded by the computation of a Gröbner basis of $\mathsf{FI}_F(\sigma, \varphi)$. The following theorem states the theoretical complexity of computing a functional probability using Algorithm 4.1.

[7] See Theorem 2 of [CLO15, pg. 91] for the proof of the termination of Buchberger algorithm or Theorem 2.2 of [Fau99] for the termination of F_4 algorithm. The termination of ReducedMonomials is due to the fact that $\mathsf{FI}_F(\sigma, \varphi)$ is a zero-dimensional ideal.

Table 1: The probability of the best known differential of Xoodoo [DHAK18a].

# rounds	1	2	3	4	5	6
probability	2^{-2}	2^{-8}	2^{-36}	$\geq 2^{-54}$	$\geq 2^{-56}$	$\geq 2^{-104}$

Theorem 4. *Assuming the semi-regularity of the system [BFSY05], the complexity of Algorithm 4.1 to compute the functional probability of (σ, φ) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ using the F_5 algorithm is equal to*

$$\mathcal{O}\left(\binom{2(n+m)+d_{\text{reg}}}{d_{\text{reg}}}^{\omega}\right) \tag{10}$$

where $2 \leq \omega \leq 3$ and d_{reg} is the index of the first non-positive coefficient of

$$\sum_{k\geq 0} c_k z^k = \frac{\prod_{i=1}^{5m+3n}(1-z^{d_i})}{(1-z)^{2(m+n)}} \tag{11}$$

with d_i being the degree of each polynomial f_i in $F \cup F' \cup \sigma \cup \varphi \cup \mathcal{P}(\mathcal{R})$.

Proof. The complexity of computing a Gröbner basis of a semi-regular zero-dimensional ideal in n variables and m equations using F_5 algorithm is equal to $\mathcal{O}(\binom{n+d_{\text{reg}}}{d_{\text{reg}}}^{\omega})$ where d_{reg} is the index of the first non-positive coefficient of $\prod_{i=1}^{m}(1-z^{d_i})/(1-z)^n$ [BFP09, BFSY05, BFS04]. Since the system of equations representing the functional ideal $\mathsf{FI}_F(\sigma, \varphi)$ consists of $2(n+m)$ variables and $5m+3n$ equations then (10) and (11) hold. □

The complexity formula in Equation 10 grows exponentially with m and n, meaning that the application of Algorithm 4.1 becomes quickly impractical for large sboxes.

5 Functional Cryptanalysis of Xoodoo

In this section we shall demonstrate the usefulness of functional cryptanalysis and compare it against the best differential of Xoodoo [DHAK18a]. We focus primarily on Xoodoo due to it being the underlying permutation of one of the finalists in the NIST lightweight standardization process. The security of Xoodoo against differential cryptanalysis was proven using a tree-search approach. Table 1 presents the probabilities of the best differentials found for its reduced-round variant. For practical purpose, we targeted the number of rounds where the complexity to mount distinguishing attack on Xoodoo using differential cryptanalysis is computationally feasible with our resources.

5.1 Description of Xoodoo

Xoodoo is a family of 384-bit permutation parameterized by the number of rounds n_r and denoted by Xoodoo$[n_r]$. It iteratively applies a round function on a state that can be seen as a 3×4 array of 32-bit words. The round function consists of 5 (five) steps: a mixing layer θ (`theta`), a composition of word-wise and bitwise rotation ρ_{west} (`rho west`), addition of round constants ι (`iota`), a nonlinear function χ (`chi`), and another composition of word-wise and bitwise rotation ρ_{east} (`rho east`). The complete description of Xoodoo$[n_r]$ is given in Algorithm 5.1 and the value of the round constants are specified in Table 5. In the description of the algorithm, the symbols $\neg, \wedge, \oplus$ denote the bitwise NOT, bitwise AND, and bitwise XOR respectively. For any $W \in \{0,1\}^{32}$ and $r \in \{1, 2, \ldots, 32\}$, $W \lll r$ denotes the bitwise left rotation of W by r bits with the rightmost bit of W as the least significant bit.

Input:
- Number of rounds n_r
- $A_{i,j} \in \{0,1\}^{32}$ for $i \in \{1,2,3\}, j \in \{1,2,3,4\}$

for $r = 1 - n_r$ **to** 0 **do**

/* theta */

$P_j \leftarrow A_{1,j} \oplus A_{2,j} \oplus A_{3,j}$ for $j \in \{1,2,3,4\}$

$E_j \leftarrow (P_{((j-2) \bmod 4)+1} \lll 5) \oplus (P_{((j-2) \bmod 4)+1} \lll 14)$ for $j \in \{1,2,3,4\}$

$A_{i,j} \leftarrow A_{i,j} \oplus E_j$ for $i \in \{1,2,3\}, j \in \{1,2,3,4\}$

/* rho west */

$A_{2,1}, A_{2,2}, A_{2,3}, A_{2,4} \leftarrow A_{2,4}, A_{2,1}, A_{2,2}, A_{2,3}$

$A_{3,j} \leftarrow A_{3,j} \lll 11$ for $j \in \{1,2,3,4\}$

/* iota */

$A_{1,1} \leftarrow A_{1,1} \oplus c_r$

/* chi */

$B_{1,j} \leftarrow \neg A_{2,j} \wedge A_{3,j}$ for $j \in \{1,2,3,4\}$

$B_{2,j} \leftarrow \neg A_{3,j} \wedge A_{1,j}$ for $j \in \{1,2,3,4\}$

$B_{3,j} \leftarrow \neg A_{1,j} \wedge A_{2,j}$ for $j \in \{1,2,3,4\}$

$A_{i,j} \leftarrow A_{i,j} \oplus B_{i,j}$ for $i \in \{1,2,3\}, j \in \{1,2,3,4\}$

/* rho east */

$A_{2,j} \leftarrow A_{2,j} \lll 1$ for $j \in \{1,2,3,4\}$

$A_{3,1}, A_{3,3} \leftarrow A_{3,3} \lll 8, A_{3,1} \lll 8$

$A_{3,2}, A_{3,4} \leftarrow A_{3,4} \lll 8, A_{3,2} \lll 8$

return $A_{i,j}$ *for* $i \in \{1,2,3\}, j \in \{1,2,3,4\}$

Algorithm 5.1: Definition of Xoodoo$[n_r]$.

5.2 Automated Search and Verification of Differential Trail using SMT

In order to compare the effectiveness of a functional distinguisher against a differential distinguisher, one has to avoid dealing with an invalid differential trail. While this seems to be unlikely for a keyed-permutation such as a block ciphers, this situation has been appearing in several previous works such as in [LIM20] or [SRB20] (see also references therein).

In CRYPTO 2020, Liu, Isobe, and Meier introduced a new MILP-based model to search for high probability differential trail [LIM20]. The novelty of their model consists in ensuring that the constructed differential trail is valid, i.e. there exists a pair of input that satisfies the trail. In a nutshell, the main idea is to construct two sets of constraints: one set defines the propagation of the differential on each round and the other defines the propagation of the input to the permutation. Both sets of constraints are then connected in the probabilistic step of the differential propagation, i.e. in the nonlinear function. This technique yields a valid differential trail for 6-round of Gimli [BKL+17] and at the same time it shows that the 12-round trail in [BKL+17] is invalid.

This work adapts a similar strategy as in [LIM20] to build an SMT model. To the best of our knowledge, this is the first time an SMT model is used to construct a valid differential trail for a symmetric-key primitive, more specifically for Xoodoo.

For the set of constraints that describes the Xoodoo permutation, we introduce Boolean variables in the input and output of each of the five (5) steps together with the Boolean constraints that represent the operations, which are relatively straightforward to derive from Algorithm 5.1. For the second set of constraints representing the propagation of

differences, we also introduce Boolean variables representing the input and output difference for all steps except for `iota`. The Boolean constraints that represent the propagation of the differences via the linear functions (`theta`, `rho west`, and `rho east`) are immediate to derive from their definitions. The only non-trivial part is to construct the constraints for the nonlinear step `chi` that not only represent the propagation of the differences, but also connects the value propagation as well as giving a straightforward way to compute the differential probability.

In order to derive the constraints, our approach is to first look at the coordinate functions of `chi` as a 3-bit permutation $\chi : \{0,1\}^3 \mapsto \{0,1\}^3$. The constraints that described the propagation of differences through χ are derived by examining all differentials on χ. The constraints that connect differential propagation and value propagation are immediate to derive from the algebraic expression of the first-order derivative of χ. We formally state it in the following theorem which can be proved exhaustively.

Theorem 5. *Let $\chi : \{0,1\}^3 \mapsto \{0,1\}^3$ be the function defined as $\chi(x_1, x_2, x_3) = (y_1, y_2, y_3)$ where*

$$\begin{aligned} y_1 &= x_1 \oplus (\neg x_2 \wedge x_3) \\ y_2 &= x_2 \oplus (\neg x_3 \wedge x_1) \\ y_3 &= x_3 \oplus (\neg x_1 \wedge x_2) \end{aligned}$$

For any differential $((\Delta X_1, \Delta X_2, \Delta X_3), (\Delta Y_1, \Delta Y_2, \Delta Y_3))$ on χ, it holds that

$$\begin{aligned} (\neg\Delta X_1 \wedge \neg\Delta X_2 \wedge \neg\Delta X_3) \wedge (\Delta Y_1 \vee \Delta Y_2 \vee \Delta Y_3) &= 0, \\ (\neg\Delta X_1 \wedge \neg\Delta X_2 \wedge \Delta X_3) \wedge \neg\Delta Y_3 &= 0, \\ (\neg\Delta X_1 \wedge \Delta X_2 \wedge \neg\Delta X_3) \wedge \neg\Delta Y_2 &= 0, \\ (\Delta X_1 \wedge \neg\Delta X_2 \wedge \neg\Delta X_3) \wedge \neg\Delta Y_1 &= 0, \\ (\neg\Delta X_1 \wedge \Delta X_2 \wedge \Delta X_3) \wedge \neg(\Delta Y_2 \oplus \Delta Y_3) &= 0, \\ (\Delta X_1 \wedge \neg\Delta X_2 \wedge \Delta X_3) \wedge \neg(\Delta Y_1 \oplus \Delta Y_3) &= 0, \\ (\Delta X_1 \wedge \Delta X_2 \wedge \neg\Delta X_3) \wedge \neg(\Delta Y_1 \oplus \Delta Y_2) &= 0, \\ (\Delta X_1 \wedge \Delta X_2 \wedge \Delta X_3) \wedge \neg(\Delta Y_1 \oplus \Delta Y_2 \oplus \Delta Y_3) &= 0, \\ ((x_1 \oplus \Delta X_1) \oplus (\neg(x_2 \oplus \Delta X_2) \wedge (x_3 \oplus \Delta X_3))) \oplus (y_1 \oplus \Delta Y_1) &= 0, \\ ((x_2 \oplus \Delta X_2) \oplus (\neg(x_3 \oplus \Delta X_3) \wedge (x_1 \oplus \Delta X_1))) \oplus (y_2 \oplus \Delta Y_2) &= 0, \\ ((x_3 \oplus \Delta X_3) \oplus (\neg(x_1 \oplus \Delta X_1) \wedge (x_2 \oplus \Delta X_2))) \oplus (y_3 \oplus \Delta Y_3) &= 0 \end{aligned}$$

and $\mathsf{DP}_\chi((\Delta X_1, \Delta X_2, \Delta X_3), (\Delta Y_1, \Delta Y_2, \Delta Y_3)) = 2^{-2\cdot(\Delta X_1 \vee \Delta X_2 \vee \Delta X_3)}$.

In order to compute the probability of the trail over multiple rounds, we assume that the rounds are independent. Based on our model, an SMT solver then searches for a valid differential trail from the user-specified probability. If there exists no trail with the given probability, the SMT solver simply states that the Boolean formula is unsatisfiable. Otherwise, the solver returns a differential trail together with an input to the permutation such that when the second input is constructed, the difference of the pair for each step of the permutation satisfies the differential trail.

There are two more usages of our SMT model. First, it can be used to check the validity of a differential trail. This is done by fixing the value of variables representing the differential trail and its probability. If the solver says the model is unsatisfiable, it implies that the provided trail is invalid, i.e. there exists no pair of input that satisfies the trail. The second usage of our model is to find the highest probability differential. This is achieved by searching for a trail with high probability and then iteratively reducing the probability until the SMT model is satisfiable. Thus our SMT model also serves as an

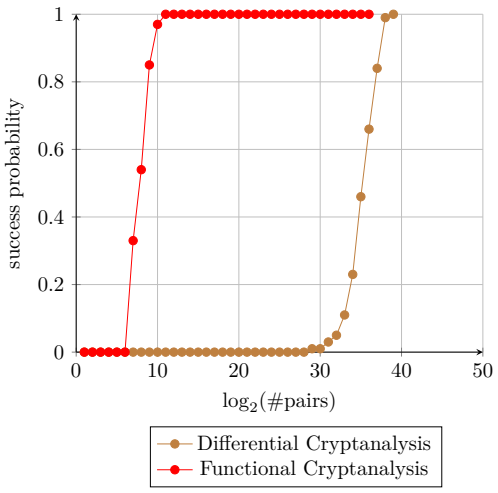

Figure 4: The success probability to mount the distinguishing attacks for 3-rounds of Xoodoo using differential and functional distinguisher with 100 trials.

alternative computational proof, in addition to the tree-search approach in [DHAK18a], on the bound of the best differential of Xoodoo. An example of the constructed valid differential trail is given in Appendix A.

5.3 Experimental Results

In order to show the advantage of a functional distinguisher in practice over a differential distinguisher, we implemented a distinguishing attack against three (3) rounds of Xoodoo. We use the functional described in Appendix D and compare it against the differential described in Appendix A. The distinguishing attack is repeated for 100 times and we measure the success probability over a different number of input pairs. The result is described in Figure 4.

Theoretically, our functional in this experiment has a probability of 2^{-8}, since we constructed it from the differential in Appendix A by replacing each translation-functional (differential) on 3-bit χ function in the last round to a deterministic functional. The list of the relevant deterministic functionals are given in Appendix B. This is certainly not the best functional that one could get for three rounds of Xoodoo but it helps to illustrate the usefulness and how to construct it in practice.

Since the cost to check the satisfiability of output pairs in the case of our functional is more costly than for differential, we also compare the timing to mount the distinguishing attack with the number of input pairs that has a comparable success probability. From Figure 4, we see that our functional distinguisher with 2^{11} input pairs and the differential distinguisher with 2^{39} input pairs have success probability approximately close to one. The implemented distinguishing attack for differential took 7 hours 48 minutes to finish while the implemented attack for functional was completed within milliseconds.[8]

6 Limitations of Functional Cryptanalysis

This section shall discuss the main limitation of functional cryptanalysis in the context of iterative permutations, specifically in the usage and construction of non-translation functionals.

[8]The comparison was done on a machine with Intel Xeon 8280 with 2.70 GHz processor using a single-thread.

A round function of an iterative permutation is generally constructed as a composition of a nonlinear map, a linear map, and a translation in $\mathbb{F}_2^n$. If we aimed to have a functional with probability one through a linear map and through a translation by a fix constant for all intermediate rounds, then we are restricted to use a translation-functional. Indeed, a translation-functional $(T_\alpha, T_{F(\alpha)})$ on an invertible $\mathbb{F}_2$-linear map $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ and (T_α, T_α) on any translation have probability one for any $\alpha \in \mathbb{F}_2^n$. Therefore, a generic strategy to construct a functional for an r-round iterative permutation is based on an $(r-2)$-round translation-functional (differential), which is then expanded in the first and the last round using non-translation functional.

6.1 Relation with Nonlinear Cryptanalysis

In [KR96] Knudsen and Robshaw introduced *nonlinear cryptanalysis* as a generalization of linear cryptanalysis [Mat93]. In order to distinguish $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ from a random permutation, linear cryptanalysis tries to find $c_i, d_i \in \mathbb{F}_2^n$ for $1 \leq i \leq n$ such that the following linear approximation

$$\sum_{i=1}^{n} c_i x_i = \sum_{i=1}^{n} d_i y_i$$

holds with a high absolute bias $|p - 1/2|$ where p is the probability of the approximation to hold. Such approximation is expected to have a probability $1/2$ for a random permutation. When F is an iterative permutation, the construction of a linear approximation for F is done by joining the compatible linear approximation for each round. Compatible means that the output approximation of round i of F acts as an input approximation for the round $i+1$ of F. The nontrivial part to find a high or low probability linear approximation for a single round is on its nonlinear function. When the nonlinear function is small, such as substitution boxes, one could generate the bias for all linear approximation of the nonlinear function known as *linear approximation table* [Mat93].

Nonlinear cryptanalysis generalizes this approach by using nonlinear approximations instead of the linear ones. This is a natural extension since there are many more nonlinear approximations than linear approximations. However, the construction of a nonlinear approximation for an iterative permutation has one challenge: the convenience of joining compatible linear approximations for each round function does not translate directly for the case of nonlinear approximations. The suggested approach to construct a nonlinear approximation is to find a linear approximation with a high absolute bias and replace the linear approximations in the outer round by nonlinear approximations.

In that respect, the construction of nonlinear approximations and non-translation functionals for an iterative permutation share an analogous approach. The former uses a linear trail with a high absolute bias and replaces the outer round linear approximations into a nonlinear one. The latter uses a differential (translation) trail with a high probability and replaces the outer round differential (translation-functional) into a non-translation one.

7 Conclusions and Future Work

In this work we have proposed functional cryptanalysis, a flexible and versatile cryptanalysis technique that generalizes differential, rotational, and rotational-xor cryptanalysis. We established the necessary notions together with the three primary aspects of the technique: the propagations, the representations, and the limitations. Functional cryptanalysis allows a unification of cryptanalysis techniques that rely on the notion of "difference", based on a binary operation, together with other cryptanalysis techniques where the notion of "difference" can not be defined. We proposed an algebraic framework to describe

functional cryptanalysis, which includes an algorithm based on Gröbner bases to compute the probability of a functional. The generic nature of functional cryptanalysis means that our algebraic framework is also applicable for differential, rotational, and rotational-xor cryptanalysis.

The high degree of flexibility of functional cryptanalysis allows it to express some functionals (σ, φ) where the cost of the evaluation map of σ and φ are non-negligible.[9] For instance, when both σ and φ are defined as F^{-1} applied on an r-round iterative permutation with identical round function F. However, the question whether the use of functionals with non-negligible evaluation cost constitute a meaningful distinguishing attack is a subject that deserves a dedicated discussion.

On the other hand, we believe that functional cryptanalysis has a potential to open up multiple new research questions. One question to ask is how the concept of functional translates to other variants of differential cryptanalysis such as truncated differential [Knu94], higher-order differential [Knu94], multiple-differential cryptanalysis [BG11], etc. Another possible future work is to explore other applications of functional ideal defined in Definition 20. Also the applications of functional cryptanalysis to other permutations or block ciphers, including other primitives such as hash functions, are still not covered in this paper. Other direction of work is how to address the limitations described in Section 6 and whether there exists a cipher where the construction of a functional is not limited to the strategy explained in the same section.

References

[AJN14] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. Analysis of NORX: investigating differential and rotational properties. In Diego F. Aranha and Alfred Menezes, editors, *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*, volume 8895 of *Lecture Notes in Computer Science*, pages 306–324. Springer, 2014.

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

[Ber] Daniel J. Bernstein. Chacha, a variant of salsa20. http://cr.yp.to/chacha/chacha-20080128.pdf.

[BFP09] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Mathematical Cryptology*, 3(3):177–197, 2009.

[BFS04] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equation. In *Proceedings of the International Conference on Polynomial System Solving*, pages 71–75, 2004.

[9] In the context of distinguishing attacks against an iterative permutation, we say that the cost of an evaluation map is non-negligible if its at least as expensive as the evaluation of a single round of the permutation. What motivates this definition is the fact that the complexity of distinguishing attack based on differential (resp. rotational-(xor)) does not take into account the cost of constructing differential pairs (resp. rotational-(xor) pairs). We argue that this is due to the cost of constructing each differential pair (resp. rotational-(xor) pair) is as costly as the translation layer (resp. affine layer) within a single round of the permutation. Also we speak loosely on the notion of "cost" here since one has a freedom to define it depending on the settings, such as when the attack is implemented in practice or simply treated as a theoretical result.

[BFSY05] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Eighth International Symposium on Effective Methods in Algebraic Geometry, Porto Conte Alghero, Sardinia (Italy)*, pages 1–14, 2005.

[BG11] Céline Blondeau and Benoît Gérard. Multiple differential cryptanalysis: Theory and practice. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2011.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.

[BS93] Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer, 1993.

[Buc06] Bruno Buchberger. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *J. Symb. Comput.*, 41(3-4):475–511, 2006.

[BWK93] Thomas Becker, Volker Weispfenning, and Heinz Kredel. *Gröbner bases - a computational approach to commutative algebra*, volume 141 of *Graduate texts in mathematics*. Springer, 1993.

[CBS19] Roberto Civino, Céline Blondeau, and Massimiliano Sala. Differential attacks: using alternative operations. *Des. Codes Cryptogr.*, 87(2-3):225–247, 2019.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer, fourth edition, 2015.

[CV94] Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 356–365. Springer, 1994.

[Dau05] Magnus Daum. *Cryptanalysis of Hash functions of the MD4-family*. PhD thesis, Ruhr University Bochum, 2005.

[DHAK18a] 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.

[DHAK18b] Joan Daemen, Seth Hoffert, Gilles Van Assche, and Ronny Van Keer. Xoodoo cookbook. *IACR Cryptol. ePrint Arch.*, 2018:767, 2018.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing gröbner bases (f4). *Journal of Pure and Applied Algebra*, 139(1):61–88, 1999.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero F5. In *International Symposium on Symbolic and Algebraic Computation Symposium - ISSAC 2002*, pages 75–83, Villeneuve d'Ascq, France, July 2002. ACM. Colloque avec actes et comité de lecture. internationale.

[HM97] Carlo Harpes and James L. Massey. Partitioning cryptanalysis. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 13–27. Springer, 1997.

[HO99] Philip Hawkes and Luke O'Connor. XOR and non-xor differential probabilities. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 272–285. Springer, 1999.

[KN10] Dmitry Khovratovich and Ivica Nikolic. Rotational cryptanalysis of ARX. In Seokhie Hong and Tetsu Iwata, editors, *Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers*, volume 6147 of *Lecture Notes in Computer Science*, pages 333–346. Springer, 2010.

[Knu94] Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.

[KR96] Lars R. Knudsen and Matthew J. B. Robshaw. Non-linear approximations in linear cryptanalysis. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 224–236. Springer, 1996.

[KR00] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 1*. Springer, 2000.

[KSW96] John Kelsey, Bruce Schneier, and David A. Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 237–251. Springer, 1996.

[KSW99] John Kelsey, Bruce Schneier, and David A. Wagner. Mod n cryptanalysis, with applications against RC5P and M6. In Lars R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 139–155. Springer, 1999.

[Lai94] Xuejia Lai. *Higher Order Derivatives and Differential Cryptanalysis*, pages 227–233. Springer US, Boston, MA, 1994.

[LIM20] Fukang Liu, Takanori Isobe, and Willi Meier. Automatic verification of differential characteristics: Application to reduced Gimli. In Daniele Micciancio

and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 219–248. Springer, 2020.

[LSL21] Yunwen Liu, Siwei Sun, and Chao Li. Rotational cryptanalysis from a differential-linear perspective: Practical distinguishers for round-reduced FRIET, Xoodoo, and Alzette. *IACR Cryptol. ePrint Arch.*, 2021:189, 2021.

[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 Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[MP13] Nicky Mouha and Bart Preneel. Towards finding optimal differential characteristics for ARX: Application to Salsa20. *IACR Cryptol. ePrint Arch.*, page 328, 2013.

[MS17] Rusydi H. Makarim and Marc Stevens. M4GB: an efficient gröbner-basis algorithm. In Michael A. Burr, Chee K. Yap, and Mohab Safey El Din, editors, *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2017, Kaiserslautern, Germany, July 25-28, 2017*, pages 293–300. ACM, 2017.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

[SPGQ06] François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. SEA: A scalable encryption algorithm for small embedded applications. In Josep Domingo-Ferrer, Joachim Posegga, and Daniel Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings*, volume 3928 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2006.

[SRB20] Sadegh Sadeghi, Vincent Rijmen, and Nasour Bagheri. Proposing an milp-based method for the experimental verification of difference trails. *IACR Cryptol. ePrint Arch.*, 2020:632, 2020.

[Vau96] Serge Vaudenay. An experiment on DES statistical cryptanalysis. In Li Gong and Jacques Stearn, editors, *CCS '96, Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, March 14-16, 1996*, pages 139–147. ACM, 1996.

[Wag04] David A. Wagner. Towards a unifying view of block cipher cryptanalysis. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised*

Papers, volume 3017 of *Lecture Notes in Computer Science*, pages 16–33. Springer, 2004.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

A Example of a Valid 3-Round Differential Trail for Xoodoo

Table 2: An example of valid input to Xoodoo for the differential trail in Table 3.

0x7FA5D96A	0x00982B9E	0xDF2A87F4	0x7DE2D5F2
0xB13D5A99	0xE7D4F65A	0x88EF158F	0x18F7CEEC
0x31F3D6E6	0xE9BD6412	0xBF28B377	0x6E22D231

Table 3: An example of a valid three (3) round differential trail of Xoodoo with theoretical probability 2^{-36}.

0x1D28B03E	0x09199081	0x46125265	0x56D31D2C	
0x5D28B03E	0x89199081	0x46125265	0x56D31D2C	
0x1D28B03E	0x89119081	0x46125265	0x56D31D2C	θ
0x00000000	0x80000000	0x00000000	0x00000000	
0x40000000	0x00000000	0x00000000	0x00000000	
0x00000000	0x00080000	0x00000000	0x00000000	ρ_{west}
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x40000000	0x00000000	0x00000000	
0x00000000	0x40000000	0x00000000	0x00000000	χ
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x40000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	ρ_{east}
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	θ
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	ρ_{west}
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x80000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	χ
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x80000000	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	ρ_{east}
0x00000000	0x80000000	0x00000000	0x00000000	
0x00000000	0x00000000	0x00000001	0x00000000	
0x00000000	0x00000000	0x00000000	0x00000000	θ
0x00000000	0x80000000	0x00002010	0x00004020	
0x00000000	0x00000000	0x00002011	0x00004020	
0x00000000	0x00000000	0x00002010	0x00004020	ρ_{west}
0x00000000	0x80000000	0x00002010	0x00004020	
0x00004020	0x00000000	0x00000000	0x00002011	
0x00000000	0x00000000	0x01008000	0x02010000	χ
0x00000000	0x80000000	0x01002010	0x02006031	
0x00004020	0x00000000	0x0000A010	0x00012011	
0x00004020	0x80000000	0x01008000	0x02014020	ρ_{east}
0x00000000	0x80000000	0x01002010	0x02006031	
0x00008040	0x00000000	0x00014020	0x00024022	
0x00800001	0x01402002	0x00402000	0x00000080	

B Deterministic Functional for 3-bit χ

Table 4: Deterministic functional for coordinate-wise 3-bit χ at the last round of Table 3

$\sigma(x_1, x_2, x_3) = (x'_1, x'_2, x'_3)$	$\varphi(y_1, y_2, y_3) = (y'_1, y'_2, y'_3)$
$x'_1 = x_1,$ $x'_2 = x_2 + 1,$ $x'_3 = x_3.$	$y'_1 = y_1y_2 + y_1 + y_2 + y_3,$ $y'_2 = y_2 + 1,$ $y'_3 = y_1 + y_2y_3 + 1.$
$x'_1 = x_1 + 1,$ $x'_2 = x_2,$ $x'_3 = x_3.$	$y'_1 = y_1 + 1,$ $y'_2 = y_1y_2 + y_3 + 1,$ $y'_3 = y_1y_3 + y_1 + y_2 + y_3.$
$x'_1 = x_1,$ $x'_2 = x_2,$ $x'_3 = x_3 + 1.$	$y'_1 = y_1y_3 + y_2 + 1,$ $y'_2 = y_1 + y_2y_3 + y_2 + y_3,$ $y'_3 = y_3 + 1.$

C Round Constants of Xoodoo

Table 5: The round constants c_r of Xoodoo with $-11 \leq r \leq 0$ in hexadecimal notation.

r	c_r	r	c_r	r	c_r	r	c_r
-11	0x00000058	-8	0x000000D0	-5	0x00000060	-2	0x000000F0
-10	0x00000038	-7	0x00000120	-4	0x0000002C	-1	0x000001A0
-9	0x000003C0	-6	0x00000014	-3	0x00000380	0	0x00000012

D Functional Distinguisher for 3-round of Xoodoo

The experimental result presented in Subsection 5.3 is based on the functional (T_α, φ) where α is the input difference of the differential trail in Appendix A and φ is the output function described in Table 6. The description of φ in Table 6 only shows the polynomial that is **not** of the form $y'_{i,j,k} = y_{i,j,k}$ where i, j, k denotes the k-th bit of the 32-bit word at row i and column j. We use 0-based indexing for the variables to make it easy to programmatically verify the result.

E Ideals for Other Cryptanalysis Techniques

In addition to the functional ideal defined in Section 4, one can also construct ideals for other cryptanalysis techniques. One examples is *differential ideal*.

Definition 21. Let $\mathcal{R}$ be a polynomial ring defined in (9). For any differential (α, β) on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, where $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_m)$, we define the differential ideal $\mathsf{DI}_F(\alpha, \beta)$ w.r.t the differential (α, β) as

$$\mathsf{DI}_F(\alpha, \beta) = \langle F, F', \{x_i + x'_i + \alpha_i \mid i = 1, \ldots, n\}, \{y_j + y'_j + \beta_j \mid j = 1, \ldots, m\}, \mathcal{P}(\mathcal{R})\rangle.$$

Clearly differential ideal is a special case of functional ideal and the result of Theorem 2 holds. This means that the Algorithm 4.1 can be used to compute the probability of a differential.

Another ideal of a particular interest is the one that can be used in linear cryptanalysis. We refer to this as *linear ideal*.

Definition 22. Let $\mathcal{R} = \mathbb{F}_2[x_1, \ldots, x_n, y_1, \ldots, y_m]$. For any $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ and $(b_1, \ldots, b_m) \in \mathbb{F}_2^m$, with $a, b \neq 0$, we define the linear ideal $\mathsf{LI}_F(a, b)$ of $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ with input mask a and output mask b as

$$\mathsf{LI}_F(a, b) = \langle F, \sum_{i=1}^{n} a_i x_i + \sum_{j=1}^{m} b_j y_j, \mathcal{P}(\mathcal{R}) \rangle.$$

Clearly $\mathsf{LI}_F(a, b)$ is zero-dimensional and radical since $\mathcal{P}(\mathcal{R}) \subset \mathsf{LI}_F(a, b)$. Hence, the following proposition holds.

Proposition 6. *Let $\mathcal{R} = \mathbb{F}_2[x_1, \ldots, x_n, y_1, \ldots, y_m]$. The bias of a linear approximation on $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ with input mask $a \in \mathbb{F}_2^n$ and output mask $b \in \mathbb{F}_2^m$ is equal to*

$$2^{-n} \cdot (\dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{LI}_F(a, b)) - 2^{n-1}).$$

By replacing $\mathsf{FI}_F(\sigma, \varphi)$ to $\mathsf{LI}_F(a, b)$ in line 2 and set the return value of Algorithm 4.1 to be $2^{-n} \cdot (|\mathsf{RM}(\mathsf{LI}_F(a, b))| - 2^{n-1})$, then we have an way to compute the bias of a linear approximation using Gröbner bases.

Also one can have an ideal for the autocorrelation of component function of $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, which is closely related with the differential-linear cryptanalysis.

Definition 23 (Autocorrelation)**.** Let $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$. The autocorrelation of a component function $F_b(x) = b \cdot F(x)$ at $\alpha \in \mathbb{F}_2^n$, where $\cdot$ here denotes a dot-product vector, is defined as $\tau_F(\alpha, b) = \sum_{x \in \mathbb{F}_2^n} (-1)^{F_b(x) + F_b(x+\alpha)}$.

Definition 24. Let $\mathcal{R}$ be a polynomial ring defined in (9). For any $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_2^n$ and $b = (b_1, \ldots, b_m) \in \mathbb{F}_2^m$, we define the *autocorrelation ideal* $\mathsf{AI}_F(\alpha, b)$ of F with input difference α and output mask b as

$$\mathsf{AI}_F(\alpha, b) = \langle F, F', \{x_i + x_i' + \alpha_i \mid i = 1, \ldots, n\}, \sum_{j=1}^{m} b_j (y_j + y_j') + 1, \mathcal{P}(\mathcal{R}) \rangle. \tag{12}$$

Note that the Hamming weight of $F_b(x) + F_b(x + \alpha)$ as an n-variable Boolean function is equal to the number of solutions of (12), which is equal to $\dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{AI}_F(\alpha, b))$ due the ideal being zero-dimensional and radical. Since for any n-variable Boolean function f we have $2^n - 2 \cdot \mathsf{wt}(f) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)}$ then $\tau_F(\alpha, b) = 2^n - 2 \cdot \dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{AI}_F(\alpha, b))$. Following the same approach as previously done, we can compute the autocorrelation $\tau_F(\alpha, b)$ by first computing a Gröbner basis of $\mathsf{AI}_F(\alpha, b)$ followed by the computation of $\dim_{\mathbb{F}_2}(\mathcal{R}/\mathsf{AI}_F(\alpha, b)) = |\mathsf{RM}(\mathsf{AI}_F(\alpha, b))|$ using the algorithm ReducedMonomials explained in Appendix F.

F Algorithm ReducedMonomials

Input: A Gröbner basis G of a zero-dimensional ideal $\{0\} \neq I \subseteq \mathbb{F}[x_1, \ldots, x_n]$
Output: The set $\mathsf{RM}(I)$ of reduced monomials of I

```
1  R ← {1}
2  for i = 1 to n do
3      M ← R
4      k_i ← max({e_i | x_1^{e_1} ··· x_i^{e_i} ··· x_n^{e_n} ∈ LM(G)})
5      while M ≠ {} do
6          Select a monomial m from M
7          M ← M \ {m}
8          for l = 1 to k_i do
9              m ← m · x_i
10             if LM(g) ∤ m, ∀g ∈ G then
11                 R ← R ∪ {m}
12 return R
```

Algorithm F.1: The algorithm ReducedMonomials(G) (adapted from RedTerms algorithm in [BWK93, pg. 424]).

Table 6: Output function φ for the functional used in Subsection 5.3.

(row, column)	
$(0,0)$	$y'_{0,0,5} = y_{0,0,5}y_{1,0,6} + y_{0,0,5} + y_{1,0,6} + y_{2,2,13}$, $y'_{0,0,14} = y_{0,0,14}y_{1,0,15} + y_{0,0,14} + y_{1,0,15} + y_{2,2,22}$.
$(0,1)$	$y'_{0,1,31} = y_{0,1,31} + 1$
$(0,2)$	$y'_{0,2,4} = y_{0,2,4} + 1$ $y'_{0,2,13} = y_{0,2,13} + 1$ $y'_{0,2,15} = y_{0,2,15}y_{2,0,23} + y_{1,2,16} + 1$, $y'_{0,2,24} = y_{0,2,24}y_{2,0,0} + y_{1,2,25} + 1$
$(0,3)$	$y'_{0,3,0} = y_{0,3,0}y_{1,3,1} + y_{0,3,0} + y_{1,3,1} + y_{2,1,8}$ $y'_{0,3,4} = y_{0,3,4}y_{1,3,5} + y_{0,3,4} + y_{1,3,5} + y_{2,1,12}$ $y'_{0,3,5} = y_{0,3,5} + 1$ $y'_{0,3,13} = y_{0,3,13}y_{1,3,14} + y_{0,3,13} + y_{1,3,14} + y_{2,1,21}$ $y'_{0,3,14} = y_{0,3,14} + 1$ $y'_{0,3,16} = y_{0,3,16}y_{2,1,24} + y_{1,3,17} + 1$ $y'_{0,3,25} = y_{0,3,25}y_{2,1,1} + y_{1,3,26} + 1$
$(1,0)$	$y'_{1,0,6} = y_{1,0,6} + 1$ $y'_{1,0,15} = y_{1,0,15} + 1$
$(1,1)$	$y'_{1,1,0} = y_{0,1,31}y_{1,1,0} + y_{2,3,7} + 1$
$(1,2)$	$y'_{1,2,5} = y_{0,2,4}y_{1,2,5} + y_{2,0,12} + 1$ $y'_{1,2,14} = y_{0,2,13}y_{1,2,14} + y_{2,0,21} + 1$ $y'_{1,2,16} = y_{0,2,15} + y_{1,2,16}y_{2,0,23} + y_{1,2,16} + y_{2,0,23}$ $y'_{1,2,25} = y_{0,2,24} + y_{1,2,25}y_{2,0,0} + y_{1,2,25} + y_{2,0,0}$
$(1,3)$	$y'_{1,3,1} = y_{1,3,1} + 1$ $y'_{1,3,5} = y_{1,3,5} + 1$ $y'_{1,3,6} = y_{0,3,5}y_{1,3,6} + y_{2,1,13} + 1$ $y'_{1,3,14} = y_{1,3,14} + 1$ $y'_{1,3,15} = y_{0,3,14}y_{1,3,15} + y_{2,1,22} + 1$ $y'_{1,3,17} = y_{0,3,16} + y_{1,3,17}y_{2,1,24} + x_{1,3,17} + y_{2,1,24}$ $y'_{1,3,26} = y_{0,3,25} + y_{1,3,26}y_{2,1,1} + y_{1,3,26} + y_{2,1,1}$
$(2,0)$	$y'_{2,0,0} = y_{2,0,0} + 1$ $y'_{2,0,12} = y_{0,2,4}y_{2,0,12} + y_{0,2,4} + y_{1,2,5} + y_{2,0,12}$ $y'_{2,0,21} = y_{0,2,13}y_{2,0,21} + y_{0,2,13} + y_{1,2,14} + y_{2,0,21}$ $y'_{2,0,23} = y_{2,0,23} + 1$
$(2,1)$	$y'_{2,1,1} = y_{2,1,1} + 1$ $y'_{2,1,8} = y_{0,3,0} + y_{1,3,1}y_{2,1,8} + 1$ $y'_{2,1,12} = y_{0,3,4} + y_{1,3,5}y_{2,1,12} + 1$ $y'_{2,1,13} = y_{0,3,5}y_{2,1,13} + y_{0,3,5} + y_{1,3,6} + y_{2,1,13}$ $y'_{2,1,21} = y_{0,3,13} + y_{1,3,14}y_{2,1,21} + 1$ $y'_{2,1,22} = y_{0,3,14}y_{2,1,22} + y_{0,3,14} + y + _{1,3,15} + y_{2,1,22}$ $y'_{2,1,24} = y_{2,1,24} + 1$
$(2,2)$	$y'_{2,2,13} = y_{0,0,5} + y_{1,0,6}y_{2,2,13} + 1$ $y'_{2,2,22} = y_{0,0,14} + y_{1,0,15}y_{2,2,22} + 1$
$(2,3)$	$y'_{2,3,7} = y_{0,1,31}y_{2,3,7} + y_{0,1,31} + y_{1,1,00} + y_{2,3,7}$