An Algebraic Attack on Ciphers with Low-Degree Round Functions: Application to Full MiMC

Maria Eichlseder[1], Lorenzo Grassi[1], Reinhard Lüftenegger[1], Morten Øygarden[2], Christian Rechberger[1], Markus Schofnegger[1], and Qingju Wang[3]

[1] IAIK, Graz University of Technology
[2] Simula UiB
[3] SnT, University of Luxembourg
firstname.lastname@iaik.tugraz.at
morten.oygarden@simula.no
qingju.wang@uni.lu

Abstract. Algebraically simple PRFs, ciphers, or cryptographic hash functions are becoming increasingly popular, for example due to their attractive properties for MPC and new proof systems (SNARKs, STARKs, among many others). In this paper, we focus on the algebraically simple construction MiMC which became an attractive cryptanalytic target due to its simplicity, but also due to its use as a baseline in an ongoing competition for more recent designs exploring this design space.
For the first time, we are able to describe key-recovery attacks on all full-round versions of MiMC over $\mathbb{F}_{2^n}$, requiring half the codebook. Recovering the key from this data for the n-bit version of MiMC takes the equivalent of less than $2^{n-\log_2(n)+1}$ calls to MiMC and negligible amounts of memory.
The attack procedure is a generalization of higher-order differential cryptanalysis, and it is based on two main ingredients: First, a zero-sum distinguisher which exploits the fact that the algebraic degree of MiMC grows much slower than originally believed. Second, an approach to turn the zero-sum distinguisher into a key-recovery attack without needing to guess the full subkey.
The attack has been practically verified on toy versions of MiMC. Note that our attack does not affect the security of MiMC over prime fields.

Keywords: Algebraic Attack, MiMC, Higher-Order Differential

1 Introduction

The design of symmetric cryptographic constructions exhibiting a clear and ideally low-degree algebraic structure is motivated by many recent use cases, for example the increasing popularity of new proof systems such as STARKs [9], SNARKs (e.g., Pinocchio [43]), Bulletproofs [20], and other concepts like secure multiparty computation (MPC). To guarantee good performance in these new applications, ciphers and hash functions are designed in order to minimize the multiplications (either the total number of multiplications, the depth, or other parameters related to the nonlinear operations). In contrast to traditional cipher design, the size of the field over which these constructions are defined has only a small impact on the final cost. In order to achieve this new performance goal, some crucial differences arise between these new designs and traditional ones. For example, we can consider the substitution (S-box) layer, that is, the operation providing nonlinearity in the permutation: In these new schemes, the S-boxes that compose this layer are relatively large compared to the ones used in classical schemes (e.g., they operate over 64 or 128 bits instead of 4 or 8 bits) and/or they can usually be described by a simple low-degree nonlinear function (e.g., $x \mapsto x^d$ for some d). Examples of these schemes include LowMC [4], MiMC [3], Jarvis/Friday [6], GMiMC [2], HadesMiMC [31], *Vision*/*Rescue* [5], and Starkad/Poseidon [30].

The structure of these schemes has a significant impact on the attacks that can be mounted. While statistical attacks (including linear [41] and differential [12] analysis) are among the most powerful attacks against traditional schemes, algebraic attacks turned out to be especially effective against these new primitives. In other words, these constructions are naturally more vulnerable to algebraic attacks than those which do not exhibit a clear and simple algebraic structure. For example, this has been shown in [1], which describes algebraic strategies covering the full-round versions of the attacked primitives. Although the approaches can be quite different, most of them exploit the low degree of the construction.

In this paper, we focus on MiMC [3]. The MiMC design constructs a cryptographic permutation by iterated cubing, interleaved with additions of random constants to break any symmetries. A secret key is added after every such round to obtain a block cipher. The design of MiMC is very flexible and can work with binary strings as well as integers modulo some prime p. Security analysis by the designers rules out various statistical attacks, and the final number of rounds is derived from an analysis of attack vectors that exploit the simple algebraic structure. We remark that the designers chose the number of rounds with a minimal security margin for efficiency. For a more detailed specification and a summary of previous analysis, we refer to Section 2.3.

Since its publication in 2016, MiMC has become the preferred choice for many use cases that benefit from a low multiplication count or algebraic simplicity [32,44]. It also serves as a baseline for various follow-up designs currently being evaluated in the context of the ongoing public "STARK-Friendly Hash Challenge" competition[4].

1.1 Our Contribution

As the main results, in this paper we present

(1) a new upper bound for the algebraic degree growth in key-alternating ciphers with low-degree round functions,
(2) a secret-key zero-sum distinguisher on almost full MiMC over $\mathbb{F}_{2^n}$,

[4] `https://starkware.co/hash-challenge/`

Table 1: Various attacks on MiMC. In this representation, n denotes the block size (and key size). The unit for the attack complexity is usually the cost of a single encryption (number of multiplications over $\mathbb{F}_{2^n}$ necessary for a single encyption). The memory complexity is negligible for all approaches listed.

Type	n	Rounds	Time	Data	Source
KR*	129	38	$2^{65.5}$	$2^{60.2}$	[40]
SK	129	80	2^{128} XOR	2^{128}	Section 4.1
SK	n	$\lceil \log_3(2^{n-1}-1)\rceil - 1$	2^{n-1} XOR	2^{n-1}	Section 4.1
KK	129	160 ($\approx 2 \times$ full)	–	2^{128}	Section 4.3
KK	n	$2 \cdot \lceil \log_3(2^{n-1}-1)\rceil - 2$	–	2^{n-1}	Section 4.3
KR	129	82 (full)	$2^{122.64}$	2^{128}	Section 5
KR	255	161 (full)	$2^{246.67}$	2^{254}	Section 5
KR	n	$\lceil n \cdot \log_3(2)\rceil$ (full)	$\leq 2^{n-\log_2(n)+1}$	2^{n-1}	Section 5

KR $\equiv$ Key-Recovery, KR* $\equiv$ attack on a variant of MiMC proposed in a low-memory scenario, SK $\equiv$ Secret-Key Distinguisher, KK $\equiv$ Known-Key Distinguisher

(3) a known-key zero-sum distinguisher on almost double the rounds of MiMC,
(4) the first key-recovery attack on *full-round* MiMC over $\mathbb{F}_{2^n}$.

We also show that the technique we use for MiMC is sufficiently generic to apply to any permutation fulfilling specific properties, which we will define in detail. Our attacks and distinguishers on MiMC, as well as other attacks in the literature, are listed in Table 1.

Secret-Key Zero-Sum Distinguishers. After recalling some preliminary facts about higher-order differentials, in Section 3, we analyze the growth of the algebraic degree for key-alternating ciphers whose round function can be described as a low-degree polynomial over $\mathbb{F}_{2^n}$.

For an SPN cipher over a field $\mathbb{F}$ where each round has algebraic degree δ, the algebraic degree of the cipher is expected to grow essentially exponentially in δ. Several analyses made in the literature [21,19,18] confirm this growth, except when the algebraic degree of the function is close to its maximum. As a result, the number of rounds necessary for security against higher-order differential attacks grows logarithmically in the size of $\mathbb{F}$.

In Section 3, we show that if the round function can be described as an invertible low-degree polynomial function in $\mathbb{F}_{2^n}$, then the algebraic degree grows linearly with the number of rounds, and not exponentially as generally expected. More precisely, let d denote the exponent of the power function $x \mapsto x^d$ used to define the S-boxes. Then, we show that in the case of key-alternating ciphers over $\mathbb{F}_{2^n}$, the algebraic degree $\delta(r)$ as a function in the number of rounds r is

$$\delta(r) \in \mathcal{O}(\log_2(d^r)) = \mathcal{O}(r).$$

As an immediate consequence, our observation implies that roughly $n \cdot \log_d(2)$ rounds are necessary to provide security against higher-order differential attacks, much more than the expected $\approx \log_\delta(n-1)$ rounds.

Distinguishers on MiMC over $\mathbb{F}_{2^n}$. Our new bounds on the number of rounds necessary to provide security against higher-order differential cryptanalysis have a major impact on all key-alternating ciphers with large S-boxes. A concrete example for this class of ciphers is MiMC [3], a key-alternating cipher defined over $\mathbb{F}_{2^n}$ (for odd $n \in \mathbb{N}$), where the round function is simply defined as the cube map $x \mapsto x^3$. Since any cubic function over $\mathbb{F}_{2^n}$ has algebraic degree 2, one may expect that approximately $\log_2(n)$ rounds are necessary to prevent higher-order differential attacks. Our new bound implies that a much larger number of rounds is required to provide security, namely approximately $n \cdot \log_3(2)$.

As a concrete example, in Section 4 we show that MiMC-n/n has a security margin of only 1 or 2 round(s) against (secret-key) zero-sum distinguishers (depending on n), which is much smaller than the one expected by the designers. Moreover, we can be set up a known-key distinguisher for approximately double the number of rounds of MiMC, by showing that the same number of rounds is necessary to reach the maximum degree in the decryption direction. We also remark that our findings have been practically verified on toy versions.

We remark that the designers presented other non-random properties (including GCD and interpolation attacks) that can cover a similar number of rounds. The number of rounds proposed by the designers were chosen in order to provide security against key-recovery attacks based on these properties. As we are going to show, the number of rounds is not sufficient against our new attack based on a higher-order differential property.

Results using the Division Property. For completeness, in Section 4.5 we search for zero-sum distinguishers for MiMC-n/n with the division property [46] proposed by Todo at Eurocrypt 2015, which is commonly believed to be the most powerful tool to find the best integral distinguishers for most block ciphers. By modeling the most recently proposed variant of the bit-based division property, which is called *modified three-subset bit-based division property* in [34], we are able to reproduce exactly the same zero-sum distinguishers for cases with small n-bit S-boxes, where $n \in \{5, 7, 9\}$. However, as far as we know, it is an open problem to model the (modified) three-subset bit-based division property for a larger S-box of size bigger than 8. Therefore, we conclude that the division property might not help us for the ciphers we focus on.

Key-Recovery Attack on MiMC-n/n and on Generic Ciphers. A trivial way to extend an r-round distinguisher to an $(r+1)$-round key-recovery attack is based on guessing the last round key, partially decrypting/encrypting, and finally exploiting the distinguisher to filter wrong key guesses. Unfortunately, this strategy does not work for MiMC, since guessing the full last round key is equivalent to exhaustive key search.

In Section 5, we show how to solve this problem. Instead of guessing the last round key, we set up an equation over $\mathbb{F}_{2^n}$ with the master key as a variable. To obtain this equation, we symbolically express the zero sum at the input to the last round as a polynomial function of the key, whose coefficients depend on the queried ciphertexts. We show how the resulting polynomial equation can be solved efficiently to recover the key. We outline a more detailed and generic procedure for our attack in Section 6. There, we also discuss the differences between our method and other related attacks present in the literature.

2 Preliminaries

In this section, we recall the most important results about polynomial representations of boolean functions and summarize the currently best known results regarding the growth of the algebraic degree in the context of SP networks. We also provide the specification of MiMC and give an overview of previous cryptanalytic results.

We emphasize that in general it is only possible to give a *lower* bound regarding the number of rounds that it is possible to attack using higher-order differential attacks, in the following denoted as "necessary number of rounds to provide security". While upper-bounding the algebraic degree is more important from an adversary's point of view, lower bounds on the degree are much more relevant when arguing about security from a designer's viewpoint. However, at the current state of the art and to the best of our knowledge, it seems hard to find such a lower bound for a given cipher without investigating concrete instances experimentally – which, of course, limits the scope of any analysis.

2.1 Polynomial Representations over Binary Extension Fields

We denote addition (and subtraction) in binary extension fields by the symbol $\oplus$. For $n \in \mathbb{N}$, every function $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ can be uniquely represented by an n-tuple $(F_1, F_2, \dots, F_n)$ of polynomials over $\mathbb{F}_2$ in n variables with a maximum degree of 1 in each variable. In this representation, F_i is of the form

$$F_i(X_1, \dots, X_n) = \bigoplus_{u=(u_1,\dots,u_n)\in\{0,1\}^n} \varphi_i(u) \cdot X_1^{u_1} \cdot \ldots \cdot X_n^{u_n}, \tag{1}$$

where the coefficients $\varphi_i(u)$ can be computed by the *Moebius transform*.

As is common, we denote functions $F : \mathbb{F}_2^n \to \mathbb{F}_2$ as *boolean functions* and functions of the form $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, for $n, m \in \mathbb{N}$, as *vectorial boolean functions*.

Definition 1. *The algebraic normal form (ANF) of a boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2$, as given in Eq. (1), is the unique representation as a polynomial over $\mathbb{F}_2$ in n variables and with a maximum univariate degree of 1. The algebraic degree $\delta(F)$ of F – or δ for simplicity – is the degree of the above representation of F as a multivariate polynomial over $\mathbb{F}_2$. If $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a vectorial boolean function and $(G_1, \dots, G_n)$ is its representation as an n-tuple of multivariate polynomials over $\mathbb{F}_2$, then its algebraic degree $\delta(G)$ is defined as $\delta(G) := \max_{1\le i\le n} \delta(G_i)$.*

The link between the algebraic degree and the univariate degree of a vectorial boolean function is well-known, and is for example established in [23]: the algebraic degree of $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ can be computed from its univariate polynomial representation, and is equal to the maximum hamming weight of the 2-ary expansion of its exponents.

Lemma 1. *Let $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ be a function and let $F(X) = \sum_{i=0}^{2^n-1} \varphi_i \cdot X^i$ denote the corresponding univariate polynomial description over $\mathbb{F}_{2^n}$. The algebraic degree $\delta(F)$ of F as a vectorial boolean function is the maximum hamming weight[5] of its exponents, i.e., it is $\delta(F) = \max_{0 \leq i \leq 2^n-1} \{\mathrm{hw}(i) \mid \varphi_i \neq 0\}$.*

2.2 Higher-Order Differential Cryptanalysis

Higher-order differential attacks [38] form a prominent class of attacks exploiting the low algebraic degree of a nonlinear transformation such as a classical block cipher. If this degree is sufficiently low, an attack using multiple input texts and their corresponding output texts can be mounted. In more detail, if the algebraic degree of a Boolean function f is δ, then, when applying f to all elements of an affine vector space $\mathcal{V} \oplus c$ of dimension greater than δ and taking the sum of these values, the result is 0, i.e., $\bigoplus_{v \in \mathcal{V} \oplus c} f(v) = 0$.

Security Against Higher-Order Differential Attacks – State of the Art. We focus on the case of *iterated block ciphers*, that is, ciphers consisting of several iterations of the same round function parameterized by different round keys. Let us assume the round function itself is of low algebraic degree. To prevent higher-order differential attacks, ideally one would like to have that after r rounds, there is no output bit and no vector subspace of $\mathbb{F}_2^n$ with dimension $d \leq n-1$ such that the d-th order derivative of the polynomial representation of this output bit with respect to this subspace is zero. To achieve this goal, one needs to estimate the growth of the algebraic degree. In other words, predicting the evolution of the algebraic degree of the cipher when the number of rounds varies is the main objective in higher-order differential cryptanalysis.

A trivial bound for the algebraic degree of the composition of two functions $F, G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is given by

$$\deg(F \circ G) \leq \deg(F) \cdot \deg(G). \tag{2}$$

This bound allows to estimate the minimum number of rounds necessary to reach the full degree. However, in general this upper bound does not reflect the real growth of the algebraic degree. For this reason, the problem of estimating this growth has been largely studied in the literature. After the initial work of Canteaut and Videau [21], a tighter upper bound was presented by Boura, Canteaut, and De Cannière [19] at FSE'11. In there, the authors show how to deduce a new bound for the algebraic degree of iterated permutations for a

[5] Given $x = \sum_{i=0}^{\chi} x_i \cdot 2^i$ for $x_i \in \{0, 1\}$, the hamming weight of x is $\mathrm{hw}(x) = \sum_{i=0}^{\chi} x_i$.

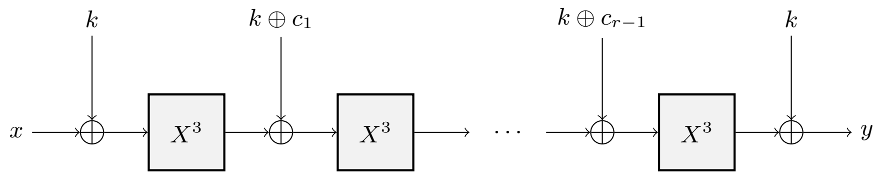

Fig. 1: The MiMC encryption function with r rounds.

special category of SP networks over $(\mathbb{F}_{2^n})^t$, which includes functions that have a number $t \geq 1$ of balanced S-boxes as their nonlinear layer. Specifically, the authors show that the algebraic degree of the considered SP network grows almost exponentially, except when it is close to its maximum.

Proposition 1 ([19]). *Let F be a function from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$ corresponding to the concatenation of t smaller S-boxes $S_1, ..., S_t$ defined over $\mathbb{F}_2^n$. Then, for any function G from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$, we have*

$$\deg(G \circ F(\cdot)) \leq \min\left\{\deg(F) \cdot \deg(G), N - \frac{N - \deg(G)}{\gamma}\right\}, \text{ where} \tag{3}$$

$$\gamma = \max_{i=1,\dots,n-1} \frac{n-i}{n-\delta_i} \leq n-1, \tag{4}$$

and where δ_i is the maximum degree of the product of any i coordinates of any of the smaller S-boxes.

Thus, the number of rounds necessary to prevent higher-order differential attacks is in general bigger than the one obtained using the trivial bound in Eq. (2). After this result, Boura and Canteaut [18] studied the influence of F^{-1} on the estimation of the algebraic degree of $\deg(F \circ G)$. This estimation turns out to be particularly useful for all ciphers where the nonlinear building blocks in the round function are *not* permutations (e.g., as is the case for DES).

2.3 Specification and Previous Analysis of MiMC

MiMC [3] is a key-alternating n-bit block cipher, where in each round the same n-bit key is added to the state. The nonlinear component of the construction is the evaluation of the cube function $f(x) = x^3$ over $\mathbb{F}_{2^n}$. Additionally, a different round constant is added in each round to break symmetries, where the first round constant is 0. The total number of rounds is then

$$r = \lceil n \cdot \log_3(2) \rceil ,$$

and we refer to Fig. 1 for a graphical representation of the encryption function.

MiMC is defined to work over prime fields and binary fields. In this paper, we focus on the binary field versions of MiMC[6], for which the block size n has to be odd in order for the S-box to be a permutation.

[6] Since the only subspaces of $\mathbb{F}_p$, where p is a prime number, are $\{0\}$ and $\mathbb{F}_p$ itself, our attack does not affect the security of MiMC over prime fields.

MiMC: Related Attacks in the Literature. The designers recommend MiMC with $\lceil n \cdot \log_3(2) \rceil$ rounds [3]. They derive this number of rounds by considering a variety of different key-recovery attacks on MiMC. According to their analysis, the most powerful attacks are interpolation [36] and GCD attacks. About higher-order differential attacks, the authors claim that "*the large number of rounds ensures that the algebraic degree of MiMC in its native field will be maximum or almost maximum. This naturally thwarts higher-order differential attacks* [...]".

The first attack on MiMC-n/n [40], presented at SAC 2019, targets a reduced-round version of MiMC proposed by the designers for a scenario in which the attacker has only limited memory, but it does not affect the security claims of full-round MiMC. The Feistel version of MiMC was attacked shortly after [17], by using generic properties of the used Feistel construction (instead of exploiting properties of the primitive itself). Finally, a specific attack on MiMC using Gröbner bases was considered in [1]. The authors state that by introducing a new intermediate variable in each round, the resulting multivariate system of equations is already a Gröbner basis and thus the first step of a Gröbner basis attack is for free. However, recovering univariate polynomials from this representation and then applying techniques like the GCD attack will result in a prohibitively large computational complexity, since the recovered polynomials will be of degree $\approx 3^r$ after r rounds. Hence, the authors conclude that MiMC cannot be attacked directly by using known Gröbner basis techniques.

3 Higher-Order Differentials of Key-Alternating Ciphers

Let us focus on a key-alternating cipher $E_k^r : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ defined as

$$E_k^r(x) := k_r \oplus R(\cdots R(k_1 \oplus R(k_0 \oplus x)) \cdots) \tag{5}$$

over $r \geq 1$ rounds, where $k_0, k_1, \ldots, k_r \in \mathbb{F}_{2^n}$ are derived from a master key $k \in \mathbb{F}_{2^n}$ using a certain key schedule, and where each round function $R : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is simply defined as some invertible (low-degree) polynomial function

$$R(x) := \rho_0 \oplus \bigoplus_{i=1}^{d} \rho_i \cdot x^i \tag{6}$$

of degree $d \geq 3$ and with $\rho_i \in \mathbb{F}_{2^n}$, $\rho_d \neq 0$. In the literature, a block cipher that falls into this category is, for example, MiMC.

Assumption On the Degree of R^{-1}. Let d_{inv} denote the degree of the inverse round function R^{-1}. For example, if $d = 3$ and $R(x) = x^3$, the inverse of the round function is $R^{-1}(x) = x^{\frac{2^{n+1}-1}{3}}$, and thus $d_{\text{inv}} = \frac{2^{n+1}-1}{3}$. In the entire paper we only consider round functions R which satisfy the condition $d \leq d_{\text{inv}}$. If this is not the case, an attacker would target the decryption function instead.

3.1 Growth of the Degree

In this section, we show that the algebraic degree δ of a key-alternating cipher E_k^r grows much slower than commonly presented in the literature (more precisely, in some cases it can grow linearly in the number of rounds and not exponentially).

Proposition 2. *Let R be the round function of a key-alternating cipher E_k^r with degree d defined as in Eq. (5). The number of rounds[7] $\mathcal{R}^{Linear}$ necessary to prevent a (secret-key) zero-sum distinguisher is given by*

$$\mathcal{R}^{Linear} = \left\lceil \log_d\left(2^{n-1}-1\right)\right\rceil \approx (n-1)\cdot \log_d(2). \tag{7}$$

Proof. To prevent higher-order differential attacks, we require the algebraic degree of E_k^r to reach its maximum value $n-1$. Due to the relation between the word-level degree and the algebraic degree, E_k^r has an algebraic degree of $n-1$ if at least one monomial with the exponent 2^n-2^j-1 (for $0 \le j < n$) appears in the polynomial representation. Indeed, note that all these monomials have an algebraic degree of $n-1$. Since the smallest exponent of this form is $2^n - 2^{n-1} - 1 = 2^{n-1}-1$, and since the degree of E_k^r after r rounds is at most d^r, we require that $d^r \ge 2^{n-1}-1$ to make $x^{2^{n-1}-1}$ appear, or equivalently

$$r \ge \lceil \log_d(2^{n-1}-1)\rceil. \qquad \square$$

A "Lower Bound" for the Growth of the Degree. We point out that it is always possible to set up a (secret-key) zero-sum distinguisher if the number of rounds is smaller than $\mathcal{R}^{\text{Linear}}$. However, a number of rounds greater than or equal to $\mathcal{R}^{\text{Linear}}$ does not necessarily provide security.

One of the main problems in order to derive a sufficient condition for the number of rounds that guarantees security is the difficulty of analyzing the non-vanishing coefficients in the polynomial representation of E_k^r. Note, in general it is not easy to give a condition guaranteeing that a particular monomial appears, since many factors (including the secret key, the constant addition, and the details of the S-box) influence this result.

Without going into the details, we consider the influence of the S-box in some concrete examples. Working with $R(x) = x^d$ for a certain $3 \le d \le 2^n-2$ (where $d \ne 2^{d'}$ for $d' \in \mathbb{N}$), we focus for simplicity only on two extreme cases[8]:

– If $d = 2^{d'}+1$ for some $d' \in \mathbb{N}$, then the output of a single round is sparse:

$$(x \oplus y)^{2^{d'}+1} = x^{2^{d'}+1} \oplus x^{2^{d'}} \cdot y \oplus y^{2^{d'}} \cdot x \oplus y^{2^{d'}+1}$$

(note that it contains only 4 terms instead of $d+1 = 2^{d'}+2$).

[7] We denote our results by $\mathcal{R}^{\text{Linear}}$ to indicate that the algebraic degree grows almost linearly.

[8] By Lucas's Theorem, $\binom{n}{m} \equiv \prod_{i=0}^{k} \binom{n_i}{m_i} \pmod 2$, where $n = \sum_{i=0}^{k} n_i \cdot 2^i$ and $m = \sum_{i=0}^{k} m_i \cdot 2^i$ is the 2-ary expansion of n and m, respectively.

– If $d = 2^{d'} - 1$ for some $d' \in \mathbb{N}$, then the output of a single round is full, since

$$(x \oplus y)^{2^{d'}-1} = \bigoplus_{i=0}^{2^{d'}-1} x^i \cdot y^{2^{d'}-1-i}.$$

Even if a single round is not sparse, the output of several combined rounds is not guaranteed to be full (even if it is in general dense). As a concrete example, while the output of $(x \oplus k_0)^3 \oplus k_1$ is full, the same is not true for

$$\begin{aligned}((x \oplus k_0)^3 \oplus k_1)^3 \oplus k_2 = x^9 \oplus x^8 \cdot k_0 \oplus x^6 \cdot k_1 \oplus x^4 \cdot k_0^2 \cdot k_1 \oplus x^3 \cdot k_1^2 \\ \oplus x^2 \cdot (k_0 \cdot k_1^2 \oplus k_0^2 \cdot k_1^2 \oplus k_0^4 \cdot k_1) \oplus x \cdot k_0^8 \oplus c(k_0, k_1, k_2),\end{aligned} \tag{8}$$

where both x^5 and x^7 are missing, and where $c(k_0, k_1, k_2)$ is a function that depends only on the keys. This simple example emphasizes the difficulty of analyzing the sparsity of the polynomial that defines E_k.

3.2 Comparison with Related Work in the Literature

Here we compare our number of rounds $\mathcal{R}^{\text{Linear}}$ necessary to guarantee security against secret-key zero-sum distinguishers with the one provided in [19] (and recalled in Proposition 1), denoted by $\mathcal{R}^{[\text{BCD11}]}$ in the following. We emphasize that our result is particularly relevant in the case in which the round function can be described as a *low-degree* polynomial function over $\mathbb{F}_{2^n}$.

Linear Growth versus Exponential Growth. The round numbers $\mathcal{R}^{\text{Linear}}$ and $\mathcal{R}^{[\text{BCD11}]}$ necessary to provide security are obtained by considerations about the growth of the algebraic degree. Here we analyze the upper bound of the degree growth of the cipher, denoted resp. by $\left(\delta^{\mathfrak{Linear}}\right)^r$ and by $\left(\delta^{[\text{BCD11}]}\right)^r$.

The number of rounds $\mathcal{R}^{[\text{BCD11}]}$ proposed in [19] is based on the assumption that the algebraic degree of the encryption/decryption function grows almost exponentially with the number of rounds, except when it is close to its maximum. Roughly speaking, an upper bound for the degree growth satisfies $\left(\delta^{[\text{BCD11}]}\right)^r \in \mathcal{O}(\delta^r)$, where δ is the algebraic degree of the round function over $\mathbb{F}_2^n$.[9]

If the round function can be described by a low-degree polynomial over $\mathbb{F}_{2^n}$, here we show for the first time that a better upper bound can be derived, i.e.,

$$\left(\delta^{\mathfrak{Linear}}\right)^r \leq \lfloor \log_2(d^r + 1) \rfloor \approx r \cdot \log_2(d) \in \mathcal{O}(r).$$

[9] For example, based on Eq. (3) where $\deg(G) = \delta^r$ and $\deg(F) = \delta$, it follows that the trivial bound (that is, the exponential growth) holds on the first r rounds where

$$\delta^{r+1} \leq n - \frac{n - \delta^r}{\gamma} \quad \rightarrow \quad \delta^r \leq n \cdot \frac{\gamma - 1}{\delta \cdot \gamma - 1} \approx \frac{n}{\delta}.$$

As a result, since $\left(\delta^{\mathfrak{Linear}}\right)^r \in \mathcal{O}(r)$ and since $\left(\delta^{[\text{BCD11}]}\right)^r \in \mathcal{O}(\delta^r)$, the round numbers $\mathcal{R}^{\text{Linear}}$ and $\mathcal{R}^{[\text{BCD11}]}$ *necessary* to provide security grow respectively linearly and logarithmically in the size n of the field, namely

$$\mathcal{R}^{\text{Linear}} \in \mathcal{O}(n) \qquad \text{and} \qquad \mathcal{R}^{[\text{BCD11}]} \in \mathcal{O}(\log_2(n)).$$

A concrete example of this will be given in the comparison in Section 4.2.

Remark. We emphasize that every (invertible) S-box/round function in $\mathbb{F}_2^n$ can be rewritten as a polynomial over $\mathbb{F}_{2^n}$. The crucial point here is that given a "random" S-box/round function over $\mathbb{F}_2^n$, its corresponding polynomial over $\mathbb{F}_{2^n}$ has in general a high degree (e.g., $d \approx 2^n - \varepsilon$ for some ε). In such a case, even if our argument still holds, the final result becomes meaningless, since $\log_d(2^n - 1) \approx \log_{2^n-\varepsilon}(2^n - 1) \approx 1$ is basically constant (i.e., it does not grow linearly with n). Hence, our results turn out to be relevant only for S-boxes/round functions for which the corresponding polynomial over $\mathbb{F}_{2^n}$ has "small" degree (namely, small compared to the field size, i.e., $d \ll 2^n$).

4 Distinguishers for Reduced-Round and Full MiMC

Exploiting the previous result, we now discuss the possibility to set up higher-order differential distinguishers and attacks on MiMC [3]. We show that

(1) MiMC has a security margin of only 1 or 2 round(s) against (secret-key) zero-sum distinguishers, depending on n, and that
(2) a zero-sum known-key distinguisher can be set up for approximately double the number of rounds of MiMC.

4.1 Secret-Key Zero-Sum Distinguisher for MiMC

The results just presented allow to set up a nontrivial (secret-key) zero-sum distinguisher on $\lceil \log_3(2^{n-1}-1)\rceil - 1$ rounds of MiMC, where $\lceil \log_3(2^{n-1}-1)\rceil - 1 < \lceil n \cdot \log_3(2)\rceil$ for all n. Consequently, the security margin is reduced to

$$1 \leq \lceil n \cdot \log_3(2)\rceil - \left(\lceil \log_3(2^{n-1} - 1)\rceil - 1\right) \leq 2$$

rounds. To give some concrete examples, MiMC has 1 round of security margin for $n \in \{33, 63, 255\}$, and 2 rounds of security margin for $n \in \{31, 65, 127, 129\}$.

4.2 Practical Results

In this section, we investigate how our results from Proposition 2 compares with $\mathcal{R}^{[\text{BCD11}]}$ and practical results for MiMC. The practical tests[10] have been performed in the following way: instead of computing the ANF of a keyed

[10] The source code for the attacks and the tests is available on `https://github.com/IAIK/mimc-analysis`.

permutation (which is expensive even for small field sizes), we evaluate the higher-order differential zero-sum property (as given in Section 2.2) for a specific input vector space. Namely, for random keys, random constants, and an input subspace of dimension $n-1$, we look for the minimum number of rounds r for which the corresponding sum of the ciphertexts is different from zero. Such a number corresponds to the number of rounds necessary to prevent zero-sum distinguishers. In order to avoid the influence of weak keys or round constants, we repeated the tests multiple times (with new random keys and round constants). The practical number of rounds we give in each row is *the smallest number of rounds among all tested keys and round constants necessary* to prevent zero-sum distinguishers. This means that a potentially higher number of rounds can be attacked by choosing the keys and round constants in a particular way.

Key-Alternating Ciphers with $\delta = 2$. In order to compare our theoretical results (namely, $\mathcal{R}^{\text{Linear}}$) with the ones already known in the literature (namely, $\mathcal{R}^{[\text{BCD11}]}$), we first provide a lower bound of $\mathcal{R}^{[BCD11]}$ – similar to the one provided in Eq. (3) – for this specific case (proof given in Appendix B).

Lemma 2. *Let $n \geq 3$. Under the assumption of Proposition 1, let S be an S-box on $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$ of algebraic degree 2, and let γ be defined as in Proposition 1. First of all, $\gamma \leq \frac{n+1}{2}$. Moreover, in the case in which the function $F : \mathbb{F}_2^N \to \mathbb{F}_2^N$ for $N = n \cdot t$ is the concatenation of t S-boxes just defined, then for any function G from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$*

$$\deg(G \circ F) \leq \min\left\{\deg(G) \cdot \deg(F), N - 2 \times \frac{N - \deg(G)}{n+1}\right\}. \tag{9}$$

By experiments and working on the cube S-box $S(x) = x^3$, we found that $\gamma = \frac{n+1}{2}$ for each odd $n \leq 33$. For this reason, we conjecture the following.

Conjecture 1. For the cube S-box $S(x) = x^3 : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, we conjecture that γ is always equal to $\frac{n+1}{2}$ for every (odd) n.

Results on MiMC. In Fig. 2 we compare the new bound $\mathcal{R}^{\text{Linear}}$ with the bound $\mathcal{R}^{[\text{BCD11}]}$ (using Eq. (9)), for MiMC-129/129 with a growing number of rounds. Where $\mathcal{R}^{[\text{BCD11}]}$ reaches the maximum algebraic degree after a short number of rounds, $\mathcal{R}^{\text{Linear}}$ needs almost the full number of rounds suggested. In Table 2, we investigate how many rounds we need to reach this maximum degree for MiMC when n is varied. Here we also include the practical number of rounds $\mathcal{R}$ necessary to prevent zero-sum distinguishers for MiMC-n/n. Referring to this table, we observe that $\mathcal{R}^{\text{Linear}}$ starts to be greater than $\mathcal{R}^{[\text{BCD11}]}$ for $n \geq 7$, and the gap is significant already for $n = 15$. Moreover, we emphasize that the practical results match the theoretical ones predicted by $\mathcal{R}^{\text{Linear}}$ in many cases, and are off by at most one round.

For completeness, we mention that the number of rounds $\mathcal{R}^{[\text{BCD11}]}$ given in Table 2 are based on the fact that a single round of MiMC has algebraic degree 2. However, we point out that better results can be derived by exploiting the fact

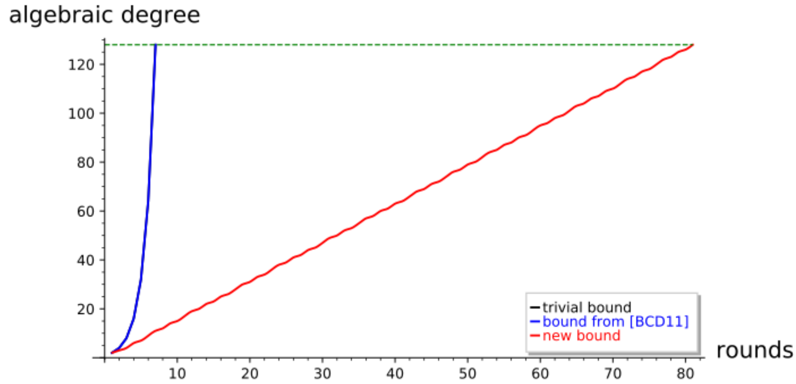

Fig. 2: Different *upper* bounds of the growth of the algebraic degree for MiMC-129/129. The trivial bound in black corresponds to the case $\delta(r) = 2^r$. A tighter bound is obtained by exploiting the result by Boura *et al.* at FSE'11 [19] in blue: the only difference regards the point $(7, 127)$ versus $(7, 128)$ of the previous case. Finally, our bound in red shows the growth of the algebraic degree (defined as $\log_2(3^r + 1)$) being linear. The green line denotes the minimum algebraic degree necessary to prevent higher-order differential attacks.

that two rounds of MiMC have algebraic degree at most 2, and not $2^2 = 4$ (see Eq. (8)). Using this observation, the number of rounds $\mathcal{R}^{[\text{BCD11}]}$ can be basically doubled w.r.t. the one given in Table 2. In any case, for a large S-box size n, the gap w.r.t. $\mathcal{R}^{\text{Linear}}$ predicted by our formula is significant.

4.3 Known-Key Zero-Sum Distinguisher for MiMC

A known-key distinguisher is a scenario introduced in [39] where the attacker knows the key, and it is important in all settings in which no secret material is present. To succeed, the attacker has to discover some property of the attacked cipher that holds with a probability higher than for an ideal cipher, or is believed to be hard to exhibit generically. The goal of a known-key *zero-sum* distinguisher is to find a set of plaintexts and ciphertexts whose sums are equal to zero. To do this, the idea is to exploit the inside-out approach. By choosing a subspace of texts $\mathcal{V}$, one simply defines the plaintexts as the r_{dec}-round decryption of $\mathcal{V}$ and the ciphertexts as the r_{enc}-round encryption of $\mathcal{V}$: such a distinguisher can cover $r_{\text{enc}} + r_{\text{dec}}$ rounds. Examples of this approach are given in the literature for Keccak [19,7,11], Luffa [19,7], or PHOTON [50].

In the case of MiMC, the idea is to choose $\mathcal{V}$ as a subspace of $\mathbb{F}_{2^n}$ of dimension $n-1$. The maximum number of encryption rounds r_{enc} for which it is possible to guarantee a zero-sum has been given in the previous paragraph. Based on Section 4.2, we can set up a known-key distinguisher on (more than) full MiMC-n/n. For our distinguisher on MiMC, we first recall the following result from [18].

Table 2: Theoretical and practical round numbers *necessary* to prevent zero-sum distinguishers for MiMC (a key-alternating cipher where the round function is just the cube function) over $\mathbb{F}_{2^n}$. We assume $\gamma = (n+1)/2$ for $\mathcal{R}^{[\text{BCD11}]}$.

Param.	Theoretical		Practical
n	$\mathcal{R}^{\text{Linear}}$	$\mathcal{R}^{[\text{BCD11}]}$ (based on Eq. (9))	Practical $\mathcal{R}$
5	3	3	4
7	4	3	5
9	6	4	6
11	7	4	7
13	8	4	9
15	9	4	10
17	11	5	11
33	21	6	21
65	41	7	-
129	81	8	-
257	162	9	-

Proposition 3 (Corollary 3 of [18]). *Let F be a permutation of $\mathbb{F}_2^n$. Then,* $\deg(F^{-1}) = n-1$ *if and only if* $\deg(F) = n-1$.

Corollary 1. *Let r_{enc} be the number of rounds necessary for MiMC over $\mathbb{F}_{2^n}$ to reach its maximum algebraic degree in the encryption direction. The same number of rounds is necessary for reaching the maximum algebraic degree in the decryption direction, i.e.,* $r_{dec} = r_{enc} = \lceil \log_3(2^{n-1}-1) \rceil$.

It follows that, given a subspace $\mathcal{V} \subseteq \mathbb{F}_{2^n}$ of dimension $n-1$, the sums of the corresponding texts after r_{dec} decryption rounds and r_{enc} encryption rounds are always equal to zero, i.e.,

$$\underbrace{\sum_{w \in \mathcal{V} \oplus v} R^{-(r_{\text{dec}}-1)}(w) = 0}_{\text{Zero sum}} \xleftarrow{R^{-(r_{\text{dec}}-1)}} \mathcal{V} \oplus v \xrightarrow{R^{r_{\text{enc}}-1}} \underbrace{0 = \sum_{w \in \mathcal{V} \oplus v} R^{r_{\text{enc}}-1}(w)}_{\text{Zero sum}}$$

for each $v \in \mathbb{F}_{2^n}$. Hence, a known-key zero-sum distinguisher can be set up for

$$\begin{aligned} 2 \cdot (\lceil \log_3(2^{n-1}-1) \rceil - 1) &\approx 2(n-1) \cdot \log_3(2) - 2 = \\ &= \underbrace{n \cdot \log_3(2)}_{= \text{ full MiMC}} + [(n-2) \cdot \log_3(2) - 2] \end{aligned}$$

rounds of MiMC-n/n, which is much more than *full* MiMC-n/n.

4.4 Impact of the Known-Key Distinguisher on Full MiMC

Sponge Function. In [3], the authors propose a hash function by instantiating a sponge construction with MiMC$^\pi$, a fixed-key version of MiMC. The sponge hash function is indifferentiable from a random oracle up to $2^{c/2}$ calls to the internal permutation P (where c is the capacity) if P is modeled as a randomly chosen permutation [10]. Thus, even if it is not strictly necessary, it is desirable that MiMC is resistant against known-key distinguishers.

For completeness, we mention that even if there is a way to distinguish a permutation from a random one, it seems difficult to exploit a zero-sum distinguisher of the internal permutation of a sponge construction in order to attack the hash function. To give a concrete example, consider the case of Keccak: As a consequence of the zero-sum distinguisher found on 18-round Keccak-f[1600], the number of rounds has been increased from 18 to 24 in the second round of the SHA-3 competition in order to avoid "non-ideal" properties (see [19,11] for more details). However, the best known attack on the Keccak hash function can only be set up when using 6-round Keccak-f [33].

In any case, we remark that such distinguishers based on zero sums cannot be set up for an arbitrary number of rounds, and they do indeed exploit the internal properties of a primitive using the inside-out approach found in this paper and in other literature. Hence, they cannot be considered meaningless.

Other Use Cases. Even though the original MiMC paper only specifies a sponge-based hash function using MiMC, there are various application-specific considerations that would make a block-cipher-based approach more advantageous (like, for example, being forced to use a block size which is too small for a sponge-based approach). Another way to turn a block cipher into a hash function is to use a compression function like the Davies–Meyer one together with something like the Merkle–Damgård construction. Similar to the case of sponge constructions, the security of such an algorithm is proven in the ideal cipher model [13]. This choice is, however, not supported by the MiMC designers, who use our results to support their advice against using a block-cipher-based approach (even though such implementations can still be found[11]).

In conclusion, since the attacker has control of the key in such scenarios, it is desirable for MiMC to be resistant against known- and chosen-key distinguishers, even if it does not seem to be strictly necessary.

4.5 Results Using the Division Property

Finally, in Appendix C we also present our practical results obtained using "Mixed Integer Linear Programming (MILP)", which models the propagation of the (conventional) bit-based division property.

The bit-based division property [48] was proposed to investigate integral characteristics of block ciphers at a bit level. With this approach [48], the integral property of each bit is studied independently. Naturally, this strategy allows to capture more information of the propagation than the word-level one, and thus integral characteristics for more rounds can be found with this new technique.

[11] `https://github.com/HarryR/ethsnarks/blob/master/src/gadgets/mimc.hpp`

For example, the integral distinguishers of SIMON32 have been improved from 10 rounds [46] (the current best result at word level) to 14 rounds [52] (obtained by the experimental method cited before).

Instead of separating the parity into the two cases "0" and "unknown" as for the (conventional) bit-based division property, three-subset bit-based division property [48] was introduced to enhance the accuracy of the conventional one, where the parity is separated into three sets, i.e., "0", "1", and "unknown". It shows that the three-subset bit-based division property can indeed be more accurate than the two-subset bit-based division property for some ciphers [35,53]. However, it becomes harder to efficiently model the three-subset division property propagation even for ciphers with simple structures. Recently, [34] pointed out that the three-subset division property has a couple of known problems when applied to cube attacks, and proposed a modified three-subset bit-based division without the "unknown" set to overcome these problems. By modeling this modified version of the bit-based property for our cases with small n-bit S-boxes, where $n \in \{5, 7, 8\}$, we can confirm the practical results given in Table 2.

However, as far as we know, it is still an open problem to model the (modified) three-subset bit-based division property for a larger S-box of size bigger than 8. The S-boxes we focus on in this paper can be described as a (low-degree) polynomial function in $\mathbb{F}_{2^n}$, where n is much larger than 8. Therefore, *the division property, which is commonly believed as the most efficient tool to find the best integral distinguishers, might not help us as much for the ciphers we focus on.*

5 Key-Recovery Attack on MiMC

Since the security margin of MiMC with respect to a (secret-key) zero-sum distinguisher is of only 1 or 2 round(s) depending on n, it is potentially possible to extend a distinguisher to a key-recovery attack. Given a subspace $\mathcal{V}$ of plaintexts whose sum is equal to zero after r rounds, we can consider $r+1$ rounds, partially guess the last subkey and decrypt, and filter wrong key guesses that do not satisfy the zero sum:

$$\mathcal{V} \oplus v \xrightarrow{R^r(\cdot)} \underbrace{\sum_{w \in \mathcal{V} \oplus v} R^r(w) = 0}_{\text{Zero sum}} \xleftarrow[\text{Key guessing}]{R^{-1}(\cdot)} \underbrace{\{R^{r+1}(w) \mid w \in \mathcal{V} \oplus v\}}_{\text{Ciphertexts}}.$$

However, since the subkeys of MiMC are equal to the master key plus constants and due to the single full-state S-box, even a (partial) decryption of a single round requires guessing the full key. As a result, a key-recovery attack on full MiMC based on this strategy seems infeasible.

In this section, we present an alternative strategy that allows to break full-round MiMC. Since a trivial key-guessing approach is inefficient, our idea is to construct a polynomial of low degree, which we can then try to solve.

5.1 Strategy of the Attack

From Proposition 2 and Proposition 3, a zero sum can be set up for at least $\lceil (n-1)\log_3(2) \rceil - 1 = \lceil n \log_3(2) \rceil - \varepsilon$ rounds in the encryption and decryption direction with a vector space $\mathcal{V} \oplus v$ of dimension $n-1$, where $\varepsilon \in \{1, 2\}$. Recalling that $\lceil n \cdot \log_3(2) \rceil$ is the number of rounds of full MiMC, we define r_{KR}, r_{ZS} as

$$r_{\text{ZS}} = \lceil (n-1) \log_3(2) \rceil - 1 \qquad \text{and} \qquad r_{\text{KR}} = 1 + (\lceil n \log_3(2) \rceil - \lceil (n-1) \log_3(2) \rceil),$$

where r_{ZS} is the number of rounds that we can cover with a zero sum, $r_{\text{KR}} = \lceil n \cdot \log_3(2) \rceil - r_{\text{ZS}} \in \{1, 2\}$.

Let $f^r(x, K)$ be the function corresponding to r rounds of $\text{MiMC}_k(\cdot)$ (and $f^{-r}(x, K)$ be r rounds of decryption, $\text{MiMC}_k^{-1}(\cdot)$), where x is the input text and K is a symbolic variable that represents the secret key k. We intend to use these functions to create a polynomial from which we can deduce k. More precisely, for a fixed vector space $\mathcal{V} \oplus v$, we consider the equations:

$$\underbrace{\bigoplus_{x \in \text{MiMC}_k^{-1}(\mathcal{V} \oplus v)} f^{r_{\text{KR}}}(x, K)}_{= F(K)} = 0 \quad \text{and} \quad \underbrace{\bigoplus_{x \in \text{MiMC}_k(\mathcal{V} \oplus v)} f^{-r_{\text{KR}}}(x, K)}_{= G(K)} = 0. \tag{10}$$

After having received all x values from an oracle, the attacker can construct one of the polynomials $F(K) = 0$ or $G(K) = 0$. The secret key k can now be determined by finding the roots of either of these polynomials.

In the case of MiMC, the degree of a single encryption round is 3, while the degree of a single decryption round is $(2^{n+1} - 1)/3$ (which is significantly larger than 3 for large n). Due to the low degree growth in the encryption direction of MiMC, we will focus on finding the roots of $F(K)$ given in Eq. (10).

Finding the Roots of Univariate Polynomials. Let $F(X) \in \mathbb{F}_{2^n}[X]/\langle X^{2^n} + X \rangle$ be a univariate polynomial of degree D. Furthermore, let $M(D)$ denote a number such that multiplying two polynomials of degree $\leq D$ over $\mathbb{F}_{2^n}$ requires $\mathcal{O}(M(D))$ operations in $\mathbb{F}_{2^n}$. For instance, a straightforward method would yield $M(D) = D^2$, whereas $M(D) = D \cdot \log(D) \cdot \log\log(D)$ holds for methods based on Fast Fourier Transforms [22]. The *Berlekamp algorithm* for determining the roots of F is then expected to require $\mathcal{C} \in \mathcal{O}\left(M(D) \log(D) \log\left(2^n D\right)\right)$ operations in $\mathbb{F}_{2^n}$ (see [29, Chapter 14.5]).

5.2 Details of the Attack

Assume $\mathcal{V} \oplus v$ is a coset of a subspace $\mathcal{V}$ of dimension $n - 1$. We define

$$\mathcal{W} = \text{MiMC}_k^{-1}(\mathcal{V} \oplus v) \equiv \{\text{MiMC}_k^{-1}(x) \in \mathbb{F}_{2^n} \mid \forall x \in \mathcal{V} \oplus v\}$$

under a fixed secret key k. Here, we present the details of the attack for the cases $r_{\text{KR}} = 1$ and $r_{\text{KR}} = 2$, and we analyze the computational cost. We introduce the following notation:

$$\forall d \in \mathbb{N}: \qquad \mathscr{P}_d := \bigoplus_{x \in \mathcal{W}} x^d, \tag{11}$$

Algorithm 1: Attack on MiMC – Case: $r_{\mathrm{KR}} = 1$.

Input: Vector subspace $\mathcal{V}$ of ciphertexts of dimension $\dim(\mathcal{V}) = n - 1$.
Output: Secret key k.

$\mathscr{P}_1, \mathscr{P}_2, \mathscr{P}_3 \leftarrow 0$.
for $x \in \mathcal{V} \oplus v$ **do**
  $p \leftarrow \mathrm{MiMC}_k^{-1}(x)$ from the decryption oracle.
  $\mathscr{P}_1 \leftarrow \mathscr{P}_1 \oplus p$.
  $q \leftarrow p^2$.
  $\mathscr{P}_3 \leftarrow \mathscr{P}_3 \oplus q \cdot p$.
$\mathscr{P}_2 \leftarrow (\mathscr{P}_1)^2$.
$F(K) = \mathscr{P}_1 \cdot K^2 \oplus \mathscr{P}_2 \cdot K \oplus \mathscr{P}_3$.
Find a solution k of $F(K) = 0$ – see Section 5.1 (filter multiple solutions by brute force).
return k.

and whenever possible we will make use of the fact that squaring is a linear operation over $\mathbb{F}_{2^n}$. More specifically, computing $\mathscr{P}_{2d}$ only requires a single squaring operation once $\mathscr{P}_d$ is calculated:

$$\mathscr{P}_{2d} := \bigoplus_{x \in \mathcal{W}} x^{2d} = \left(\bigoplus_{x \in \mathcal{W}} x^d \right)^2 = \mathscr{P}_d^2. \tag{12}$$

This allows to reduce the total number of `XOR` operations.

Case: $\boldsymbol{r_{\mathrm{KR}} = 1}$. Since a single round of MiMC is described by $(x \oplus k)^3 = k^3 \oplus k^2 \cdot x \oplus k \cdot x^2 \oplus x^3$, the function $F(K)$ is given by

$$F(K) = K^2 \cdot \mathscr{P}_1 \oplus K \cdot \mathscr{P}_2 \oplus \mathscr{P}_3.$$

A complete pseudo code of the attack can be found in Algorithm 1, which makes it easy to see that the cost of the attack is well approximated by

- $|\mathcal{V}| = 2^{n-1}$ multiplications,
- $|\mathcal{V}| = 2^{n-1} + 1$ squarings,
- $2 \cdot |\mathcal{V}| + 1 = 2^n + 1$ n-bit `XOR` operations,
- cost of finding a solution of an univariate polynomial equation of degree 2.

Case: $\boldsymbol{r_{\mathrm{KR}} = 2}$. The attack for the case $r_{\mathrm{KR}} = 2$ is similar. From Eq. (8) (using $k_0 = k$, $k_1 = k \oplus c_1$ and $k_2 = 0$), the function $F(K)$ is described by

$$\begin{aligned} F(K) &= K^8 \cdot \mathscr{P}_1 \oplus K^5 \cdot \mathscr{P}_2 \oplus K^4 \cdot (\mathscr{P}_2 \cdot c_1 \oplus \mathscr{P}_1) \oplus K^3 \cdot (\mathscr{P}_4 \oplus \mathscr{P}_2) \\ &\oplus K^2 \cdot (\mathscr{P}_4 \cdot c_1 \oplus \mathscr{P}_3 \oplus \mathscr{P}_1 \cdot c_1^2) \oplus K \cdot (\mathscr{P}_8 \oplus \mathscr{P}_6 \oplus \mathscr{P}_2 \cdot c_1^2) \oplus (\mathscr{P}_9 \oplus \mathscr{P}_6 \cdot c_1 \oplus \mathscr{P}_3 \cdot c_1^2), \end{aligned}$$

where c_1 is the round constant of the first round. As also noted in Section 3.1; while $\mathscr{P}_9$ is the largest $\mathscr{P}_d$ in this expression, both $\mathscr{P}_5$ and $\mathscr{P}_7$ are missing, and

Algorithm 2: Attack on MiMC – Case: $r_{\text{KR}} = 2$.

Input: Vector subspace $\mathcal{V}$ of ciphertexts of dimension $\dim(\mathcal{V}) = n - 1$.
Output: Secret key k.

$\mathscr{P}_1, \mathscr{P}_2, \mathscr{P}_3, ..., \mathscr{P}_9 \leftarrow 0$.
for $x \in \mathcal{V} \oplus v$ **do**
  $p \leftarrow \text{MiMC}_k^{-1}(x)$ from the decryption oracle.
  $\mathscr{P}_1 \leftarrow \mathscr{P}_1 \oplus p$.
  $q_2 \leftarrow p^2$.
  $q_3 \leftarrow q_2 \cdot p$.
  $\mathscr{P}_3 \leftarrow \mathscr{P}_3 \oplus q_3$.
  $q_6 \leftarrow q_3^2$.
  $\mathscr{P}_9 \leftarrow \mathscr{P}_9 \oplus q_6 \cdot q_3$.
$\mathscr{P}_2 \leftarrow (\mathscr{P}_1)^2$.
$\mathscr{P}_4 \leftarrow (\mathscr{P}_2)^2$.
$\mathscr{P}_6 \leftarrow (\mathscr{P}_3)^2$.
$\mathscr{P}_8 \leftarrow (\mathscr{P}_4)^2$.
$F(K) = K^8 \cdot \mathscr{P}_1 \oplus K^5 \cdot \mathscr{P}_2 \oplus K^4 \cdot (\mathscr{P}_2 \cdot c_1 \oplus \mathscr{P}_1) \oplus K^3 \cdot (\mathscr{P}_4 \oplus \mathscr{P}_2) \oplus K^2 \cdot (\mathscr{P}_4 \cdot c_1 \oplus \mathscr{P}_3 \oplus \mathscr{P}_1 \cdot c_1^2) \oplus K \cdot (\mathscr{P}_8 \oplus \mathscr{P}_6 \oplus \mathscr{P}_2 \cdot c_1^2) \oplus (\mathscr{P}_9 \oplus \mathscr{P}_6 \cdot c_1 \oplus \mathscr{P}_3 \cdot c_1^2)$.
Find a solution k of $F(K) = 0$ (filter multiple solutions by brute force).
return k.

hence do not need to be computed. A complete pseudo code of the attack can be found in Algorithm 2. Again, it is easy to see that the cost of the attack is well approximated by

- $2 \cdot |\mathcal{V}| + 6 = 2^n + 6$ multiplications,
- $2 \cdot |\mathcal{V}| + 4 = 2^n + 4$ squarings,
- $3 \cdot |\mathcal{V}| + 8 = 3 \cdot 2^{n-1} + 8$ n-bit `XOR` operations,
- cost of finding a solution to an univariate polynomial equation of degree 8.

5.3 Complexity Estimation

As we have just seen, our attack requires half of the codebook (namely, 2^{n-1} chosen ciphertexts). Here we show that our attacks are better than brute force (from the computational point of view). In order to do this, we measure the time complexities in equivalent encryption operations.

A single encryption round in MiMC requires one addition, one squaring operation, and one multiplication in the extension field. Since the cost of a single n-bit `XOR` operation is much smaller than the cost of a multiplication over $\mathbb{F}_{2^n}$, and since the number of `XOR` operations is similar to the number of multiplications, in the following we do not consider `XOR` operations. After this simplification, we find that the time complexity of $r_{\text{KR}} = 1$ is dominated by 2^{n-1} squaring and multiplication operations or, equivalently, 2^{n-1} encryption rounds. A similar line of reasoning reveals that $r_{\text{KR}} = 2$ is comparable to 2^n encryption rounds.

Since the cost of solving a single low-degree equation is negligible, and one unit of encryption contains $\lceil n \cdot \log_3(2) \rceil$ rounds, it follows that the cost of our attacks is about

$$\frac{r_{\text{KR}} \cdot 2^{n-1}}{\lceil n \cdot \log_3(2) \rceil} \text{ encryptions}$$

for $r_{\text{KR}} \in \{1, 2\}$. That is, the computational cost of our attacks is upper-bounded by $2^{n-\log_2(n)+1}$, and so smaller than the cost of a brute-force attack (namely, 2^n encryptions) for each $n \geq 3$.

5.4 Practical Verification

We implemented Algorithm 1 and Algorithm 2 in the computer algebra system Magma, and verified both algorithms for all odd integers $n \in [5, 33]$. We note that Algorithm 1 ($r_{\text{KR}} = 1$) yields the correct answer for all the tested $5 \leq n \leq 33$, even if $\lceil n \log_3(2) \rceil \neq \lceil (n-1) \log_3(2) \rceil$. Namely, in practice it is possible to cover one more round with a zero sum than what we theoretically expect. In other words, $\lceil (n-1) \log_3(2) \rceil$ rounds of the decryption function of MiMC fail to obtain the maximum algebraic degree for these parameters, which is reached after $\lceil (n-1) \log_3(2) \rceil + 1$ rounds (c.f. Appendix A for more details on the degree growth of MiMC^{-1}). Since we are not able to prove this behavior for larger values of n, we leave it as an open question whether Algorithm 1 can be applied to MiMC for odd integers $n > 33$.

Considerations on Data and Computational Costs of this Attack. A possible drawback of our attack is obviously the cost. Since we are not able to provide an estimation of the growth of the degree in the decryption direction, we can only exploit the fact that a certain number of rounds are necessary in order to achieve maximum degree. It follows that the attacker is forced to use half of the code book in order to set up the attack, which has also an impact on the computational cost.

Even if our attack is not practical, we believe it provides valuable theoretical insight. It is also in line with several other attacks found in the literature, that are set up under a similar assumption on the data and/or computational cost. To give some concrete examples, consider the case of zero-correlation attacks [15], which exploit linear approximations that hold with probability $\frac{1}{2}$. The crucial limitation for basic zero-correlation linear cryptanalysis is that it requires half of the code book. Only follow-up works have been able to reduce this data requirement, including the more powerful distinguisher called multiple zero-correlation (MPZC) linear distinguisher proposed in [16], which exploits the fact that there are numerous zero-correlation linear approximations in susceptible ciphers.

Splice-and-cut meet-in-the-middle attacks and biclique attacks are other examples of attacks that often come with time complexities relatively close to exhaustive search. Indeed, an extension of the biclique approach first described in [14] has a brute-force phase for a number of rounds as part of the attack. It can in principle work for any number of rounds and is hence best described as a particular optimization of brute-force key guessing. However, later variants then showed examples where the gain over brute force was in the order of millions [37].

Finally, we point out that any attack that is better than brute force is relevant, even if it requires unrealistic amounts of data or storage. Indeed, the main goal of cryptanalysis is finding a "certificated weakness", that is, an evidence that the cipher does not perform as advertised. In other words, in academic cryptography, a weakness or a break in a scheme is usually defined quite conservatively: it may require impractical amounts of time, memory, or plaintexts.

The Number of Rounds Needed for Security. It may be of interest to estimate the number of rounds needed for MiMC to be resistant against this attack. To this end, we bound the operations needed to compute all monomials of odd degree, up to a maximum degree D (see Appendix B for a proof):

Lemma 3. *Let $1 \leq D \leq 2^n - 1$ and $x \in \mathbb{F}_{2^n}$. The overall number of operations needed to compute all odd powers x^i for $i \in [3, D]$ is given by 1 squaring and $\left\lfloor \frac{D-1}{2} \right\rfloor$ multiplications.*

Assume for simplicity that $\lceil n \cdot \log_3(2) \rceil - 1$ rounds can be covered by a zero sum, and that the cost of solving the final polynomial equation is negligible. As before, we expect the time complexity to be dominated by the number of operations needed to construct the polynomial $F(K)$. Since the degree of this polynomial is upper-bounded by $3^{r_{\text{KR}}}$, by Lemma 3 at most $(3^{r_{RK}} - 1)/2 \times 2^{n-2}$ multiplications are required to compute all monomials with odd exponents in $F(K)$ (where all monomials with even exponents are computed via Eq. (12)).

Since one encryption of MiMC costs $\lceil n \cdot \log_3(2) \rceil$ multiplications, the number of extra rounds ρ for MiMC must satisfy

$$(3^{\rho+1} - 1) \cdot 2^{n-2} \geq 2^n \cdot (\lceil n \cdot \log_3(2) \rceil + \rho)$$

in order to provide security against our attack just presented. This would, for example, require at least $\rho = 5$ extra rounds for $n = 129$. We remark that *this rough estimation is not intended to replace the number of rounds proposed by the designers.*

6 An Algebraic Attack on Ciphers with Low-Degree Round Functions

In this section, we generalize the key-recovery attack on MiMC described in Section 5 and discuss a generic attack strategy for any block cipher working over $(\mathbb{F}_{2^n})^t$, where $n, t \in \mathbb{N}$, $n, t \geq 1$.

6.1 Setting

We consider an r-round block cipher $E_k^r : (\mathbb{F}_{2^n})^t \to (\mathbb{F}_{2^n})^t$ with

$$E_k^r(x) = (R_r \circ R_{r-1} \circ \cdots \circ R_1)(x \oplus k),$$

and where $R, R_i : (\mathbb{F}_{2^n})^t \to (\mathbb{F}_{2^n})^t$ are defined by $R_i(x) = R(x) \oplus k^{(i)}$. Then we can write

$$E_k^r(x) = (E_{k,1}^r(x), \ldots, E_{k,t}^r(x)),$$

where $E_{k,i}^r : (\mathbb{F}_{2^n})^t \to \mathbb{F}_{2^n}$. The compositional inverse of E_k^r is denoted by E_k^{-r}. We assume that

(1) the i-th round key $k^{(i)} \in (\mathbb{F}_{2^n})^t$ is derived from the master key $k = (k_1, \ldots, k_t) \in (\mathbb{F}_{2^n})^t$ by some *low-degree* key schedule, and that
(2) the round function R can be described by a *low-degree* polynomial

$$R(x = (x_1, \ldots, x_t)) = \bigoplus_{\substack{j=(j_1,\ldots,j_t)\in\{0,1,\ldots,2^n-1\}^t \\ j_1+\ldots+j_t\le d}} \alpha_j \cdot x_1^{j_1} \cdot \ldots \cdot x_t^{j_t}$$

of degree d with coefficients $\alpha_j \in (\mathbb{F}_{2^n})^t$.

We highlight that several primitives satisfy above assumptions and do indeed use low-degree round functions (e.g., LowMC [4] and HadesMiMC [31]).

Our attack requires the symbolic evaluation of the encryption function $E_k^{r'}$ for a small number of rounds r' to be relatively easy, which motivates the requirements of a low-degree round function R and a low degree key-schedule. This ensures that the polynomial representation of $E_k^{r'}$ can be computed efficiently.

6.2 Strategy of the Attack

The idea of our generic attack is to recover the secret master key k of a cipher E_k^r by exploiting a given zero-sum distinguisher over the subset $\mathcal{X} \subseteq (\mathbb{F}_{2^n})^t$ covering $1 \le r_{\mathrm{ZS}} < r$ rounds in encryption or decryption direction[12]. For the sake of simplicity, we follow the approach of the attack on MiMC in Section 5 and assume the zero-sum distinguisher covers the decryption direction.

Roughly speaking, in our attack we symbolically evaluate $E_k^{r_{\mathrm{KR}}}$ with respect to the remaining $r_{\mathrm{KR}} := r - r_{\mathrm{ZS}}$ rounds in encryption direction and obtain polynomials $F_1(K_1, \ldots, K_t), \ldots, F_t(K_1, \ldots, K_t)$ over $\mathbb{F}_{2^n}$ with the master key words K_i as indeterminates. Eventually we solve the polynomial equation system $F_1(k_1, \ldots, k_t) = \cdots = F_t(k_1, \ldots, k_t) = 0$ for $k_1, \ldots, k_t \in \mathbb{F}_{2^n}$.

Assuming the attacker can set up a zero-sum distinguisher in the decryption direction (note that $E_k^r = E_k^{r_{\mathrm{ZS}}} \circ E_k^{r_{\mathrm{KR}}}$ and $E_k^{-r_{\mathrm{ZS}}} = E_k^{r_{\mathrm{KR}}} \circ E_k^{-r}$), we now describe the main idea behind our attack: having a zero sum after r_{ZS} rounds of decryption means there is a proper subset $\mathcal{X} \subseteq (F_{2^n})^t$ such that

$$\bigoplus_{x\in\mathcal{X}} E_k^{-r_{\mathrm{ZS}}}(x) = 0.$$

[12] As done before, we assume that the degree of the round function R is smaller than the degree of the inverse round function R^{-1}. If this is not the case, it is sufficient to work on the decryption direction instead of the encryption one.

Algorithm 3: Attack on a generic cipher E_k^r over $(\mathbb{F}_{2^n})^t$.

Input: Number of rounds r of the cipher E_k^r, number of rounds r_{ZS} in decryption direction for which the zero sum holds, a subset $\mathcal{X} \subseteq (\mathbb{F}_{2^n})^t$ satisfying the zero sum $\bigoplus_{x \in \mathcal{X}} E_k^{-r_{\mathrm{ZS}}}(x) = 0$.
Output: Secret key $k = (k_1, \ldots, k_t)$.

```
for each (i_1, ..., i_t) ∈ {0, 1, ..., D}^t with i_1 + ... + i_t ≤ D do
    P_{i_1,...,i_t} ← 0.
r_KR ← r − r_ZS.
Let D = D(r_KR) be the degree of E_k^{r_KR}.
for x ∈ X do
    y = (y_1, ..., y_t) ← E_k^{−r}(x) from the decryption oracle.
    for each (i_1, ..., i_t) ∈ {0, 1, ..., D}^t with i_1 + ... + i_t ≤ D do
        P_{i_1,...,i_t} ← P_{i_1,...,i_t} ⊕ y_1^{i_1} · ... · y_t^{i_t}.
for each 1 ≤ i ≤ t do
    Compute the symbolic evaluation
     f_i = f_i(Y_1, ..., Y_t, K_1, ..., K_t) = E^{r_KR}_{(K_1,...,K_t),i}(X_1, ..., X_t) of word i in
     encryption direction for r_KR rounds.
    for each Y_1^{i_1} ... Y_t^{i_t} · K_1^{j_1} ... K_t^{j_t} in f_i do
        Replace Y_1^{i_1} ... Y_t^{i_t} · K_1^{j_1} ... K_t^{j_t} with "P_{i_1,...,i_t} · K_1^{j_1} · ... · K_t^{j_t}".
    F_i(K_1, ..., K_t) ← f_i(K_1, ..., K_t).
Find a solution k = (k_1, ..., k_t) of F_1(k_1, ..., k_t) = ··· = F_t(k_1, ..., k_t) = 0.
return k = (k_1, ..., k_t).
```

We then exploit the relation

$$0 = \bigoplus_{x \in \mathcal{X}} E_k^{-r_{\mathrm{ZS}}}(x) = \bigoplus_{x \in \mathcal{X}} \left(E_k^{r_{\mathrm{KR}}} \circ E_k^{-r}\right)(x) = \bigoplus_{y \in E_k^{-r}(\mathcal{X})} E_k^{r_{\mathrm{KR}}}(y)$$

to set up the following equations ($1 \leq i \leq t$) over $\mathbb{F}_{2^n}$ in the variables $k_1, \ldots, k_t$:

$$F_i(k_1, \ldots, k_t) := \bigoplus_{y \in E_k^{-r}(\mathcal{X})} E^{r_{\mathrm{KR}}}_{(k_1,\ldots,k_t),i}(y) = 0. \tag{13}$$

Here $E^{r_{\mathrm{KR}}}_{(k_1,\ldots,k_t),i}(y)$ denotes the symbolic evaluation of word i after r_{KR} rounds in encryption direction with the master key words as variables $k_1, \ldots, k_t$ and evaluated at $y \in \mathbb{F}_{2^n}$. Once we have set up the equation system arising from Eq. (13), we apply Gröbner basis techniques to solve this system over $\mathbb{F}_{2^n}$ for the key variables $k_1, \ldots, k_t$. In Algorithm 3 we summarize our approach and present a pseudo code of the generic attack strategy.

6.3 Complexity Estimations

For our complexity estimations we count finite field operations over $\mathbb{F}_{2^n}$. We consider multiplications and squarings separately, since the squaring operation is an $\mathbb{F}_2$-linear operation in fields of characteristic 2.

As is the case for the attack on MiMC in Section 5, the generic attack strategy is composed of two steps. First, we construct the system of equations $F_i(k_1, \ldots, k_t) = 0$ for $1 \le i \le t$, and then we solve this system over $\mathbb{F}_{2^n}$ for $k_1, k_2, \ldots, k_t$. We recall that the cost of the first step grows with the size of $\mathcal{X}$, the subset needed for a zero sum. Since estimating the complexity for these steps more precisely would require a thorough analysis of the particular polynomial system in question, in the following we briefly describe these two steps without going into all the details an attacker could potentially exploit.

Setting Up the Equation System. For the equation system, we first need to symbolically evaluate r_{KR} encryption rounds, which results in t polynomials

$$E^{r_{\text{KR}}}_{(K_1,\ldots,K_t),i}(Y_1, \ldots, Y_t), \quad 1 \le i \le t,$$

of degree $D = D(r_{\text{KR}})$ over $\mathbb{F}_{2^n}$ in variables $K_1, \ldots, K_t$ and $Y_1, \ldots, Y_t$. Every monomial $Y_1^{i_1} \cdots Y_t^{i_t}$ in any polynomial $E^{r_{\text{KR}}}_{(K_1,\ldots,K_t),i}(Y_1, \ldots, Y_t)$ needs to be replaced by

$$\mathscr{P}_{i_1,\ldots,i_t} := \bigoplus_{y=(y_1,\ldots,y_t)\in E_k^{-r}(\mathcal{X})} y_1^{i_1} \cdot \ldots \cdot y_t^{i_t},$$

leaving us with t polynomials in the key variables $K_1, \ldots, K_t$ as indeterminates. Here we need an estimation for computing all $\mathscr{P}_{i_1,\ldots,i_t}$, or equivalently to write down a system of equations of the form as in Eq. (13).

For $t = 1$, the number of multiplications and squarings needed was stated in Lemma 3. The situation is more complicated for $t \ge 2$, since several strategies can be used to compute the monomials and minimize the number of multiplications, the number of squarings, or the memory cost. Since this depends on the details of the considered primitives, we limit ourselves to present a high-level analysis of two extreme cases in Appendix B, namely $n = 1$ (which corresponds e.g. to LowMC) and $n \ge 3$ and $D \le 2^n - 1$ (which corresponds e.g. to HadesMiMC).

Complexity Estimation for Solving the Equation System. For $t > 1$, the resulting equation system is a multivariate polynomial system. If we additionally have $n > 1$, the standard strategy for finding the solutions of such systems[13] is through a Gröbner basis [24]. Such an attack essentially consists of first computing a Gröbner basis in *degrevlex* order, then converting it to the *lex* order, and finally factorizing a univariate polynomial in this basis and back-substituting its roots. It is in general a hard problem to estimate the complexity needed for these steps. As largely done in the literature, we assume that the most expensive step is the first one (i.e., computing a Gröbner basis in *degrevlex* order). For generic systems, the complexity of this step for a system of $\mathfrak{N}$ polynomials f_i in $\mathfrak{V}$ variables is $\mathcal{O}\left(\binom{\mathfrak{V}+D_{\text{reg}}}{D_{\text{reg}}}^{\omega}\right)$ operations over the base field $\mathbb{F}$, where D_{reg} is the *degree of regularity* [8] and $2 \le \omega < 3$ is the linear algebra constant. The degree

[13] Strategies that involve guessing (parts of) the variables may be more viable over $\mathbb{F}_2$.

of regularity depends on the number of polynomials $\mathfrak{N}$, their degrees d_i, as well as the algebraic structure of the system. Closed-form formulas for D_{reg} are only known for some special cases: e.g. if $\mathfrak{V} = \mathfrak{N}$ (namely, the case considered in this attack), a simple closed form is given by $D_{\text{reg}} = 1 + \sum_{i=0}^{\mathfrak{N}-1}(d_i - 1)$.

As discussed later in details, we remark that this may be a pessimistic upper bound: the algebraically simple ciphers we are considering can end up exhibiting more algebraic structure than what is the case for generic systems.

6.4 Comparison with Related Work

Cube Attacks. Our attack relies on similar properties as cube attacks [49,26], which exploit low-degree relations between components of a cryptosystem. Given a cipher with secret variables $\boldsymbol{x} \in \mathbb{F}_2^n$ and public variables $\boldsymbol{v} \in \mathbb{F}_2^m$, the idea is to regard it as a polynomial of $\boldsymbol{x}$ and $\boldsymbol{v}$, and denote it as $f(\boldsymbol{x}, \boldsymbol{v})$. For a randomly chosen set $I = \{i_1, i_2, ..., i_{|I|}\} \subset \{1, \ldots, m\}$, $f(\boldsymbol{x}, \boldsymbol{v})$ can be represented uniquely as

$$f(\boldsymbol{x}, \boldsymbol{v}) = t_I \cdot p(\boldsymbol{x}, \boldsymbol{v}) + q(\boldsymbol{x}, \boldsymbol{v}),$$

where $t_I = v_{i_1} \cdots v_{i_{|I|}}$, the polynomial $p(\boldsymbol{x}, \boldsymbol{v})$ only relates to v_s's ($s \notin I$) and the secret key bits $\boldsymbol{x}$, and $q(\boldsymbol{x}, \boldsymbol{v})$ misses at least one variable in t_I. A specific structure where all variables in the set $\{v_{i_1}, v_{i_2}, \cdots, v_{i_{|I|}}\}$ (with indices determined by the set I) take all possible values and where the remaining variables are static is called a *cube* of f and is denoted by C_I. The sum of f over the cube C_I

$$\bigoplus\nolimits_{C_I} f(\boldsymbol{x}, \boldsymbol{v}) = \bigoplus\nolimits_{C_I} (t_I \cdot p(\boldsymbol{x}, \boldsymbol{v}) + q(\boldsymbol{x}, \boldsymbol{v})) = p(\boldsymbol{x}, \boldsymbol{v})$$

is called the *superpoly* of the cube C_I, and it is found by the attacker during an offline phase. Then, in the online phase, they query the encryption oracle with the cube, and finally get the value of the superpoly. The secret key can be recovered easily when the polynomial $p(\boldsymbol{x}, \boldsymbol{v})$ is simple.

Compared to our attack, cube attacks involve the additional (offline) step of identifying suitable superpolys. Note that this attack does not exploit the algebraic structure of the cipher. Instead, the goal of the sum over the cube is to "eliminate" all the details of the attacked scheme (apart from linear relations in the key). In this sense, our attack is different, since it makes heavy use of the algebraic structure of the cipher which is evaluated symbolically. The disadvantage regards the strong assumption that some rounds of the cipher can be described by simple algebraic equations. On the other hand, this allows to work at word level in the case of ciphers natively defined in $(\mathbb{F}_{2^n})^t$, and no (potentially expensive) offline phase is needed. Our method is thus particularly relevant for primitives designed for new applications such as STARKs and MPC, which work over larger fields.

Optimized Interpolation Attacks [25]. One type of optimized interpolation attacks was described in [25], where the authors use it to find attacks on reduced-round versions of LowMC. A similar attack has been proposed in [27], and later on it was also used to break the full-round version of the FRIT permutation in an Even–Mansour setting [28].

The overall strategy of this interpolation attack is to first find a distinguisher (for example a constant sum in the encryption direction in the case of LowMC) and to then attack the construction by finding the unknown monomials of the sums of the symbolic representations in the inverse direction. By determining these (key-dependent) monomials, the full key can eventually be found. Since this approach has some similarities with our proposal, here we describe the differences between these two strategies in detail.

The main difference regarding the two strategies concerns the way in which the system of equations $F_i(K) = 0$ is constructed and consequently solved:

- In [25], the idea is to construct the function using a "standard" interpolation technique. Specifically, the attacker does not care about the specification of the monomials of F, which are simply considered as unknowns. Hence, the idea is to recover (interpolate) the unknown coefficients of $F_K(C)$, and then use various ad-hoc techniques (which are not part of the framework described in this section) in order to recover the actual secret key.
- In our case, we heavily exploit the simple algebraic structure of the round function in order to construct the system of equations $F(K) = 0$. In other words, the system of equations is constructed by using a symbolic evaluation and not by interpolation techniques.

Each one of the two strategies has advantages and disadvantages. Therefore, the choice of which variant to use in order to optimize the attack depends on the details of the underlying cryptosystem:

Data Cost. In the first case, more data is necessary in order to set up the interpolation step. Indeed, besides the data necessary for the distinguisher, the attacker requires more data in order to recover the coefficients of F. In the second case, however, the data for the distinguisher is sufficient.

Assumption on the Round Function. For the symbolic evaluation, our attack assumes the round function to exhibit a simple and low-degree algebraic structure. This is not necessary in the other case, in which the attacker can (mostly) ignore the algebraic structure, which is then found by interpolation.

We emphasize that the possibility to set up one of the two attacks does not imply the possibility to set up the other one. For example, it seems hard to use the attack presented in [25] against full-round MiMC, while we show that our strategy can indeed break it. Indeed, since we already need 2^{n-1} data for the distinguishing property (i.e., half of the code book), we do not see how to apply the approach from [25] to MiMC without further increasing the data complexity due to data needed for the interpolation step.

Higher-Order Diff. Attack on CAST [42]. In an attack on the CAST cipher from 1998 [42], the authors use a higher-order differential distinguisher to set up an equation system and finally solve this systems for the key variables. The difference to our approach is that the authors work with (linearized) equation systems over $\mathbb{F}_2$ and thus only with linear equations. While this is sufficient for CAST, working at bit level is in general much more expensive than working on word level when focusing on ciphers that are natively defined at word level.

6.5 Concluding Remarks

Better Cost Estimations. The previous estimations of the complexity of the attacks can be improved by exploiting the details of the cipher. To give a concrete example, consider the case of MiMC given in Algorithm 2: The attack and its computational complexity benefit from the fact that $F(K)$ does not depend on $\mathscr{P}_5$ or $\mathscr{P}_7$. As another example, consider the case of an SPN cipher where the round function is defined as

$$R(x = (x_1, \dots, x_t)) = M \cdot (S(x_1), S(x_2), ..., S(x_t)),$$

where $M \in (\mathbb{F}_{2^n})^{t \times t}$ and $S : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. The cost of the attack can potentially be reduced by taking into account the fact that all monomials in the polynomial representation R depend only on a single variable x_i.

Using Dedicated Tools. While for MiMC it is possible to handle the symbolic evaluation of $F(K)$, this may not be the case for more complicated ciphers like LowMC or HadesMiMC. In these cases, we suggest to set up a tool or script which allows to deal with the symbolic evaluation of $F(K)$ in an easier way.

Minimizing the Cost of the Attack. As already pointed out, two steps mainly contribute to the cost of the attack. In general, it could make sense to balance the costs of the two steps in order to either minimize the total cost of the attack or maximize the number of rounds that can be broken.

In more detail, consider the case in which the cost of the attack is well approximated by the cost of *constructing* the system of equations $F_i(K) = 0$. Since this cost grows with the size of the subspace $\mathcal{V}$, one strategy could be to consider a smaller subset[14] $\mathcal{X}$. Obviously, this implies in general the possibility to cover less rounds r_{ZS} using a zero-sum distinguisher, which means that more rounds r_{KR} must be covered in general. However, the overall cost of the attack may benefit from this strategy.

On the other hand, the case that the attack cost is well approximated by the cost of *solving* the system of equations $F_i(K) = 0$ requires the opposite strategy.

Further Generalization: Ciphers over $\mathbb{F}_p$. Finally, the attack strategy can be generalized to include ciphers over $(\mathbb{F}_p)^t$ for a prime p. This is of particular importance since many of the new applications named in the introduction (e.g., STARKs and MPC) natively work over $\mathbb{F}_p$, which means that many of the recently

[14] We note that we cannot adopt this strategy for MiMC since we are not able to predict the growth of the degree of MiMC^{-1}. With such an estimation, the strategy proposed here can potentially reduce the cost of the attack.

proposed primitives are natively constructed over $\mathbb{F}_p$. We remark that the strategy of the attack does not depend on the details of the field $\mathbb{F}$. Hence, the only thing that seems to preclude this possibility seems to be a lack of knowledge regarding efficient distinguishers over $(\mathbb{F}_p)^t$. Indeed, while it is well-known how to find a zero-sum distinguisher over boolean fields (e.g., by exploiting division property tools present in the literature [47,51,53]), the same is not yet true for prime fields.

References

1. Albrecht, M.R., Cid, C., Grassi, L., Khovratovich, D., Lüftenegger, R., Rechberger, C., Schofnegger, M.: Algebraic Cryptanalysis of STARK-Friendly Designs: Application to MARVELlous and MiMC. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 371–397 (2019)
2. Albrecht, M.R., Grassi, L., Perrin, L., Ramacher, S., Rechberger, C., Rotaru, D., Roy, A., Schofnegger, M.: Feistel Structures for MPC, and More. In: ESORICS 2019. LNCS, vol. 11736, pp. 151–171 (2019)
3. Albrecht, M.R., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: ASIACRYPT 2016. LNCS, vol. 10031, pp. 191–219 (2016)
4. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454 (2015)
5. Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., Szepieniec, A.: Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. IACR Cryptology ePrint Archive, Report 2019/426 (2019)
6. Ashur, T., Dhooghe, S.: MARVELlous: a STARK-Friendly Family of Cryptographic Primitives. IACR Cryptology ePrint Archive, Report 2018/1098 (2018)
7. Aumasson, J.P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi (2009), presented at the Rump Session of CHES 2009, `https://131002.net/data/papers/AM09.pdf`
8. Bardet, M., Faugere, J., Salvy, B., Yang, B.: Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In: The Effective Methods in Algebraic Geometry Conference (MEGA). pp. 1–14 (2005)
9. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. IACR Cryptology ePrint Archive **2018**, 46 (2018)
10. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: On the indifferentiability of the sponge construction. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197 (2008)
11. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Note on zero-sum distinguishers of Keccak-f, `http://keccak.noekeon.org/NoteZeroSum.pdf`
12. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: CRYPTO. Lecture Notes in Computer Science, vol. 537, pp. 2–21. Springer (1990)
13. Black, J., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: CRYPTO 2002. LNCS, vol. 2442, pp. 320–335 (2002)
14. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full AES. In: ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371 (2011)
15. Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Cryptogr. **70**(3), 369–383 (2014), see also: Cryptology ePrint Archive, Report 2011/123

16. Bogdanov, A., Wang, M.: Zero Correlation Linear Cryptanalysis with Reduced Data Complexity. In: FSE 2012. LNCS, vol. 7549, pp. 29–48 (2012)
17. Bonnetain, X.: Collisions on Feistel-MiMC and univariate GMiMC. IACR Cryptology ePrint Archive **2019**, 951 (2019)
18. Boura, C., Canteaut, A.: On the Influence of the Algebraic Degree of F^{-1} on the Algebraic Degree of $G \circ F$. IEEE Trans. Information Theory **59**(1), 691–702 (2013)
19. Boura, C., Canteaut, A., De Cannière, C.: Higher-Order Differential Properties of Keccak and *Luffa*. In: FSE 2011. LNCS, vol. 6733, pp. 252–269 (2011)
20. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: IEEE Symposium on Security and Privacy. pp. 315–334. IEEE Computer Society (2018)
21. Canteaut, A., Videau, M.: Degree of Composition of Highly Nonlinear Functions and Applications to Higher Order Differential Cryptanalysis. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 518–533 (2002)
22. Cantor, D.G., Kaltofen, E.: On Fast Multiplication of Polynomials over Arbitrary Algebras. Acta Inf. **28**(7), 693–701 (1991)
23. Carlet, C., Charpin, P., Zinoviev, V.A.: Codes, bent functions and permutations suitable for DES-like cryptosystems. DCC **15**(2), 125–156 (1998)
24. Cox, D.A., Little, J., O'Shea, D.: Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.). Undergraduate texts in mathematics, Springer (1997)
25. Dinur, I., Liu, Y., Meier, W., Wang, Q.: Optimized Interpolation Attacks on LowMC. In: ASIACRYPT 2015. LNCS, vol. 9453, pp. 535–560 (2015)
26. Dinur, I., Shamir, A.: Cube Attacks on Tweakable Black Box Polynomials. In: EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299 (2009)
27. Dobraunig, C., Eichlseder, M., Mendel, F.: Higher-Order Cryptanalysis of LowMC. In: ICISC 2015. LNCS, vol. 9558, pp. 87–101 (2015)
28. Dobraunig, C., Eichlseder, M., Mendel, F., Schofnegger, M.: Algebraic Cryptanalysis of Variants of Frit. In: SAC 2019. LNCS, vol. 11959, pp. 149–170 (2019)
29. von zur Gathen, J., Gerhard, J.: Modern Computer Algebra (3. ed.). Cambridge University Press (2013)
30. Grassi, L., Kales, D., Khovratovich, D., Roy, A., Rechberger, C., Schofnegger, M.: Starkad and Poseidon: New Hash Functions for Zero Knowledge Proof Systems. Cryptology ePrint Archive, Report 2019/458 (2019)
31. Grassi, L., Lüftenegger, R., Rechberger, C., Rotaru, D., Schofnegger, M.: On a Generalization of Substitution-Permutation Networks: The HADES Design Strategy. In: EUROCRYPT 2020 (2020), to Appear
32. Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., Smart, N.P.: Mpc-friendly symmetric key primitives. In: ACM Conference on Computer and Communications Security. pp. 430–443. ACM (2016)
33. Guo, J., Liao, G., Liu, G., Liu, M., Qiao, K., Song, L.: Practical Collision Attacks against Round-Reduced SHA-3. Journal of Cryptology **33**(1), 228–270 (2020)
34. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for Three-Subset Division Property without Unknown Subset–Improved Cube Attacks against Trivium and Grain-128AEAD. EUROCRYPT 2020, to Appear
35. Hu, K., Wang, M.: Automatic Search for a Variant of Division Property Using Three Subsets. In: CT-RSA 2019. LNCS, vol. 11405, pp. 412–432 (2019)
36. Jakobsen, T., Knudsen, L.R.: The interpolation attack on block ciphers. In: FSE. LNCS, vol. 1267, pp. 28–40 (1997)

37. Khovratovich, D., Leurent, G., Rechberger, C.: Narrow-bicliques: Cryptanalysis of full IDEA. In: EUROCRYPT 2012. LNCS, vol. 7237, pp. 392–410 (2012)
38. Knudsen, L.R.: Truncated and Higher Order Differentials. In: FSE 1994. LNCS, vol. 1008, pp. 196–211 (1994)
39. Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324 (2007)
40. Li, C., Preneel, B.: Improved Interpolation Attacks on Cryptographic Primitives of Low Algebraic Degree. In: SAC 2019. LNCS, vol. 11959, pp. 171–193 (2019)
41. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397 (1993)
42. Moriai, S., Shimoyama, T., Kaneko, T.: Higher Order Differential Attak of CAST Cipher. In: FSE 1998. LNCS, vol. 1372, pp. 17–31 (1998)
43. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly practical verifiable computation. In: IEEE Symposium on Security and Privacy. pp. 238–252. IEEE Computer Society (2013)
44. Rotaru, D., Smart, N.P., Stam, M.: Modes of Operation Suitable for Computing on Encrypted Data. IACR Trans. Symmetric Cryptol. **2017**(3), 294–324 (2017)
45. Sasaki, Y., Todo, Y.: New Impossible Differential Search Tool from Design and Cryptanalysis Aspects - Revealing Structural Properties of Several Ciphers. In: EUROCRYPT 2017. LNCS, vol. 10212, pp. 185–215 (2017)
46. Todo, Y.: Structural Evaluation by Generalized Integral Property. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 287–314 (2015)
47. Todo, Y., Isobe, T., Hao, Y., Meier, W.: Cube Attacks on Non-Blackbox Polynomials Based on Division Property. In: CRYPTO 2017. LNCS, vol. 10403, pp. 250–279 (2017)
48. Todo, Y., Morii, M.: Bit-Based Division Property and Application to Simon Family. In: FSE 2016. LNCS, vol. 9783, pp. 357–377 (2016)
49. Vielhaber, M.: Breaking ONE.FIVIUM by AIDA an algebraic IV differential attack. IACR Cryptology ePrint Archive **2007**, 413 (2007)
50. Wang, Q., Grassi, L., Rechberger, C.: Zero-Sum Partitions of PHOTON Permutations. In: CT-RSA 2018. LNCS, vol. 10808, pp. 279–299 (2018)
51. Wang, Q., Hao, Y., Todo, Y., Li, C., Isobe, T., Meier, W.: Improved Division Property Based Cube Attacks Exploiting Algebraic Properties of Superpoly. In: CRYPTO 2018. LNCS, vol. 10991, pp. 275–305 (2018)
52. Wang, Q., Liu, Z., Varici, K., Sasaki, Y., Rijmen, V., Todo, Y.: Cryptanalysis of Reduced-Round SIMON32 and SIMON48. In: INDOCRYPT 2014. LNCS, vol. 8885, pp. 143–160 (2014)
53. Wang, S., Hu, B., Guan, J., Zhang, K., Shi, T.: MILP-aided Method of Searching Division Property Using Three Subsets and Applications. In: ASIACRYPT 2019. LNCS, vol. 11923, pp. 398–427 (2019)

SUPPLEMENTARY MATERIAL

Scripts and Implementations

The MAGMA script *Magma_Script_MiMC_Univariate_Attack* has two input parameters: N and *version*. N is an odd integer that decides the block size of MiMC, i.e., MiMC-N/N. The second parameter $version \in \{1, 2\}$ determines whether to use Algorithm 1 or Algorithm 2. The script creates an instance of MiMC-N/N, and runs a key-recovery attack using the chosen algorithm. It outputs the roots of $F(K)$, as well as the secret key k for comparison.

We also provide the file *zero_sum_tester.cpp*, which contains the code we used to find the zero sums for MiMC. It accepts three parameters: the field size, the number of rounds, and the dimension of the vector space.

A Algebraic Degree Growth of MiMC^{-1}

While not needed for our attack, we also analyzed the degree growth of MiMC in the decryption direction. The results of the tests we applied and the size of the vector space dimensions necessary for zero-sum distinguishers are shown in Table 3.

As we can see, the algebraic degree does not increase in the second round for the instances we tested, and after that it starts growing slowly. Moreover, it seems to remain consistent after roughly half the number of rounds, until it finally reaches its maximum in the final round.

n \ r'	2	3	4	5	6	7	8	9	10	11	12
7	5	6	6	–	–	–	–	–	–	–	–
9	6	7	8	8	–	–	–	–	–	–	–
11	7	9	9	10	10	–	–	–	–	–	–
13	8	10	11	11	12	12	12	–	–	–	–
15	9	11	12	13	14	14	14	14	–	–	–
17	10	13	14	15	16	16	16	16	16	–	–
19	11	14	15	16	17	18	18	18	18	18	–

Table 3: Data complexities necessary for zero sums when evaluating MiMC in decryption direction for various block sizes and round numbers.

B Proofs

Proof of Lemma 2. By definition, note that $\delta_i \leq 2i$ and that $\delta_i \leq n-1$ for each i. Since $2i \leq n-1$ if $i \leq (n-1)/2$, it follows that

$$\gamma = \max_{1 \leq i \leq n-1} \binom{n-i}{n-\delta_i} \leq \max\left\{ \max_{1 \leq i \leq \frac{n-1}{2}} \binom{n-i}{n-2i} ; n - \frac{n-1}{2} - 1 \right\} = \frac{n+1}{2},$$

where $\max_{\frac{n-1}{2}+1 \le i \le n-1} \binom{n-i}{n-\delta_i} = \max_{\frac{n-1}{2}+1 \le i \le n-1} \binom{n-i}{n-(n-1)} = n - \frac{n-1}{2} - 1$. The bound given in Eq. (9) is obtained by replacing γ with $(n+1)/2$ in Eq. (3). $\square$

Proof of Lemma 3. From x, calculate and store $q := x^2$. The odd powers of x can now be successively computed as $x^{i+2} = x^i \cdot q$ for all odd integers i in the interval $[1, D-2]$. This yield a total of 1 squaring and $\lfloor \frac{D-1}{2} \rfloor$ multiplications. $\square$

The Number of Multiplications Needed in Section 6.3

Here we limit ourselves to analyze two extreme cases, namely

(1) $n = 1$, and
(2) $n \geq 3$ and $D \leq 2^n - 1$.

In the first case, the number of multiplications can be upper-bounded by the number of different monomials, namely

$$\sum_{i=1}^{D} \binom{t}{i}$$

where $D < t$. This is done through successive multiplications by degree, i.e. every monomial of degree d can be computed by combining a monomial of degree $d-1$ with a single multiplication.

In the second case, we propose the following Lemma.

Lemma 4. *Let D be an integer with $1 \leq D \leq 2^n - 1$. The number of non-squaring based multiplications needed to compute all monomials of total degree at most D in t variables over $\mathbb{F}_{2^n}$ is upper-bounded by*

$$\left(\sum_{i=2}^{D} \binom{i+t-1}{t-1} \right) - t \cdot \frac{D-1}{2}$$

Proof. It is a well-known fact that the number of different monomials of degree d in t variables is

$$M_d = \binom{d+t-1}{t-1}.$$

As above we use a successive multiplication, where every monomial of degree d can be computed by combining a monomial of degree $d-1$ with a single multiplication. It follows that the number of multiplications needed to compute all monomials in at least two variables up to degree D is upper-bounded by

$$M = \sum_{i=2}^{D} (M_i - t) = \left(\sum_{i=2}^{D} \binom{i+t-1}{t-1} \right) - (D-1)t.$$

Lastly, we add the t univariate monomials to M, which by Lemma 3 amounts to at most $t \cdot \frac{D-1}{2}$ multiplications. $\square$

We note that a monomial with all univariate degrees being even can be generated by squaring a lower-degree monomial. This fact is not considered in Lemma 4 since such a squaring is counted as a non-squaring based multiplication. Hence, there might be a different trade-off between squarings and non-squaring based multiplications when counting the number of multiplications for computing all monomials of total degree at most D in t variables. Potentially, the trade-off might be improved in favour of squarings when dealing with a concrete cipher.

C Division Property and Automatic Tools

In this section, we evaluate some of our practical cases by mixed integer linear programming (MILP) based on the division property. By using this tool, we show that we can get exactly the same zero-sum distinguisher as the practical implementation. However, the biggest limitation for the new tool is that it can only handle S-boxes of small size (e.g., 9 bits), while our new bound $\mathcal{R}^{\text{Linear}}$ has no limitation on the S-box size.

C.1 Brief Recall: (Word-Based) Division Property

The division property [46] – proposed by Todo at Eurocrypt 2015 – can be seen as a generalization of integral and higher-order differential distinguishers, and was used to present new generic distinguishers against both SPN and Feistel constructions.

We first introduce some notations for bit vectors. For any n-bit vector $\boldsymbol{x}$ and $0 \le i \le n-1$, we denote x_i as its i-th bit. Given two n-bit vectors $\boldsymbol{u}$ and $\boldsymbol{x}$, we define $\pi_{\boldsymbol{u}}(\boldsymbol{x}) = \Pi_{i=0}^{n-1} x_i^{u_i}$. Moreover, $\boldsymbol{u} \succeq \boldsymbol{k}$ denotes $u_i \geqslant k_i$ for all i.

Definition 2 (Word-Based Division Property[46]). *Let $\mathbb{X}$ be a multiset of n-bit vectors, and let k, $0 \le k \le n$, be an integer. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_k^n$, it fulfills $\forall u \in \mathbb{F}_2^n$ s.t. $\text{hw}(u) < k$:*

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = 0.$$

The novelty of the division property is that it introduces intermediate properties $\mathcal{D}_k^n$ for $3 \le k \le n-1$, which do not appear in classical integral attacks. These intermediate properties allow to easily propagate the property through the successive rounds of a cipher by capturing some information resulting from the algebraic degree of the round function. As it is already known, the original division property improved many previous integral distinguishers for ciphers. However, since it treated the round function at word level, by its nature some propagation information through it cannot be captured.

C.2 Two-Subset Bit-Based Division Property

Todo and Morii [48] introduced the bit-based division property where the propagation of the integral property of the concrete structures of the target primitives can be treated at bit level. As a consequence, more rounds of integral characteristics have been found with this new technique [60,57,50,58].

In the bit-based division property, two cases are considered where $\boldsymbol{u}$ can be classified into two sets, which is therefore called conventional bit-based division property or two-subset bit-based division property (`2-Set-BDP`), according to which the parity of $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ is even or unknown. The definition of the two-subset bit-based division property is as follows.

Definition 3 (Two-Subset Bit-Based Division Property [48]). *Let $\mathbb{X}$ be a multiset of n-bit vectors, and $\mathbb{K}$ be a set of n-bit vectors. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{K}}^{1^n}$, it fulfills the following conditions:*

$$\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \begin{cases} \textit{unknown}, & \textit{if } \exists\ \boldsymbol{k} \in \mathbb{K} \textit{ s.t. } \boldsymbol{u} \succeq \boldsymbol{k}, \\ 0, & \textit{otherwise.} \end{cases}$$

Our Practical Results with the Two-Subset Bit-Based Division Property. We model the propagation of `2-Set-BDP` using a MILP-aided tool [60].

"Small" S-Box. We model the S-boxes of size 5 and 7 by a set of linear equations as in [60]. Given an S-box, we first compute a set of vectors A (often called division trail table) which is composed of all division property propagation pairs, and then calculate the H-Representation of the convex hull of A by using the `inequality_generator()` function in SageMath,[15] and this will return a set of linear inequalities $\mathcal{L}$ that are the H-Representation of Conv(A). Since $\mathcal{L}$ is an accurate description of A, adding all the linear inequalities in $\mathcal{L}$ to the MILP model of searching for the division trails of a block cipher, will always return a valid division trail. A greedy algorithm is usually applied to reduce the number of inequalities in order to make the MILP problem computationally feasible.

By the above method, we can add 21 and 1216 inequalities respectively for the 5-bit and 7-bit cube S-boxes in Table 2 to the MILP model. After calling the MILP solvers, we find a 2-round zero-sum property for them, i.e., this MILP-aided evaluation only provides us a lower bound of 3 rounds that are necessary to prevent zero-sum distinguisher attacks. As one can see, for $n = 5$, the result obtained by this automatic tool is 1 round less than the practical result obtained by the implementation experiments (4 rounds); and for $n = 7$, it is 2 rounds less than the practical result. It seems that the accuracy of the MILP automatic tool based on the `2-Set-BDP` is much reduced, which refutes the commonly believed fact that one can always find the best integral distinguisher using `2-Set-BDP` for block ciphers, even when not taking the secret keys into consideration.

"Big" S-box. As far as we know, there is no efficient method that can describe the division trail table of an S-box larger than 8 bits. In fact, generating the linear inequalities for the H-representation of the convex hull, often by using SageMath, requires an exponential complexity in the number of input and output bits. For

[15] `https://www.sagemath.org`

example, for our 9-bit cube S-box, we obtain 15612 equations with the help of SageMath and a greedy algorithm. Such a large number of linear equations make the whole MILP system quite heavy for the off-shell optimization solvers, which might eventually result in out-of-memory errors. Sasaki and Todo [45] demonstrated an exhaustive list of compact representations in logical condition modeling against 4-bit S-boxes, but it is not applicable to larger S-boxes. In [54], Quine-McCluskey and the Espresso algorithm is proposed as a tool to generate constraint inequalities for 8-bit S-boxes, unfortunately this is not helpful especially for our applications with S-boxes much larger than 8 bits (e.g., 129 bits).

Therefore, we choose to model larger S-boxes by their ANF. Assume the n-bit S-box $\boldsymbol{y} = f(\boldsymbol{x})$, where we describe each coordinate $y_i = f_i(x_0, x_1, \ldots, x_{n-1})$ by operation rules for `COPY`, `AND`, and `XOR`. More details can be found in [60]. The bounds obtained by this automatic tool are far worse than the practical results, actually they are even worse than the bounds given by `2-Set-BDP`. Besides the hereditary inaccuracy of `2-Set-BDP`, another reason for this gap is that the way we model the S-box easily inserts a large amount of invalid division trails to the solution pool, which results in a quicker loss of the balanced property than the cipher itself would.

C.3 Three-Subset Bit-Based Division Property

Three-subset bit-based division property (`3-Set-BDP`) [48], where $\boldsymbol{u}$ is classified into three sets, i.e., "0", "1" and "unknown", seems to be more accurate than `2-Set-BDP`[35,53]. A formal definition is given as follows:

Definition 4. *(Three-Subset Bit-Based Division Property [48]). Let $\mathbb{X}$ be a multiset of n-bit vectors. Let $\mathbb{K}$ and $\mathbb{L}$ be two sets of n-bit vectors. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}^{1^n}_{\mathbb{K},\mathbb{L}}$, it fulfills the following conditions:*

$$\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \begin{cases} \textit{unknown}, & \textit{if } \exists\, \boldsymbol{k} \in \mathbb{K} \textit{ s.t. } \boldsymbol{u} \succeq \boldsymbol{k}, \\ 1, & \textit{else if } \exists\, \boldsymbol{\ell} \in \mathbb{L} \textit{ s.t. } \boldsymbol{u} = \boldsymbol{\ell}, \\ 0, & \textit{otherwise}. \end{cases}$$

So, compared to the `2-Set-BDP` where only $\mathbb{K}$ is used to trace the propagation, more accurate features will be revealed if we can model this propagation in an efficient way.

Influence of Secret Keys. For a public function, there is no effect that the propagation of $\mathbb{K}$ and $\mathbb{L}$ are evaluated independently. Moreover, in order to speed up the searching process, removing the redundant vectors in $\mathbb{K}$ and $\mathbb{L}$ will of course not result in any problem. However, when a secret round key is added to the intermediates, which is a common case in many block ciphers, the vectors in $\mathbb{L}$ will affect the vectors in $\mathbb{K}$.

The problem involving the secret key described above is handled by following the propagation rules in [48]: Assuming a round key is xored with the i-th bit, then for all $\boldsymbol{\ell} \in \mathbb{L}$ satisfying $\ell_i = 0$, a new vector $(\ell_1, \ell_2, \ldots, \ell_i \vee 1, \ldots, \ell_n)$ is appended to $\mathbb{K}'$. This propagation rule evokes the problem which is called *unknown-producing problem* in [34].

Influence of Focusing on Single Trail. Another important propagation rule is XOR rules for calculating vectors in $\mathbb{L}'$ from $\mathbb{L}$. If $\boldsymbol{\ell}$ is not included in $\mathbb{L}$ before, then it is inserted to $\mathbb{L}'$. If $\boldsymbol{\ell}$ has already been included in $\mathbb{L}$, then it is removed from $\mathbb{L}'$. This XOR rule results in the problem which is called *cancellation problem* in [34].

C.4 New Model for the Three-Subset Bit-Based Division Property without the Unknown Set

According to the propagation rules for the 3-Set-BDP, the unknown-producing problem implies that we need to know all the vectors in $\mathbb{L}_i$ when the secret key is xored, and the cancellation problem implies that focusing only on one single trail is not enough. Furthermore, after iterating i rounds, the amount of bit vectors in set $\mathbb{K}_i$ and $\mathbb{L}_i$ explodes, and this makes it even harder to trace the propagation of 3-Set-BDP directly.

Motivated to model 3-Set-BDP efficiently, Hu and Wang [35] proposed the *variant three-subset division property* to handle the unknown-producing problem. However, the cancellation problem is not considered in their model. As a result, the accuracy of this model is worse than the original 3-Set-BDP, though it is better than the 2-Set-BDP. In [53], the breadth-first search algorithm and the pruning technique were combined to model the 3-Set-BDP. As a result, it guarantees that the sizes of $\mathbb{K}_i$ and $\mathbb{L}_i$ decrease dramatically, and it seems that the evaluations based on the 3-Set-BDP becomes possible. However, the pruning technique is useful only when the size of $\mathbb{L}_i$ is reasonably small, which limits its applications.

In order to overcome these problems and trace the 3-Set-BDP efficiently, very recently, [34] proposed a new model formulating the 3-Set-BDP without the unknown set[16].

Definition 5. *(Modified Three-Subset Bit-Based Division Property [34]). Let $\mathbb{X}$ and $\mathbb{L}$ be multisets of n-bit vectors. When $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{L}}^{1^n}$, it fulfills the following conditions:*

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \begin{cases} 1, & \textit{if there are odd-number of } \boldsymbol{u} \textit{ in } \mathbb{L}, \\ 0, & \textit{otherwise.} \end{cases}$$

In this new model $\mathbb{L}$ is a multiset, which means it allows multiple bit vectors to exist. When undertaking the propagation of bit vectors, we count the number of bit vectors in $\mathbb{L}$. Accordingly, the propagation rules are slightly modified to guarantee the propagation of vectors in the multiset. More details can be found in [34].

[16] The idea of handling the cancellation problem is mentioned in [53], but it is not utilized in their MILP models.

Our Practical Results for Three-Subset Bit-Based Division Property. We build MILP models for the modified `3-Set-BDP` for our practical experiments for cases of MiMC with an n-bit S-box, where $n \in \{5, 7, 9\}$. We obtain exactly the same results as for the practical ones in Table 2. Therefore, we conclude that by modeling the modified `3-Set-BDP` with help of the MILP automatic tool, we can evaluate an accurate bound resistant to zero-sum distinguishing attacks for MiMC with "small" S-boxes.

However, as far as we know, there are no efficient methods to model a larger S-box with the (modified) `3-Set-BDP`. Thus, our $\mathcal{R}^{\text{Linear}}$ bound derived in this paper can evaluate S-boxes of any size, and give a bound very close to the practical result by implementation experiments (as can be seen in Table 2 for the case of the cube round function).

D Multivariate Attack Approach for MiMC

In this section, we consider attacking MiMC by solving a system of equations over $\mathbb{F}_2$. We will thus have n key variables. While this approach leads to a less efficient attack on MiMC when compared to our main approach described in Section 5, it may be useful for other cryptographic constructions which work only over $\mathbb{F}_2$.

D.1 Generating Low-Degree Equations in the Key Bits

Our goal is to find the key bits by solving a system of n key variables in n polynomials over $\mathbb{F}_2$. Only for simplicity, we focus on the instances where we can choose $r_{\text{KR}} = 1$ rounds of encryption. In order to build this system, we evaluate MiMC in encryption direction over one single round symbolically, where we keep the key bits as variables and where we use the concrete values obtained by the oracle for the input bits.

This step results in n sums of 2^{n-1} values, where each sum is a degree-1 polynomial over $\mathbb{F}_2$ in the variables $k_1, k_2, \ldots, k_n$. This is the case because all monomials p_i, $p_i \cdot p_j$ for $i \neq j$, and $k_i \cdot k_j$ for $i \neq j$, where $i \in [1, n], j \in [1, n]$, are removed after substitution and summation. The remaining monomials $p_i \cdot k_j$, where $i \in [1, n], j \in [1, n]$, are linear in the key bits after substitution.

D.2 Solving a System of n Linear Equations in n Variables

Since we know that the sum in each bit after one single round is 0 due to the number of chosen ciphertexts, our equation system has the following structure:

$$\begin{cases} f_1(k_1, k_2, \ldots, k_n) = 0 \\ f_2(k_1, k_2, \ldots, k_n) = 0 \\ \vdots \\ f_n(k_1, k_2, \ldots, k_n) = 0 \end{cases}$$

n	Time	Data
33	$2^{28.61}$	2^{n-1}
63	$2^{57.68}$	2^{n-1}
193	$2^{186.07}$	2^{n-1}
255	$2^{247.67}$	2^{n-1}
513	$2^{504.66}$	2^{n-1}

Table 4: Attack complexities when using the multivariate approach.

where each $f_i : (\mathbb{F}_2)^n \to \mathbb{F}_2$ is a degree-1 polynomial. As shown in the following, the complexity of solving such a system of n linear equations in n variables can be given as the complexity of Gaussian elimination, which is

$$T_3 \in \mathcal{O}(n^3)$$

bit operations, and thus well within the allowed time frame for the attack.

Low-Degree Polynomial. Here we briefly analyze the cost of solving a polynomial system over $\mathbb{F}_2^n$ of algebraic degree d. For $d = 1$, this system is linear and can be solved in a number of bit operations in $\mathcal{O}(n^3)$ with Gaussian elimination. If $d > 2$, the best strategy may be to solve the system using a dedicated brute-force algorithm, as presented in [56]. For optimal choices of algorithm parameters[17], this is expected to require $4d \cdot \log(n) \cdot 2^n$ bit operations. In many instances, it may therefore be less costly to brute-force the polynomial system in this way than brute-forcing the encryption system directly. Lastly, techniques of solving quadratic polynomial systems (i.e., $d = 2$) have received extensive study from the cryptographic community. Under some assumptions on the polynomial system, [55] estimates the asymptotic time complexity of this problem to be in $\mathcal{O}\left(2^{0.841n}\right)$.

D.3 Summary of the Attack

In total, following steps are necessary.

1. *(Online)* Request the decryptions of 2^{n-1} chosen ciphertexts.
2. *(Offline)* For each of the obtained plaintexts, evaluate a single round of MiMC in the encryption direction and keep the key bits as variables.
3. *(Offline)* Solve the resulting system of n linear equations in n unknown key variables.

[17] Here, we mean optimality with respect to the time complexity. In practice, the authors note that the optimal choice depends on the available hardware (see [56, Sect. 5]).

D.4 Attack Complexity

Note that since the algebraic degree of one round is only 2, we can obtain at most n different monomials for each bit position (namely, degree-1 monomials in the key bits) if we directly substitute the plaintext bits with the concrete values obtained from our oracle. Since we can therefore omit the computation of all monomials of the form $k_i \cdot k_j$, where $i \neq j$ and $i \in [1, n], j \in [1, n]$, the symbolic evaluation of a single round of MiMC is similarly expensive as the direct evaluation, and we approximate this complexity by n^2. Building the sums adds an additional $\leq n^2$ bit operations, and due to the number of input vectors we thus arrive at a total complexity of

$$\mathcal{C}_{\mathcal{A}} \leq 2^{n-1} \left(2n^2\right)$$

bit operations. Optimistically assuming[18] that we need only n^2 bit operations for a direct evaluation of $f(x) = x^3$, the cost of exhaustively searching for the correct key is around

$$\mathcal{C}_{\mathcal{E}} = 2^n \cdot \left(n^2 \cdot \left\lceil \frac{n}{\log_2(3)} \right\rceil\right)$$

bit operations, and $\mathcal{C}_{\mathcal{A}} < \mathcal{C}_{\mathcal{E}}$.

Finally, the number of chosen ciphertexts required for the zero sum results in a data complexity of 2^{n-1}, and the memory complexity is negligible at n^2, both for the symbolic evaluations and for the final solving step involving an $n \times n$ matrix over $\mathbb{F}_2$. The final complexities are shown in Table 4.

Supplementary Material References

54. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP Modeling for (Large) S-boxes to Optimize Probability of Differential Characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017)
55. Bardet, M., Faugère, J., Salvy, B., Spaenlehauer, P.: On the complexity of solving quadratic boolean systems. J. Complexity **29**(1), 53–75 (2013)
56. Bouillaguet, C., Chen, H., Cheng, C., Chou, T., Niederhagen, R., Shamir, A., Yang, B.: Fast Exhaustive Search for Polynomial Systems in F_2. In: CHES 2010. LNCS, vol. 6225, pp. 203–218 (2010)
57. Sun, L., Wang, W., Wang, M.: Automatic Search of Bit-Based Division Property for ARX Ciphers and Word-Based Division Property. In: ASIACRYPT 2017. LNCS, vol. 10624, pp. 128–157 (2017)
58. Sun, L., Wang, W., Wang, M.: Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. IET Information Security **14**(1), 12–20 (2020). https://doi.org/10.1049/iet-ifs.2018.5283, `https://doi.org/10.1049/iet-ifs.2018.5283`
59. Wang, Q., Grassi, L., Rechberger, C.: Zero-Sum Partitions of PHOTON Permutations. In: CT-RSA 2018. LNCS, vol. 10808, pp. 279–299 (2018)

[18] We ignore the cost for key additions and constant additions, as well as the memory (or additional computation time) needed to store (or compute) the round constants.

60. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers. In: ASIACRYPT 2016. LNCS, vol. 10031, pp. 648–678 (2016)