Report on the Security of STARK-friendly Hash Functions (Version 2.0)

Anne Canteaut, Tim Beyne, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Yu Sasaki, Yosuke Todo, et al.

Report on the Security of STARK-friendly Hash Functions (Version 2.0)

Anne Canteaut[1] (ed.), Tim Beyne[2], Itai Dinur[3], Maria Eichlseder[4], Gregor Leander[5], Gaëtan Leurent[1], María Naya Plasencia[1], Léo Perrin[1], Yu Sasaki[6], Yosuke Todo[6,5], Friedrich Wiemer[5]

[1]Inria, France
[2]Imec-COSIC, KU Leuven, Belgium
[3]Department of Computer Science, Ben-Gurion University, Israel
[4]Graz University of Technology, Austria
[5]Ruhr-Universität Bochum, Germany cryptosolutions, Germany
[6]NTT Secure Platform Laboratories, Japan

Version 2: 26/02/2020

Contents

1 Introduction

The emergence of cryptographic protocols with advanced functionalities, such as fully-homomorphic encryption, multi-party computation or new types of proof systems, has led to a strong demand for new symmetric primitives, offering good performance in the context of these specific applications. Indeed, the usual criteria which rule the design of symmetric primitives are usually not appropriate when we focus on these applications. For instance, the cost of the homomorphic evaluation of a symmetric primitive is mainly determined by its multiplicative size and depth (see e.g. [ARS+15]). Similarly, the ZK-STARK protocol [BBHR18], which is expected to be deployed on top of the Ethereum blockchain within the next year, uses as a building-block a collision-resistant hash function, and the performance of the proof system highly depends on the number of arithmetic operations required for describing the hash function (see [AAB+19] for details). In other words, STARK-friendly hash functions (SFH) are hash functions which are specified as a sequence of low-degree polynomials or low-degree rational maps over a finite field.

Therefore, several new ciphers or hash functions have been proposed in the last five years for these advanced protocols. They include several FHE-friendly symmetric encryption schemes like LowMC [ARS+15], FLIP [MJSC16], Kreyvium [CCF+18] or Rasta [DEG+18], some MPC-friendly block ciphers like MiMC [AGR+16] and its variants [AGP+19a, GKK+19], and some primitives dedicated to ZK-STARK proof systems like the functions from the Marvellous family, including Jarvis, Friday [AD18], Vision and Rescue [AAB+19].

However, all these primitives are very innovative constructions and the implementation constraints which rule their designs may have introduced some unexpected weaknesses. This was the case of LowMC which has been broken a few weeks after its publications [DLMW15, DEM16, RST18]. More recently, a practical attack against Jarvis has been mounted [ACG+19], showing that these types of designs are probably not mature enough for practical applications and require a more in-depth security evaluation.

Actually, it is not surprising that the very first attempts to design a primitive with some specific properties are broken. We had a similar experience during the Nessie competition in 2000-2003[1] for instance, where all submitted stream ciphers were broken. This however served as a fruitful lesson before the eSTREAM project[2], which recommended in 2008 several stream ciphers which are still considered secure. Clearly, new implementation constraints and new design principles usually open the door to new vulnerabilities and new types of attacks. Several design attempts combined with an important cryptanalytic effort are then mandatory to reach a secure primitive. This is why, since the end of the 90s and the AES competition, symmetric primitives are now standardized after a public international competition which includes a cryptanalytic effort from the whole community during four or five years (e.g. AES in 1997-2000, CRYPTREC in 2000-2003, eSTREAM in 2004-2008, SHA-3 in 2008-2012 or CAESAR in 2014-2019). A similar effort of several years is therefore needed for specifying secure MPC-friendly or STARK-friendly primitives.

The aim of this report is to present some preliminary analysis on the security of three families of STARK-friendly hash functions, for some sets of parameters specified by StarkWare. This analysis relies on the ideas, directions and results developed during a 3-day meeting held in Paris on November 2019. This effort enables us to exhibit some severe weaknesses in some of the proposed primitives. However, the fact that no attack has been obtained against some other algorithms does not imply that these algorithms are secure, but only that they cannot be easily broken with the state-of-the-art cryptanalytic techniques. Obviously, a much longer and intensive work is needed to assess their security level: concrete primitives are usually recommended after 4 years, not a few days or weeks,

[1] https://en.wikipedia.org/wiki/NESSIE
[2] https://www.ecrypt.eu.org/stream/

of analysis.

Organization of this report. The next section presents the STARK-friendly hash functions which are analyzed in the report and details the sets of parameters which have been chosen, based on the performance analysis provided by StarkWare. Section 3 presents the criteria used for evaluating the security of these hash functions. It especially discusses the cryptographic relevance of the notion of distinguishers for the inner permutation of the sponge construction. The following three sections then present the results of our analysis on the three families of functions proposed by StarkWare: Section 4 focuses on GMiMC, Section 5 on HadesMiMC and its two variants, Poseidon and Starkad, and Section 6 on two members of the Marvellous family, namely Vision and Rescue. The report ends with a comparison of the security levels offered by these hash functions.

2 The SFH Contenders

2.1 Hash functions derived from the sponge construction

Three families of hash functions have been proposed by StarkWare as candidates for SFH recommendations and have been included in the STARK-friendly hash challenges[3]. These three families all follow the sponge construction [BDPV07, BDPV08] depicted on Figure 1, and they only differ in the choice of the underlying inner permutation.

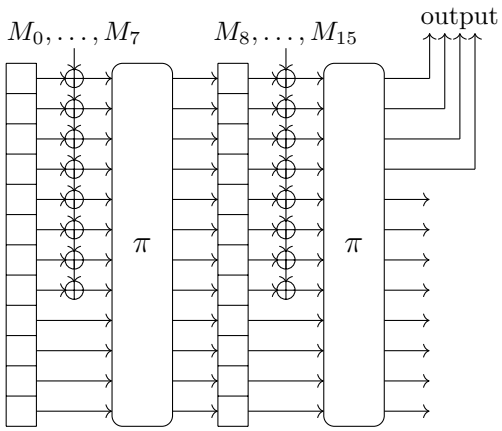


Figure 1: Sponge construction with inner permutation π, internal state with $t = 12$ words and capacity $c = 4$.

In the following, we extensively use the following notation: the sponge operates on a state composed of t elements in a finite field $\mathbb{F}_q$. The main parameter which determines the security level of the sponge construction with respect to generic attacks (i.e., attacks which do not exploit any detail of the inner permutation) is its *capacity* c, as well as the size of the underlying alphabet $\mathbb{F}_q$. Namely, a random sponge whose capacity consists of c elements in $\mathbb{F}_q$ provides a generic security level of $\frac{c}{2}\log_2 q$ queries both for collision and (second)-preimage resistance [BDPV07].

The different members in each of these families are determined by the triple (c, t, q) representing respectively the number of words in the capacity, the number of words in the state and the field size. In the following, when referring to practical examples, we will focus on the values (c, t, q) considered in the StarkWare challenges given in Table 1. To

[3] https://starkware.co/hash-challenge/

each triple (c, t, q) correspond two variants: over a prime field and over a binary field, and the exact values of q are detailed in Table 1.

Table 1: Parameters proposed for the permutation and sponge construction.

Security level	$\log_2 q$	q (prime)	q (binary)	c	t	Variant
128 bits	**64**	$2^{61} + 20 \times 2^{32} + 1$	2^{63}	**4**	**12**	**128-d**
	128	$2^{125} + 266 \times 2^{64} + 1$	2^{125}	2	4	128-a
				2	12	128-c
	256	$2^{253} + 2^{199} + 1$	2^{255}	1	3	128-b
				1	11	128-e
256 bits	128	$2^{125} + 266 \times 2^{64} + 1$	2^{125}	4	8	256-a
				4	14	256-b

It is worth noticing that the STARK-friendly hash challenges also include some members of these three families aiming at lower security levels, namely 45 bits and 80 bits, but these functions will not be considered in this report because they do not guarantee a sufficient security level.

2.2 Candidates for the inner permutations

The inner permutations for each of these families of sponges are the following ones:

- **GMiMC** designed by Albrecht *et al.*, where only the variant $\mathsf{GMiMC}_{\mathsf{erf}}$ over a prime field is considered, as defined in [AGP+19a, AGP+19b];
- **HadesMiMC** proposed by Grassi *et al.* [GKK+19, GLR+19], for which two versions are distinguished depending on the characteristic of the underlying field: Starkad over a field of characteristic 2, and Poseidon over a prime field;
- **Marvellous** designed by Aly *et al.* [AAB+19], which consists of two different permutations: Vision over a field of characteristic 2, and Rescue over a prime field.

All these permutations are recent evolutions of a block cipher named MiMC designed by Albrecht *et al.* in 2016 [AGR+16], and offer much more flexibility than the original construction.

2.3 Round constants and MDS matrices

Most of these constructions actually correspond to STARK-friendly block ciphers, which implies that they are defined as families of permutations parametrized by a secret key. More precisely, these three designs consist of several iterations of the same round function, and a round-key derived from the secret key by a key-scheduling algorithm is added to the internal state between any two consecutive rounds. In the previously described setting, the inner permutation of the sponge function is a single instance of the block cipher, i.e., the permutation corresponding to a specific and randomly chosen secret key. However, the key-scheduling does not have any cryptographic relevance in the case of a single permutation and it may induce some performance overhead. Then, the round-keys are replaced by fixed independent round-constants which are computed by the same method for all three designs: the i-th round-constant is derived from the image under SHA-256 of the string formed by the name of the primitive and by the index i, e.g. $\mathsf{sha256}('\text{Hades}i')$. This choice guarantees that that the rounds are not too similar (which avoids classical

attacks like slide attacks), and also that there is no strong regularity within the set of all round-constants (like sparse round-constants).

Also, both HadesMiMC and Marvellous are Substitution-Permutation Networks with a linear diffusion layer defined by an MDS matrix. In both cases, this matrix is chosen as a Cauchy matrix, i.e, its entry at Row i and Column j is defined by

$$\frac{1}{x_i - y_j}$$

where $(x_1, \ldots, x_t)$ and $(y_1, \ldots, y_t)$ are randomly chosen constants generated in a similar way as the round-constants. For the StarkWare challenge, such MDS matrices are then generated until one with no eigenvalue in the field is found.

2.4 Performance

Performance in terms of trace size, proving time, and verification cost, is an essential criterion for choosing a STARK-friendly hash function. The implementation results presented by StarkWare show that, for each of the three previously described families of hash functions, the variant 128-d (for the target 128-bit security) is by far the most efficient. For this reason, this report mainly focuses on this member in the three families, i.e., on sponges whose internal state consists of $t = 12$ words in a finite field $\mathbb{F}_q$ of size close to 2^{64} and with capacity $c = 4$. More precisely, the finite field $\mathbb{F}_q$ corresponds to $\mathbb{F}_{2^{63}}$ in the case of a binary field and to $\mathbb{F}_p$ with $p = 2^{61} + 20 \times 2^{32} + 1$ in the case of a prime field.

It is also worth noticing that, in terms of performance and suitability, prime fields are more STARK-friendly than binary fields for a given size.

3 Security Evaluation of Hash Functions: Methodology

It is expected that none of the three hash functions we evaluate is extremely weak and that no practical attack could be found after a few weeks only. In this context, the relevant information for estimating the security level of a symmetric primitive is its security margin with respect to several types of attacks and weaknesses.

3.1 STARK-friendly hash challenges

As previously mentioned, several challenges have been launched by StarkWare in order to evaluate and compare the security of these hash functions. The objective of the challenge is to *exhibit a collision* for these hash functions, for the parameters listed in Table 1 aiming at 128-bit and 256-bit security. Some challenges with a weaker security level, namely 45 bits and 80 bits, are also proposed. However, it is rather difficult to estimate the security margin offered by the hash functions aiming at 128-bit security from these easier challenges. Indeed, the easier challenges only reduce the parameters in the sponge construction, then making a generic brute-force attack feasible. But, since the involved hash functions are based on full versions of the inner permutations and not on weakened variants (typically with fewer rounds), it is almost impossible to exploit potential weaknesses of the permutation in a concrete attack in this setting. The only property of the inner permutation used so far in the broken challenges for 45-bit security is the fact that the implementation cost of some of these permutations (especially GMiMC) can be reduced in order to speed up the brute-force attack [Udo19].

3.2 Attacks on weakened variants

The aim of this report is to identify whether weakened variants of the hash functions can be attacked with a lower complexity than the complexity of the generic attacks. Typical *weakened variants* are obtained by reducing the number of rounds of the inner iterative permutation. Also, some building-blocks (e.g. the MDS matrix) can be slightly modified to measure whether some unexpected weaknesses may appear in the underlying construction in some specific cases.

3.3 Distinguishers for the inner permutation

Obviously, the primary cryptanalytic goal is to exhibit collision or preimage attacks on some weakened variants of the three hash functions. Finding such attacks requires, as a preliminary work, exhibiting some specific properties of the inner permutation of the hash function which may be exploited in an attack. Therefore, this report also describes some specific properties, like the existence of differentials or some properties of algebraic nature, which hold for the considered permutations and not for a randomly chosen permutation of $\mathbb{F}_q^t$. In some cases, we have not found any concrete attack so far based on the exhibited property. However, the existence of a property which distinguishes a given cryptographic function from an ideal function of the same size is commonly considered as a weakness.

This issue has been discussed in several contexts, e.g., for evaluating the security of hash functions [AMPH14, Page 19] or through the notion of known-key distinguishers against block-ciphers [KR07]. For instance, in the case of hash functions, it is explicitly stated in [FSK10] that *an attack on a hash function is a non-generic method of distinguishing the hash function from an ideal hash function.* Obviously, it is difficult to provide a formal and rigorous definition of a distinguisher on a given function. Also, any cryptographic function admits a trivial distinguisher since it has a compact expression which makes its implementation feasible, while it is not the case of a randomly chosen function. In our context, since all the inner permutations we consider use randomly-chosen round-constants, *distinguishers* (aka *structural distinguishers*) are related to the classical notion of *indistinguishability* of the family of keyed permutations, obtained when the round-constant sequence varies. For HadesMiMC and Marvellous which both use randomly generated MDS matrices, the family of permutations can even be further extended by taking into account all round-constants and all Cauchy MDS matrices.

Another question is whether the existence of distinguishers on the inner permutation π impacts the security of the whole hash function. While a distinguisher on π cannot always be turned into a distinguisher for the hash function, it clearly invalidates the security arguments provided by the indifferentiability proof of the sponge construction [BDPV08]. Indeed, proving that a hash function following the sponge construction has $\frac{c}{2}\log_2 q$ security bits requires that the underlying inner permutation is ideal. For this reason, the authors of Keccak first advocate following the so-called *hermetic sponge strategy* [BDPV09, Page 13], i.e. using the sponge construction with an inner permutation that should not have any structural distinguisher (except the existence of a compact description). It is worth noticing that the authors of most of the designs we consider claim that their proposal follow the hermetic sponge strategy, which means that their security claims assume that the inner permutation cannot be distinguished from an ideal permutation. In this sense, any distinguisher on the inner permutation of one of these candidates invalidates the designers' security claims. Even if it does not imply the existence of an attack against the full hash function, it reveals some non-ideal behavior which was not expected by the designers.

A related question is to understand to which extent the sponge construction differs from an ideal hash function when the inner permutation differs from an ideal permutation. One may think of this notion as a form of robustness: even if a weakness of a certain

type is discovered on the inner permutation in the future, the corresponding hash function would remain almost equally ideal. During the SHA-3 competition, several works try to adapt the original indifferentiability proof of some constructions to the case where the inner permutation is not ideal but randomly chosen within the set of permutations having a specific structural property, see e.g. [BCC+09, BFL11]. But such proofs are quite involved and it usually seems difficult to guarantee an appropriate security level in the presence of structural distinguisher for the inner permutation.

4 GMiMC

4.1 Description

GMiMC is a family of block ciphers designed by Albrecht *et al.* in 2019 [AGP+19a] based on different types of Feistel network using x^3 over the field corresponding to the branch alphabet as a round function. Among the several variants proposed by the designers, the one chosen by StarkWare and analyzed in this report is the variant using an unbalanced Feistel network with an expanding round function, named $\mathsf{GMiMC}_{\mathsf{erf}}$ (see Figure 2). For the sake of simplicity, since the other variants are not studied in the report, this particular variant will be called GMiMC. In the whole paper, the rounds (and round constants) are numbered starting from 1, and the branches are numbered from 1 to t where Branch 1 is the leftmost branch. A specificity of GMiMC is that the designers' security claims concern the primitive instantiated over a prime field. They mention that, "even if GMiMC can be instantiated over $\mathbb{F}_{2^n}$, [they] do not provide the number of rounds to guarantee security in this scenario".

In the block-cipher setting with key size equal to $n = \log_2 q$ bits, the key schedule is trivial, i.e. the master key is added to the input of the cube function at every round. It is worth noticing that this very simple key schedule has been shown to be a major weakness. Indeed, due to the existence of a related-key differential characteristics with probability 1, a variant of the slide attack allows a key recovery in $2^{n/2}$ operations independently from the number of rounds [Bon19]. However, it seems difficult to leverage the underlying property in the hash-function setting we are focusing on.

Algorithm 1 GMiMC with block-size t and R rounds.

Input: $(x_0, \ldots, x_{t-1}) \in \mathbb{F}^t$
  for i from 0 to $(R-1)$ **do**
    $y \leftarrow (x_0 + \mathsf{RC}_i)^3$
    $(x_0, \ldots, x_{t-1}) \leftarrow (x_1 + y, x_2 + y, \ldots, x_{t-1} + y, x_0)$
  end for
Output: $(x_0, \ldots, x_{t-1})$

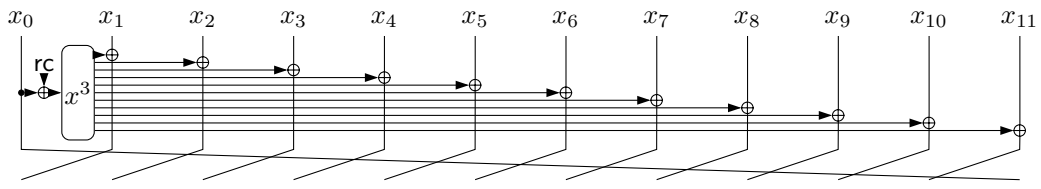


Figure 2: One round of the GMiMC permutation with $t = 12$.

For the parameters we focus on, namely $t = 12$ and $\mathbb{F}_q = \mathbb{F}_p$ with $p = 2^{61} + 20 \times 2^{32} + 1$, i.e. $n = 61$, the number of rounds is $R = 101$. Note that it is unclear from the discussion in [AGP+19b, Page 10] whether this number of rounds aims at avoiding distinguishers of complexity less than q^t, or less than q only.

4.2 Integral distinguishers on the full GMiMC

4.2.1 Integral attacks over non-binary fields

The notion of *integral attacks* has been introduced by Knudsen and Wagner [KW02] and captures several variants including saturation attacks and higher-order differential attacks. These attacks have been used for cryptanalyzing many ciphers, but to our best knowledge, all of them operate on a binary field. Indeed, the main property behind these attacks is that, for any $F : \mathbb{F}_2^m \to \mathbb{F}_2^m$ and for any affine subspace $V \subset \mathbb{F}_2^m$,

$$\sum_{x \in V} F(x) = 0$$

when $\deg F < \dim V$. This comes from the fact that the sum of the images by F of all inputs in V corresponds to a value of a derivative of F of order $(\dim V)$ [Lai94]. It follows that this derivative has degree at most $(\deg(F) - \dim V)$ and thus vanishes when $\deg F < \dim V$. It is then possible to *saturate* some input bits of F and to use as a distinguishing property the fact that the output bits are balanced, i.e. they sum to zero. The fact that the sum over all $x \in V$ of $F(x)$ corresponds to the value of a higher-order derivative does not hold anymore in odd characteristic, and the same technique cannot be applied directly.

Higher-order differentials over $\mathbb{F}_q$ then need to use a generalized notion of differentiation as analyzed in [SWMSP17] (see also [AP11]). However, we can show that for the particular case of saturation attacks, the same technique can be used in the general case of a field $\mathbb{F}_q$ – even in odd characteristic. Indeed, we can exploit the following result.

Proposition 1. *For any $F : \mathbb{F}_q \to \mathbb{F}_q$ with $\deg(F) < q - 1$,*

$$\sum_{x \in \mathbb{F}_q} F(x) = 0 \,.$$

Proof. The result is due to following well-known property: for any exponent k with $1 \leq k \leq q - 2$,

$$\sum_{x \in \mathbb{F}_q} x^k = 0 \,.$$

Moreover, when $k = 0$, we have $\sum_{x \in \mathbb{F}_q} x^0 = q = 0$. □

Proposition 1 can be generalized to the multivariate case, i.e. to functions from $\mathbb{F}_q^k$ into $\mathbb{F}_q$, which can be expressed as polynomials in the ring

$$\mathbb{F}_q[x_1, \ldots, x_k] / \left(x_1^q - x_1, \ldots, x_k^q - x_k\right).$$

Corollary 1. *For any $F : \mathbb{F}_q^t \to \mathbb{F}_q$ with $\deg(F) < k(q-1)$ and any affine subspace $V \subseteq \mathbb{F}_q^t$ of dimension at least k,*

$$\sum_{x \in V} F(x) = 0 \,.$$

Proof. Let V be an affine space of dimension $\kappa \leq k$ and A an affine permutation over $\mathbb{F}_q^t$ such that $A(V) = \{(y, 0, \ldots, 0), y \in \mathbb{F}_q^\kappa\}$. Then,

$$\sum_{x \in V} F(x) = \sum_{x \in V} (F \circ A^{-1})(A(x)) = \sum_{y_1, \ldots, y_\kappa \in \mathbb{F}_q} (F \circ A^{-1})(y_1, \ldots, y_\kappa, 0, \ldots, 0) = 0$$

since $\deg(F \circ A^{-1}) = \deg F < k(q-1)$. □

Based on this observation, a saturation attack with data complexity q^k can be mounted whenever the degree of F as a polynomial over $\mathbb{F}_q$ is strictly less than $k(q-1)$, even if $\mathbb{F}_q$ is a field of odd characteristic.

4.2.2 Integral distinguishers over multiplicative subgroups

We generalize the notion of integral distinguishers to multiplicative subgroups using the following property.

Proposition 2. *Let $\mathbb{G}$ be a multiplicative subgroup of $\mathbb{F}_q^\times$. For any $F : \mathbb{F}_q \to \mathbb{F}_q$ such that $\deg(F) < |\mathbb{G}|$,*

$$\sum_{x \in \mathbb{G}} F(x) - F(0) \cdot |\mathbb{G}| = 0 \, .$$

This is a strict generalization of Proposition 1, for which $|\mathbb{G}| = q - 1$.

Proof. The result is a direct consequence of the following well-known property: for any exponent k with $1 \leq k \leq |\mathbb{G}| - 1$,

$$\sum_{x \in \mathbb{G}} x^k = 0 \, .$$

Moreover, when $k = 0$, we have $\sum_{x \in \mathbb{G}} x^0 = |\mathbb{G}|$. □

We also note that Corollary 1 can be adapted to multiplicative subgroups in a straightforward manner. The power of summing over multiplicative subgroups (rather than over the entire field $\mathbb{F}_q$) comes from the fact that if $\mathbb{F}_q$ contains small multiplicative subgroups (as for the fields used for the concrete instances specified in Table 1), the complexity of the attacks may be fine-tuned and significantly reduced.

4.2.3 Integral distinguisher on GMiMC

Using Corollary 1, we can exhibit a distinguisher for $(3t - 4 + \lfloor \log_3(q-2) \rfloor)$ rounds of GMiMC. A remarkable property is that this distinguisher holds for any finite field. It is obtained by saturating a single branch of the Feistel network and consequently has data complexity q. Indeed, we choose a set of inputs where the $(t-2)$ leftmost branches are inactive, while the rightmost branch is determined by the value of Branch $(t-1)$. More precisely, we consider a set of inputs of the form

$$\mathcal{X} = \{(\alpha_1, \ldots, \alpha_{t-2}, x, f(x)) \mid x \in \mathbb{F}_q\} \tag{1}$$

where the α_i are arbitrary constants in $\mathbb{F}_q$ and f is defined by

$$f(x) = -\Big(x + \sum_{i=1}^{t-2} \beta_i + \mathsf{RC}_{t-1}\Big)^3 - x - 2\sum_{i=1}^{t-2} \beta_i - \mathsf{RC}_{t-1} - \mathsf{RC}_t$$

and $\beta_1, \ldots, \beta_{t-2}$ are constant values derived from $\alpha_1, \ldots, \alpha_{t-2}$ by

$$\beta_1 = (\alpha_1 + \mathsf{RC}_1)^3 \text{ and } \beta_{i+1} = \Big(\alpha_{i+1} + \sum_{j=1}^{i} \beta_j + \mathsf{RC}_{i+1}\Big)^3 \, .$$

Let us first consider the first $(t-2)$ rounds. We observe that, at Round i, $1 \leq i \leq t-2$, the output of the Sbox corresponds to β_i and is added to all branches except the leftmost branch of the input. It follows that the output of Round $(t-2)$ corresponds to

$$(x + \textstyle\sum_{i=1}^{t-2} \beta_i, f(x) + \sum_{i=1}^{t-2} \beta_i, \gamma_1, \ldots, \gamma_{t-2})$$

where $(\gamma_1, \ldots, \gamma_{t-2})$ are constants (see Figure 3). Therefore, if x' denotes the value of Branch 1, i.e., $x' = x + \sum_{i=1}^{t-2} \beta_i$, we have that Branch 2 corresponds to

$$f\left(x' - \sum_{i=1}^{t-2} \beta_i\right) + \sum_{i=1}^{t-2} \beta_i = -\left(x' + \mathsf{RC}_{t-1}\right)^3 - x' - \mathsf{RC}_{t-1} - \mathsf{RC}_t \ .$$

The inputs of Round t are then

$$\{(-x' - \mathsf{RC}_t - \mathsf{RC}_{t-1}, \gamma_1 + (x' + \mathsf{RC}_{t-1})^3, \ldots, \gamma_{t-2} + (x' + \mathsf{RC}_{t-1})^3, x') \mid x' \in \mathbb{F}_q\}$$

and the inputs of Round $(t+1)$ are

$$\{(\gamma_1, \ldots, \gamma_{t-2}, x' - (x' + \mathsf{RC}_{t-1})^3, -x' - \mathsf{RC}_t - \mathsf{RC}_{t-1}) \mid x' \in \mathbb{F}_q\} \ .$$

The following $(t-2)$ rounds do not activate the Sbox, implying that the input set at Round $(2t-1)$ has the form

$$\{(x' - (x' + \mathsf{RC}_{t-1})^3 + \delta_1, -x' + \delta_2, \delta_3, \ldots, \delta_t) \mid x' \in \mathbb{F}_q\} \tag{2}$$

for some fixed values $\delta_1, \ldots, \delta_t$ determined by the constants. Each coordinate of this input word can then be seen as a q-ary polynomial in x' of degree at most three. It follows that, after r additional rounds, the set (2) is transformed into a set of elements $(z_1, \ldots, z_t)$, whose coordinates have degree at most 3^{r+1}. Prop. 1 then implies that all z_i are balanced if $3^{r+1} \leq q-2$, i.e., if $r \leq \lfloor \log_3(q-2) \rfloor - 1$.

Adding $(t-1)$ rounds. We can add some more rounds by using the following relation over $(t-1)$ rounds of GMiMC.

Proposition 3. *Let $(x_1, \ldots, x_t)$ and $(y_1, \ldots, y_t)$ denote the input and output of $(t-1)$ rounds of* GMiMC.

$$\sum_{i=2}^{t} y_i - (t-2)y_1 = \sum_{i=1}^{t-1} x_i - (t-2)x_t \ . \tag{3}$$

Proof. Let $(x_1^\ell, \ldots, x_t^\ell)$ denote the input of Round ℓ. It can be observed that, for any $i, j \in \{1, \ldots, t-1\}$,

$$x_i^\ell = x_{i+1}^{\ell-1} + (x_j^\ell - x_{j+1}^{\ell-1}) \text{ and } x_t^\ell = x_1^{\ell-1} \ .$$

It follows that, for any j, $1 \leq j \leq (t-1)$,

$$\sum_{i=1}^{t} x_i^\ell - (t-1)x_j^\ell = \sum_{i=1}^{t} x_i^{\ell-1} - (t-1)x_{j+1}^{\ell-1} \ .$$

By applying this equality $(t-1)$ times, we deduce (3). □

From the previous proposition, we deduce that after a total of

$$R = 3t - 4 + \lfloor \log_3(q-2) \rfloor \text{ rounds},$$

the output $(v_1, \ldots, v_t)$ of GMiMC satisfies $\sum_{i=2}^{t} v_i - (t-2)v_1 = \sum_{i=1}^{t-1} z_i - (t-2)z_t$, which is a polynomial in x' of degree at most $(q-2)$. This leads to a distinguisher with complexity q on R rounds, i.e., 70 rounds for the parameters we focus on.

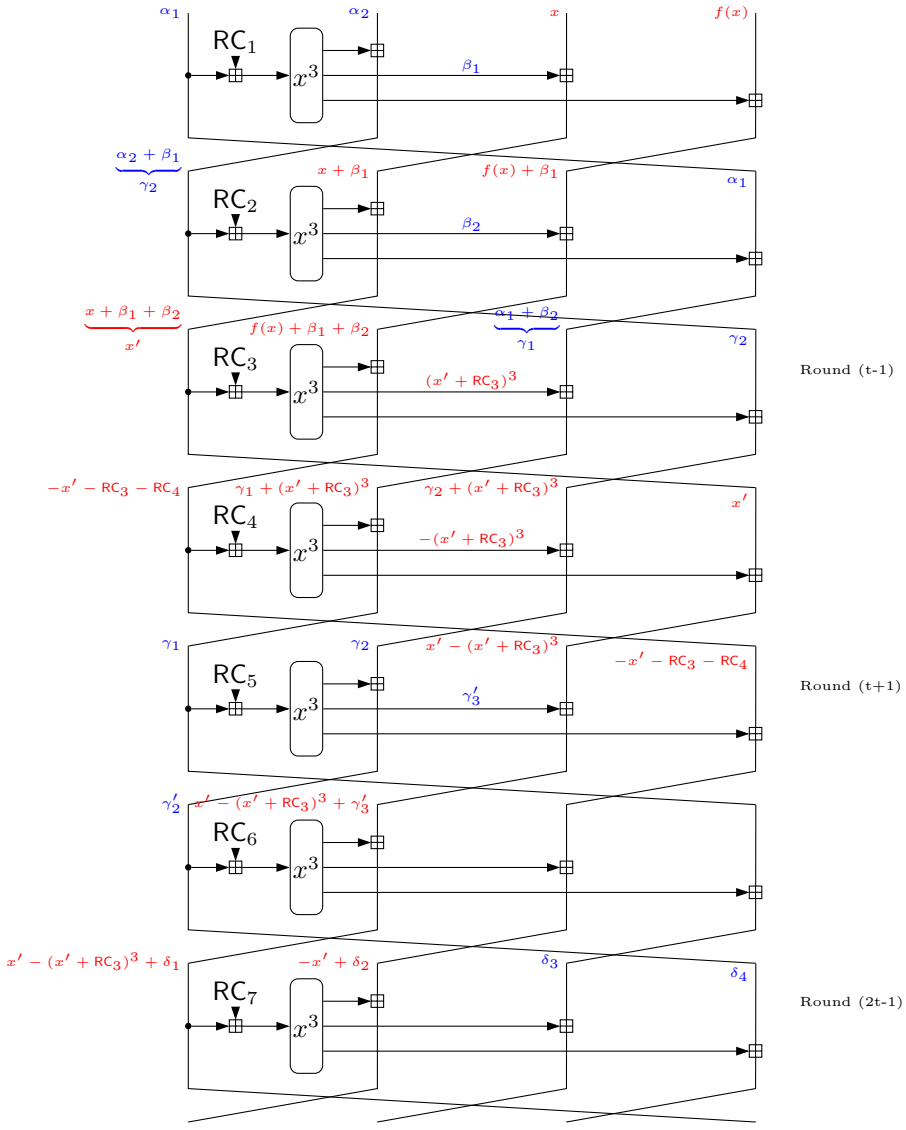


Figure 3: First rounds of the integral distinguisher on GMiMC (with $t = 4$) as described in Section 4.2.3.

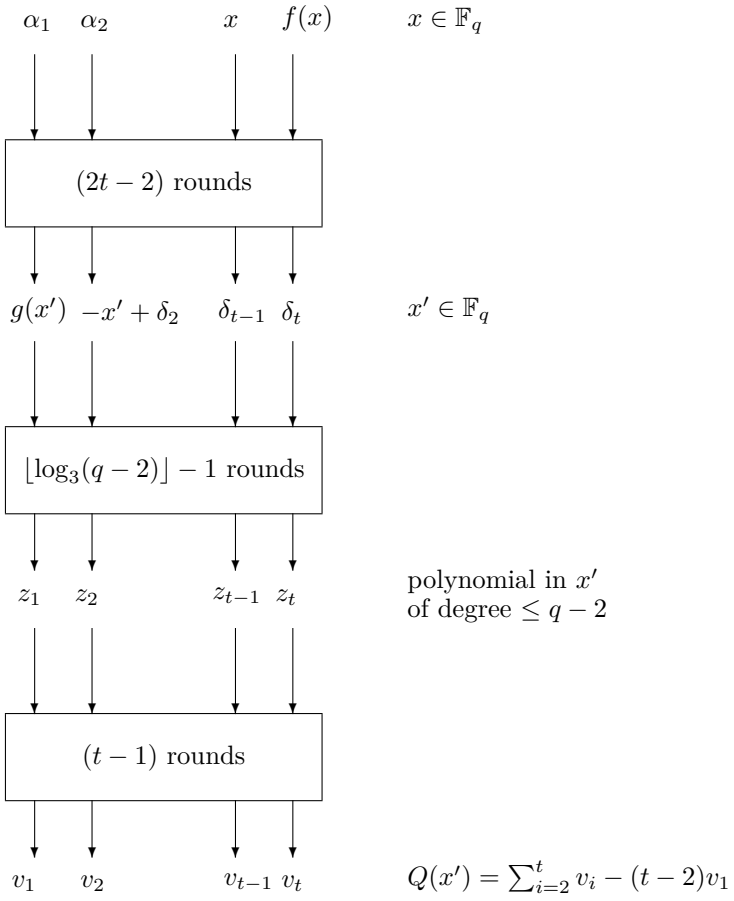


Figure 4: Integral distinguisher on GMiMC.

4.2.4 Zero-sum distinguishers on the full permutation

Saturating a single branch. Since we are analyzing a permutation (or a family of permutations parameterized by the round-constants), there is no secret material involved in the computation, implying that a distinguisher can be built from some internal states in the middle of the primitive, not only from inputs and outputs, exactly as in the *known-key setting* for block ciphers [KR07]. This leads to *zero-sum distinguishers*, which were introduced by Aumasson and Meier [AM09] and exhibited for several hash functions, including SHA-3 [AKK+10, BCD11].

The previously described distinguisher can be extended by $(t-2+\lfloor \log_3(q-2) \rfloor)$ rounds backwards. This is realized by choosing the internal states after $(t-2+\lfloor \log_3(q-2) \rfloor)$ rounds in $\mathcal{X}$, as defined by (1). The inverse of one round of GMiMC is still a round of a Feistel network of the same form and it has degree three over $\mathbb{F}_q$. Then, the coordinates $(y_1, \ldots, y_t)$ of the images of the elements in $\mathcal{X}$ by r backward rounds can be seen as univariate polynomials in x with degree at most 3^{r+1}. Exactly as in the forward direction, after $(\lfloor \log_3(q-2) \rfloor - 1)$ rounds, the degree of these polynomials cannot exceed $(q-2)$.

Based on Prop. 3, we can then add $(t-1)$ rounds backwards. Indeed, the input of the first round of the permutation $(u_1, \ldots, u_t)$ is related to the output of Round $(t-1)$, i.e. $(y_1, \ldots, y_t)$, by

$$\sum_{i=2}^{t} y_i - (t-2)y_1 = \sum_{i=1}^{t-1} u_i - (t-2)u_t \ ,$$

and the left-hand term of this equation is a polynomial in x of degree at most $(q-2)$, implying that $\left(\sum_{i=1}^{t-1} u_i - (t-2)u_t\right)$ sum to zero.

Similarly, we can apply the previously described distinguisher in the forward direction, and deduce that the outputs $(v_1, \ldots, v_t)$ of the permutation after $(3t-4+\lfloor \log_3(q-2) \rfloor)$ additional rounds are such that $\left(\sum_{i=2}^{t} v_i - (t-2)v_1\right)$ sum to zero. This leads to a distinguisher with complexity q for a total of

$$4t - 6 + 2\lfloor \log_3(q-2) \rfloor) \text{ rounds,}$$

which is higher than the number of rounds proposed in all StarkWare challenges, except in the case where q exceeds the claimed security level (see Table 2).

Saturating two branches. When $t \geq 4$, it is possible to exhibit a similar distinguisher on more rounds with complexity q^2 by saturating two branches. In this case, we start from Round m in the middle with a set of internal states

$$\mathcal{Y} = \{(\alpha_1, \ldots, \alpha_{t-4}, x, f(x), g(y), y) \mid x, y \in \mathbb{F}_q\}$$

where

$$\begin{aligned} f(x) &= -\Big(x + \sum_{i=1}^{t-4} \beta_i + \mathsf{RC}_{m+t-4}\Big)^3 - x - 2\sum_{i=1}^{t-4} \beta_i - \mathsf{RC}_{m+t-4} - \mathsf{RC}_{m+t-3} \\ g(y) &= (y + \mathsf{RC}_{m-1})^3 - y - \mathsf{RC}_{m-1} - \mathsf{RC}_{m-2} \end{aligned}$$

and $\beta_1, \ldots, \beta_{t-4}$ are defined as before by replacing RC_i by RC_{m+i-1}.

- **Computing forwards.** As depicted on Figure 6, the corresponding set at the input of Round $(m+t-4)$ is then of the form

  $$\{(x', -(x' + \mathsf{RC}_{m+t-4})^3 - x' - \mathsf{RC}_{m+t-4} - \mathsf{RC}_{m+t-3}, \gamma_1(y), \ldots, \gamma_{t-2}(y)) \mid x', y \in \mathbb{F}_q\}$$

  where $(\gamma_1, \ldots, \gamma_{t-2})$ are some values which depend on y only. After two more rounds, we then get some internal states whose $(t-2)$ leftmost branches do not depend on x'.

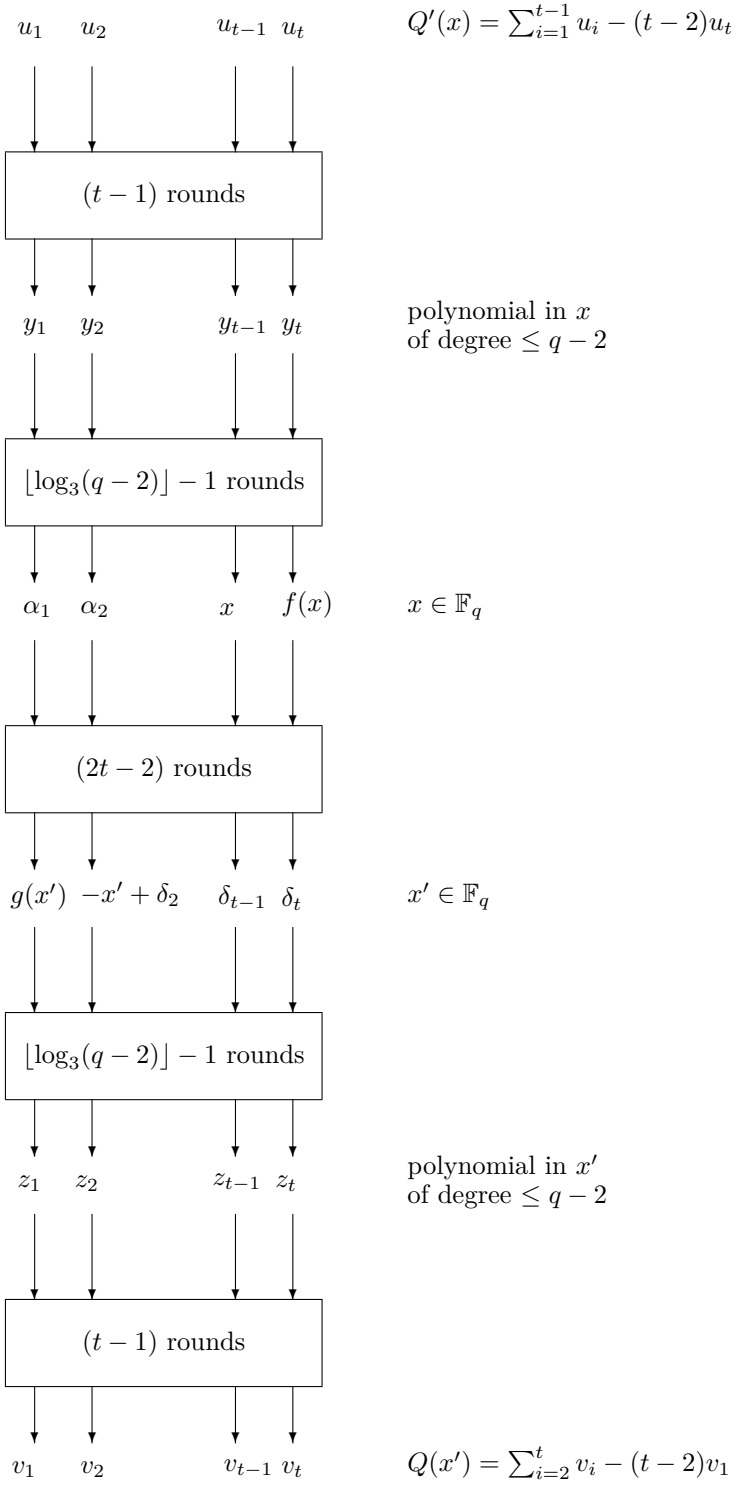


Figure 5: Zero-sum distinguisher on GMiMC with complexity q.

It follows that each coordinate of the input of Round $(m+2t-4)$ is a polynomial in x' and y of degree at most three in x'. After $(\lfloor \log_3(q-2) \rfloor - 1)$ rounds, we get that each coordinate is a polynomial of degree at most $(q-2)$ in x'. Then, with the same technique as before, we can add $(t-1)$ rounds and show that the output of the permutation $(v_1, \ldots, v_t)$ is such that the linear combination $\left(\sum_{i=1}^{t-1} v_i - (t-2)v_t\right)$ sums to zero after $(3t-6+\lfloor \log_3(q-2) \rfloor)$ rounds.

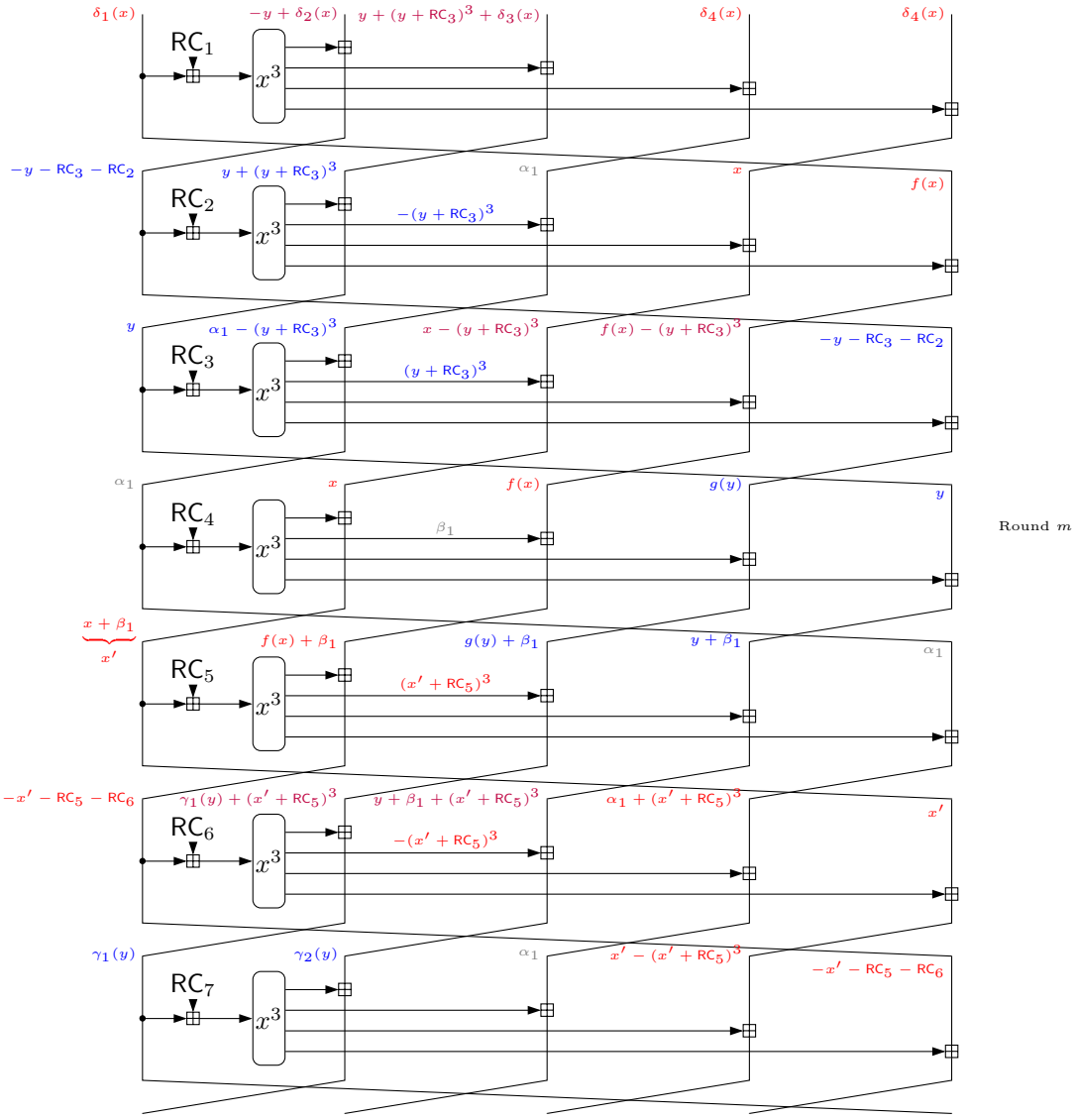


Figure 6: Middle rounds of the zero-sum distinguisher on GMiMC (with $t = 5$) as described in Section 4.2.4.

- **Computing backwards.** Starting from Round m and computing backwards, we get that the input of Round $(m-1)$ is of the form

$$(y, \alpha_1 - (y+\mathsf{RC}_{m-1})^3, \ldots, x - (y+\mathsf{RC}_{m-1})^3, f(x) - (y+\mathsf{RC}_{m-1})^3, -y - \mathsf{RC}_{m-1} - \mathsf{RC}_{m-2})$$

and the input of Round $(m-2)$ equals

$$(-y - \mathsf{RC}_{m-1} - \mathsf{RC}_{m-2}, y + (y + \mathsf{RC}_{m-1})^3, \alpha_1, \ldots, x, f(x)) \ .$$

Then, the following $(t-2)$ rounds do not activate the Sbox, implying that all the coordinates of the input of Round $(m-t)$ are polynomials in x and y of degree at most three in y. We deduce that the input $(u_1, \ldots, u_t)$ of Round $(m-2t+2-\lfloor \log_3(q-2) \rfloor)$ is such that the linear combination $\left(\sum_{i=1}^{t-1} u_i - (t-2)u_t\right)$ sums to zero.

This zero-sum distinguisher then covers a total of

$$5t - 8 + 2\lfloor \log_3(q-2) \rfloor) \text{ rounds,}$$

which is detailed in Table 2 for the relevant parameters.

Table 2: Number of rounds of GMiMC covered by the zero-sum distinguishers of complexity q and q^2.

Security	Parameters		Number of rounds		
	$\log_2 q$	t	Full	ZS with complexity q	ZS with complexity q^2
128 bits	61	12	101	118	128
	125	4	166	166	–
	125	12	182	198	–
	253	3	326	–	–
	253	11	342	–	–
256 bits	125	8	174	182	188
	125	14	186	206	218

4.2.5 Exploiting integral distinguishers over multiplicative subgroups

A noticeable shortcoming of the integral attacks over $\mathbb{F}_q$, as demonstrated by Table 2, is that they do not give any result for primitives over large fields $\mathbb{F}_q$ (for which $\log_2 q \approx 256$). However, by exploiting integral distinguishers over multiplicative subgroups of $\mathbb{F}_q$ (e.g., for the specific choice of $q = 2^{253} + 2^{199} + 1$), we obtain essentially the same results for GMiMC instances with large q as we obtain for instances with small q. For example, in Section 4.2.3 we derived an integral distinguisher on

$$R = 3t - 4 + \lfloor \log_3(q-2) \rfloor \text{ rounds,}$$

with complexity q. By exploiting any multiplicative subgroup of size $|\mathbb{G}| = 2^s$ for $s \leq 199$ when $q = 2^{253} + 2^{199} + 1$, we obtain an integral distinguisher on

$$R = 3t - 4 + \lfloor \log_3(|\mathbb{G}| - 1) \rfloor \text{ rounds,}$$

with complexity $|\mathbb{G}| + 1$. Following this idea, we obtain zero-sum distinguishers with complexity 2^{128} for the two variants with $\log_2 q = 253$. Using a subgroup of size $|\mathbb{G}| = 2^{128}$, this distinguisher covers 166 rounds for $t = 3$ and 198 rounds for $t = 11$ (see Table 3).

Moreover, even for smaller fields, we can fine-tune the size of $\mathbb{G}$ to reduce the complexity of the attack. This is relevant especially for cases where an attack with complexity q can reach more rounds than the ones used by the primitive (which is indeed the case, as shown in Table 2). For example, as derived in Section 4.2.4, we have a zero-sum property for

$$4t - 6 + 2\lfloor \log_3(q-2) \rfloor \text{ rounds,}$$

with complexity q. For the GMiMC variant with $q = 2^{61} + 20 \times 2^{32} + 1$ and $t = 12$, we use a subgroup of size $2^{33} \cdot 167 \cdot 211 \approx 2^{48}$ (which divides $q - 1$), and obtain a zero-sum property for

$$4t - 6 + 2\lfloor \log_3(2^{48} - 1) \rfloor = 102 \text{ rounds},$$

with complexity of about 2^{48} (which covers the full permutation). Similar results are presented for all variants in Table 3.

Table 3: Complexities of the zero-sum distinguishers of GMiMC obtained by saturating a multiplicative subgroup.

Security	Parameters			zero-sum distinguisher	
	$\log_2 q$	t	nb of rounds	nb of rounds	complexity
128 bits	61	12	101	102	2^{48}
	125	4	166	166	2^{125}
	125	12	182	182	2^{111}
	253	3	326	166	2^{128}
	253	11	342	198	2^{128}
256 bits	125	8	174	174	2^{118}
	125	14	186	186	2^{108}

4.3 Impossible differential attacks

The longest impossible differential obtained by the designers of GMiMC covers $(2t - 2)$ rounds [AGP+19b, Page 46]. Also for this type of attack, we observe that the designers' security claim is too optimistic. Indeed, we can construct impossible differentials for $(3t-4)$ rounds, which improves the one observed by the designers by $(t - 2)$ rounds.

The designers focused on the fact that the longest differential trail with probability 1 covers $(t - 1)$ rounds, which is described as $(0, \ldots, 0, \alpha) \overset{t-1 \text{ rounds}}{\longrightarrow} (\alpha, 0, \ldots, 0)$, where α is an arbitrary non-zero difference. Then, by combining two of those trails, they observed that the input difference $(0, \ldots, 0, \alpha)$ and output difference $(\beta, 0, \ldots, 0)$ are impossible differentials on $(2t - 2)$ rounds, where β is an arbitrary non-zero difference that can be equal to α.

$$(0, \ldots, 0, \alpha) \overset{\mathcal{R}^{t-1}}{\longrightarrow} (\alpha, 0, \ldots, 0) \neq (0, \ldots, 0, \beta) \overset{\mathcal{R}^{t-1}}{\longrightarrow} (\beta, 0, \ldots, 0)$$

The designers concluded that conservatively $2t$ rounds are secure when the security level corresponds to the block size n and $(3t - 1)$ rounds are secure for security level of tn bits.

Now, to construct impossible differentials over $(3t - 4)$ rounds, the idea is to reuse the previous $(t - 1)$-round truncated differentials, but further include a few more rounds in the middle. Hence, the impossible differentials can be described as

$$(0, \ldots, 0, \alpha_0) \overset{\mathcal{R}^{3t-4}}{\nrightarrow} (\beta_0, 0, \ldots, 0),$$

where α_0, β_0 are non-zero difference satisfying $\alpha_0 \neq \beta_0$.

The brief explanation of this impossible differentials is as follows. We further extend $(\alpha_0, 0, \ldots, 0)$ by $t/2 - 1$ rounds. The new differences successively produced by the cube mapping are unpredictable, hence we denote them by $\alpha_1, \alpha_2, \ldots, \alpha_{t/2-1}$. Similarly, we further extend $(0, \ldots, 0, \beta_0)$ by $t/2 - 1$ rounds backwards by using the similar notation $\beta_1, \beta_2, \ldots, \beta_{t/2-1}$. Then the difference of each branch in the very middle is described in two ways: by a linear combination of α_i and by a linear combination of β_i. To be a valid

differential propagation, those differences must be equal for all the branches, which yields a system of t linear equations with $2(t/2-1) = t-2$ variables. By solving the system, we obtain that $\alpha_0 = \beta_0$ is a necessary condition to be a valid differential propagation. In other words, for any α_0, β_0 with $\alpha_0 \neq \beta_0$, the propagation is impossible.

The detailed analysis for $t = 12$ is as follows. Here we use the notation $\alpha[i, j, k, \cdots]$ to denote $\alpha_i \oplus \alpha_j \oplus \alpha_k \oplus \cdots$. A notation $\beta[i, j, k]$ is defined in the same way for the backward propagation. Table 4 details the propagation of the differences over 16 rounds, in the forward and in the backward directions.

Then, the propagation of differences in Table 4 is valid only if we have equality between all differences, i.e., if the following equations are satisfied.

$$\begin{aligned}
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_0 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_0 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_3 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_4 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_5 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5,\\
\alpha_1 \oplus \alpha_2 \oplus \alpha_3 \oplus \alpha_4 &= \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4 \oplus \beta_5.
\end{aligned}$$

By summing up all the equations, we get $\alpha_0 = \beta_0$. Hence, the propagation is impossible when $\alpha_0 \neq \beta_0$.

The number of rounds covered by this impossible differential is higher than expected by the designers. However, since it is rather far from the total number of rounds, we do not provide a detailed analysis of the complexity of the best attacks exploiting this differential.

4.4 A Differential Distinguisher

The original paper [AGP+19b, Appendix D] analyzes the resistance of GMiMC against differential attacks. Most notably, the designers exhibit a differential characteristic over $(t+1)$ rounds with two active Sboxes, with probability $2^{-(2n+2)}$ and they conjecture that the corresponding differential is optimal. They deduce that

$$R = 2 + (t+1)\left\lceil \frac{tn}{2(n-1)} \right\rceil \text{ rounds}$$

are sufficient to resist differential cryptanalysis in the sense that the data complexity of the attack exceeds the size of the full codebook. With the parameters we focus on, this corresponds to 93 rounds.

A better differential. We have exhibited another differential, over t rounds, which leads to a much more efficient attack. Let α and α' be two differences in $\mathbb{F}_q$. Then, the difference $(0, \ldots, 0, \alpha, \alpha')$ propagates through t rounds of the permutation as follows:

$$(0, \ldots, 0, \alpha, \alpha') \xrightarrow{\mathcal{R}^{t-2}} (\alpha, \alpha', 0 \ldots, 0) \xrightarrow{\mathcal{R}} (\alpha' + \beta, \beta, \ldots, \beta, \alpha) \xrightarrow{\mathcal{R}} (\beta + \beta', \ldots, \beta + \beta', \alpha + \beta', \alpha' + \beta)\ ,$$

where $\alpha \xrightarrow{S} \beta$ denotes the Sbox transition occurring at Round $(t-1)$ and $\alpha' \xrightarrow{S} \beta'$ the Sbox transition occurring at Round t.

Table 4: New Impossible Differentials of $3t-4$ rounds for $t=12$

	ΔS_0	ΔS_1	ΔS_2	ΔS_3	ΔS_4	ΔS_5	ΔS_6	ΔS_7	ΔS_8	ΔS_9	ΔS_{10}	ΔS_{11}
	0	0	0	0	0	0	0	0	0	0	0	α_0
1	0	0	0	0	0	0	0	0	0	0	α_0	0
2	0	0	0	0	0	0	0	0	0	α_0	0	0
3	0	0	0	0	0	0	0	0	α_0	0	0	0
4	0	0	0	0	0	0	0	α_0	0	0	0	0
5	0	0	0	0	0	0	α_0	0	0	0	0	0
6	0	0	0	0	0	α_0	0	0	0	0	0	0
7	0	0	0	0	α_0	0	0	0	0	0	0	0
8	0	0	0	α_0	0	0	0	0	0	0	0	0
9	0	0	α_0	0	0	0	0	0	0	0	0	0
10	0	α_0	0	0	0	0	0	0	0	0	0	0
11	α_0	0	0	0	0	0	0	0	0	0	0	0
12	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_1	α_0
13	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[12]$	$\alpha[02]$	α_1
14	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[123]$	$\alpha[023]$	$\alpha[13]$	$\alpha[12]$
15	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[1234]$	$\alpha[0234]$	$\alpha[134]$	$\alpha[124]$	$\alpha[123]$
16	$\alpha[12345]$	$\alpha[12345]$	$\alpha[12345]$	$\alpha[12345]$	$\alpha[12345]$	$\alpha[12345]$	$\alpha[12345]$	$\alpha[02345]$	$\alpha[1345]$	$\alpha[1245]$	$\alpha[1235]$	$\alpha[1234]$
	impossible when $\alpha_0 \neq \beta_0$											
16	$\beta[1234]$	$\beta[1235]$	$\beta[1245]$	$\beta[1345]$	$\beta[02345]$	$\beta[12345]$	$\beta[12345]$	$\beta[12345]$	$\beta[12345]$	$\beta[12345]$	$\beta[12345]$	$\beta[12345]$
17	$\beta[123]$	$\beta[124]$	$\beta[134]$	$\beta[0234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$	$\beta[1234]$
18	$\beta[12]$	$\beta[13]$	$\beta[023]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$	$\beta[123]$
19	β_1	$\beta[02]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$	$\beta[12]$
20	β_0	β_1	β_1	β_1	β_1	β_1	β_1	β_1	β_1	β_1	β_1	β_1
21	0	0	0	0	0	0	0	0	0	0	0	β_0
22	0	0	0	0	0	0	0	0	0	0	β_0	0
23	0	0	0	0	0	0	0	0	0	β_0	0	0
24	0	0	0	0	0	0	0	0	β_0	0	0	0
25	0	0	0	0	0	0	0	β_0	0	0	0	0
26	0	0	0	0	0	0	β_0	0	0	0	0	0
27	0	0	0	0	0	β_0	0	0	0	0	0	0
28	0	0	0	0	β_0	0	0	0	0	0	0	0
20	0	0	0	β_0	0	0	0	0	0	0	0	0
30	0	0	β_0	0	0	0	0	0	0	0	0	0
31	0	β_0	0	0	0	0	0	0	0	0	0	0
32	β_0	0	0	0	0	0	0	0	0	0	0	*

It follows that, for any possible value of β, we obtain the following t-round differential as soon as $\beta' = \beta$, which occurs with probability 2^{-n} on average:

$$(0, \ldots, 0, \alpha, \alpha') \xrightarrow{\mathcal{R}^t} (0, \ldots, 0, \alpha + \beta, \alpha' + \beta) \; .$$

Since this probability does not depend on the choice of α and α', this differential can obviously be iterated several times to cover more rounds.

For instance, when $t = 12$ and $n = 61$, the 101 rounds of GMiMC can be decomposed into 8 blocks of $t = 12$ rounds, followed by 5 rounds. We then get a differential of the form

$$(0, \ldots, 0, \alpha, \alpha') \longrightarrow (0, 0, 0, 0, 0, \gamma, \gamma', 0, 0, 0, 0, 0)$$

over the full cipher with probability at least

$$P = (2^{-61})^8 = 2^{-488}$$

since the characteristic over the last 5 rounds has probability 1. This leads to a differential distinguisher over the full permutation with complexity $P^{-1} = 2^{488}$ which is much lower than the size of the full codebook (2^{732}).

It is worth noticing that P is a lower bound on the probability of the 101-round differential since we considered pairs following some specific characteristics by fixing the forms of some differences at intermediate rounds. Obviously, some additional input pairs may lead to an output difference of the same form but not to these specific intermediate differences.

Improving the complexity of the distinguisher with structures. The data complexity of the previous distinguisher can be improved by using structures of inputs. Here, a structure is a set of 2^{2n} inputs of the form $\mathcal{S}_c = \{(c_1, \ldots, c_{t-2}, x, y) \mid x, y \in \mathbb{F}_p\}$. Clearly, the difference between any two elements in the same structure has the form $(0, \ldots, 0, \alpha, \alpha')$. It follows that, from any structure, we can construct 2^{4n-1} pairs of inputs whose difference conforms with the differential. Then, the number of structures required to obtain $P^{-1} = 2^{8n}$ pairs with an appropriate difference is

$$2^{8n-4n+1} = 2^{4n+1},$$

leading to an overall data complexity of $2^{6n+1} = 2^{367}$. The time complexity is equal to the data complexity here, since the distinguisher consists in identifying the output pairs which coincide on all output words except the two in the middle. This obviously does not require computing all pairs of elements in each structure, but only to store the values $\pi(x), x \in \mathcal{S}_c$ according to their first coordinates.

This differential distinguisher does not lead to an attack with complexity below the target security level. However, this must clearly be considered as an unsuitable property since its complexity is much lower than what we expect for a randomly chosen permutation on a set of size 2^{732}.

It is worth noticing that, if we restrict ourselves to distinguishers with complexity below the target security level of 128 bits, then we can use at most $2^{128}/2^{2n} = 2^6$ structures. Therefore, we can derive from these structures $2^{6+4n-1} = 2^{249}$ pairs of inputs conforming with the differential. These pairs be can used to distinguish 4 blocks of t rounds since the differential has probability at least 2^{-244}. Moreover, a valid pair propagates to a differential of the form $(\gamma, \gamma', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ with probability one over $(t-2)$ rounds, and we can extend it by a few more rounds by considering the number of state words that have the same difference. After another 6 rounds, the pair has a differential of the form

$$(\Delta, \Delta, \Delta, \Delta, \Delta, \Delta, *, *, *, *, *, *),$$

with probability one, where $*$ is an unknown difference that we do not care about. This differential form has a constraint of the size $5n$: the left-most six state words have an identical difference. The number of queries to satisfy the same property for a randomly chosen permutation is lower bounded by $2^{5n/2} \approx 2^{152.5}$. This implies that we can distinguish $4t + (t-2) + (t-6) = 64$ rounds of GMiMC from a randomly chosen permutation with complexity less than 2^{128}.

Improved distinguisher using three active words. If we consider a differential with only two active words, the biggest structure we can build is of size 2^{2n}, which limits the advantage of using structures in reducing the cost of the distinguishers. Let us now consider the following differential:

$$\begin{aligned}(0,\dots,0,\alpha,\alpha',\alpha'') &\xrightarrow{\mathcal{R}^{t-3}} (\alpha,\alpha',\alpha'',0\dots,0)\\ &\xrightarrow{\mathcal{R}} (\alpha'+\beta,\alpha''+\beta,\beta,\dots,\beta,\alpha)\\ &\xrightarrow{\mathcal{R}} (\alpha''+\beta+\beta',\beta+\beta',\beta+\beta',\dots,\beta+\beta',\alpha+\beta',\alpha')\\ &\xrightarrow{\mathcal{R}} (\beta+\beta'+\beta'',\dots,\beta+\beta'+\beta'',\alpha+\beta'+\beta'',\alpha'+\beta+\beta'',\alpha''+\beta+\beta')\,,\end{aligned}$$

where $\alpha \xrightarrow{S} \beta$, $\alpha'+\beta \xrightarrow{S} \beta'$ and $\alpha''+\beta+\beta' \xrightarrow{S} \beta''$ denote the Sbox transitions occurring at Round $(t-2)$, at Round $(t-1)$ and at Round t.

As with the previous differential, if $\beta+\beta'+\beta''=0$, which occurs with probability 2^{-n} on average, we have:

$$(0,\dots,0,\alpha,\alpha',\alpha'') \xrightarrow{\mathcal{R}^{t}} (0,\dots,0,\alpha-\beta,\alpha'+\beta+\beta'',\alpha''-\beta'')\,.$$

Again, the probability of this transition is independent of the values of α, α' and α'', so it can be iterated with probability 2^{-n}.

For this differential, we can build structures of size 2^{3n}. This will allow us to consider around 2^{6n} pairs with the required input differential, so we can expect to be able to iterate the characteristic for $6t$ rounds. The total distinguisher will cover $6t + (t-3)$ rounds. As for the previous one, we can add 4 more rounds, generating an output state with 8 words having the same difference with a cost of 2^{3n}, compared to a cost of $2^{7n/2}$ for a random permutation. For GMiMC with $t = 12$, this allows to distinguish 85 rounds with a cost of 2^{3n}. By repeating this procedure 2^n times, we can expect t more rounds to be covered, and distinguish the whole permutation with 101 rounds with a complexity of $2^{5n} = 2^{320}$ and having 9 words with a zero difference (as we do not need to add the final four rounds).

Let us point out that using four instead of three words would not improve the number of rounds attacked on GMiMC-128-d, as the cost of one structure is already the same as the cost of obtaining the 8 non-zero differences in the output for a random permutation. Nevertheless, in the case of the GMiMC variant 256-b with $t = 14$, if we use a similar differential with four active words, we can distinguish up to $8t + (t-4) = 122$ rounds while finding 10 words with no difference and with a complexity of about $2^{4n} = 2^{500}$.

Optimality of this differential. To determine whether further improvements of these differentials are possible, we have searched for other differential characteristics with a Mixed-Integer Linear Programming (MILP) model.

Note that our model only lower-bounds the probability of a fixed differential as done for the previously described characteristics and does not take the details of the initial structure or truncation of the output difference into account.

Previously proposed models for differential characteristics usually represent either each state word [MWGP11] or each bit [AST+17] with a binary decision variable. Neither is well-suited for GMiMC over prime fields: with a word-wise model, we cannot identify

whether two differences are identical and will thus find many invalid characteristics; with a bit-wise model, the model would be unpractically large due to the large state size and number of rounds. We thus model each state word with an integer variable $x \in [-\ell, \ell]$, and addition modulo q simply as $x + y = z$. The variable x in this relation does not define the value of the difference (except for $x = 0$), but only captures properties such as equality and additive relations. The bound ℓ limits the number of distinct difference values that can be modelled and also defines the helper constant $M = 2\ell$.

If x and y are the input and output of an S-box, we only require that $x = 0 \Leftrightarrow y = 0$. One direction $x = 0 \Rightarrow y = 0$ of this implication can be encoded using binary helper variables π_i:

$$1 - \pi_1 M \leq x \leq -1 + \pi_2 M, \qquad 0 - \pi_3 M \leq y \leq 0 + \pi_4 M, \qquad \textstyle\sum_{i=1}^{4} \pi_i \leq 2\,.$$

Each Sbox is associated with a cost $c \in \{0, 1\}$ ($x \neq 0 \Rightarrow c = 1$) as well as a gain $g \in \{0, 1\}$, where $g = 1$ means the output difference $y \neq 0$ is arbitrary and thus the transition does not reduce the success probability. We identify these cases by requiring that $|y|$ is larger than any possible sum of defined differences z, bounded by $2\,z$ (with helper variable g_z), including the permutation's input, output, and S-box outputs in the previous r rounds:

$$-c\,M \leq x \leq c\,M,\ y \geq 2\,z - (1-g_z)M,\ y \geq -2\,z - (1-g_z)M,\ \textstyle\sum_z g_z \geq g\,(2t+r).$$

Finally, we require nontriviality, with helper variables π_x, π'_x for each input x:

$$1 - \pi_x M \leq x \leq -1 + \pi'_x M, \qquad \textstyle\sum_x \pi_x + \pi'_x \leq 2t - 1\,.$$

The minimization objective is the sum of the cost minus the gain of each Sbox. This corresponds to $-\log_q P$, where P is the approximate probability of the differential with fixed input and output difference.

The structured characteristics we describe above yield a cost of $k + 1$ when the number of rounds is between $k\,t - 1$ and $(k + 1)\,t - 2$. We used the MILP model to look at all possible characteristics for up to $3\,t$ rounds. The obtained bounds match our solutions except for $k\,t - 1$ and $k\,t$ rounds, where a small and general modification improves the cost to k instead of $k + 1$. We conclude that the previously described characteristic is essentially optimal with respect to the defined search space.

4.5 Algebraically controlled differential attacks

In this section we show how to use algebraic techniques in order to efficiently find inputs that satisfy a given differential characteristic. The basic idea is to represent the initial state of the permutation symbolically by assigning variables to some of its branches, while the remaining branches are assigned constant values. We then compute the permutation symbolically for several rounds. Namely, for each round, we derive a polynomial expression for each branch of the internal state in terms of the allocated variables.

We repeat this process starting from two initial states (representing two inputs to the permutation), perhaps assigning them different variables. We can now represent the difference between the internal states at each round in these two computations using polynomial expressions in the allocated variables. In particular, each differential transition of the given differential characteristic (whose probability is smaller than one) is expressed as a polynomial equation in the variables. Collecting the equations for all differential transitions, we obtain a system of polynomial equations, whose solution immediately gives two inputs to the permutation that satisfy the differential characteristic. For this approach to be useful, the equation system has to be efficiently solvable, which generally implies that we cannot allocate too many variables and need to minimize the algebraic degree of the polynomial equations.

We start by demonstrating this approach with a basic example.

Satisfying $3t-2$ rounds. We show how to efficiently satisfy $3t-2$ rounds of the iterative differential characteristic of Section 4.4,

$$\begin{aligned}(0,\ldots,0,\mu_0,\mu_0') &\xrightarrow{\mathcal{R}^{t-2}} (\mu_0,\mu_0',0\ldots,0)\\ &\xrightarrow{\mathcal{R}} (\mu_0'+\mu_1,\mu_1,\ldots,\mu_1,\mu_0)\\ &\xrightarrow{\mathcal{R}} (\mu_1+\mu_1',\ldots,\mu_1+\mu_1',\mu_0+\mu_1',\mu_0'+\mu_1),\end{aligned}$$

where we require that $\mu_1+\mu_1'=0$.

Consider an initial state of the permutation of the form

$$X_0=(\alpha_1,\ldots,\alpha_{t-2},x,f(x)),$$

where the α_i are constants in $\mathbb{F}_q$, x is a variable and the function $f(x)$ is described in Section 4.2.3 (see (1)). Then, as described in Section 4.2.3, the internal state at Round $(t-2)$ is described as

$$X_{t-2}=(x+\textstyle\sum_{i=1}^{t-2}\beta_i, f(x)+\sum_{i=1}^{t-2}\beta_i,\gamma_1,\ldots,\gamma_{t-2}),$$

while the state at Round $(2t-2)$ is described as

$$X_{2t-2}=(x'-(x'+\mathsf{RC}_{t-1})^3+\delta_1,-x'+\delta_1,\delta_2,\ldots,\delta_t),$$

where $x'=x+\sum_{i=1}^{t-2}\beta_i$. Starting from Round $(2t-2)$, the algebraic degree of the branches generally grows by a multiplicative factor of 3 per round, namely, the algebraic degree of Round $(2t-2+r)$ is at most 3^{r+1}.

Next, consider another initial state of the permutation of the form

$$Y_0=(\alpha_1,\ldots,\alpha_{t-2},y,f(y)),$$

where the initial constants α_i are identical to those of X_0. Note that the initial difference between the states is of the form

$$\Delta_0=X_0-Y_0=(0,\ldots,0,\mu_0(x,y),\mu_0'(x,y)).$$

Then, the state Y_{2t-2} after Round $(2t-2)$ is described as

$$Y_{2t-2}=(y'-(y'+\mathsf{RC}_{t-1})^3+\delta_1,-y'+\delta_2,\delta_3,\ldots,\delta_t).$$

Therefore, the choice of the initial states of the two inputs, ensures that $(2t-2)$ rounds of the differential characteristic are satisfied with probability one. At Round $2t$, we have

$$\begin{aligned}&\Delta_{2t}=X_{2t}-Y_{2t}=\\ &(\mu_2(x,y)+\mu_2'(x,y),\ldots,\mu_2(x,y)+\mu_2'(x,y),\mu_1(x,y)+\mu_2'(x,y),\mu_1'(x,y)+\mu_2(x,y)),\end{aligned}$$

and we require $\mu_2(x,y)+\mu_2'(x,y)=0$, which is a polynomial equation of degree $3^{2+1}=27$ in the variables x,y. Since we have 2 variables and only one equation in $\mathbb{F}_q$, we can set one of the variables to an arbitrary constant and solve a univariate polynomial equation in the other variable. We expect one solution on average, which gives an input pair that satisfies the differential characteristic for $2t$ rounds. Since the next $(t-2)$ rounds are satisfied with probability one, we can satisfy $3t-2$ rounds at the cost of solving a univariate polynomial equation over $\mathbb{F}_q$ of degree 27.

Satisfying $4t-2$ rounds in an inside-out setting. In an inside-out setting, the differential characteristic can be extended from $(3t-2)$ rounds to $(4t-2)$ rounds algebraically, by adding t rounds before the initial state. Indeed, since the initial state is described by polynomials of degree 3, the state at Round (-2) can be described by polynomials of degree 27:

$$\Delta_{-2} = X_{-2} - Y_{-2} =$$
$$(\mu_{-1}(x,y) + \mu'_{-1}(x,y), \ldots, \mu_{-1}(x,y) + \mu'_{-1}(x,y), \lambda_1(x,y) + \mu'_{-1}(x,y), \lambda'_1(x,y) + \mu_{-1}(x,y)).$$

Therefore, we require $\mu_{-1}(x,y) + \mu'_{-1}(x,y) = 0$, in addition to $\mu_2(x,y) + \mu'_2(x,y) = 0$. This defines a system of two equations of degree 27 in two variables. Any solution with $x \neq y$ defines a pair of states that satisfies a differential characteristic from Round $(-t)$ to Round $(3t-2)$, because Rounds $(-t)$ to (-2) are satisfied with probability 1.

In order to solve the system, we first divide each equation by $(y-x)$ to eliminate trivial solutions with $x = y$. Then we compute a Gröbner basis of the resulting system. Using the MAGMA software, this can be done in less than one minute on a standard PC. Moreover, this can be extended to a distinguisher on 66 rounds by considering a truncated difference in the input and output. We give an example in Figure 7.

```
load('GMiMC_erf.sage') # https://starkware.co/hash-challenge/
S128d_40 = GMiMCParams(field=F61, r=8, c=4, num_rounds=66)

x = vector(F61, [
    2136504846259473744, 1283314153929910666, 1750372136437271205,
    1867169825994287512, 821961362109051955, 1707450857617152361,
    552784820823413051, 484096115705447781, 887825053625051502,
    527122293700370254, 925898050459212322, 1348485354687005037])
y = vector(F61, [
    605957700298844821, 2195497570512456035, 1242887650166759306,
    1453303426557585887, 2164561375454964764, 333859287618218787,
    1549736142184771152, 1358466196860349803, 121930483920884288,
    647266587342612993, 425900737534652142, 848488041762444857])

print ("Input diff : "+" ".join(["{:20}".format(u.lift()) for u in y-x]))
x = erf_feistel_permutation(x, S128d_40)
y = erf_feistel_permutation(y, S128d_40)
print ("Output diff: "+" ".join(["{:20}".format(u.lift()) for u in y-x]))
```

Figure 7: Sagemath code verifying a pair of inputs with a distinguishing property on 66 rounds of GMiMC-128-d: $\Delta_0[10] = \Delta_0[11]$ and $\Delta_{65}[0] = \Delta_{65}[1]$

Satisfying $4t-4$ rounds. If we want to use the differential in a collision attack, we must preserve the value of some initial state words, and we cannot use the inside-out technique. We describe an alternative technique, using a modified differential with four active state

words:

$$
\begin{aligned}
&(0,\dots,0,\mu_0,\mu_0',\mu_0'',\mu_0''')\\
&\xrightarrow{\mathcal{R}^{t-4}} (\mu_0,\mu_0',\mu_0'',\mu_0''',0\dots,0)\\
&\xrightarrow{\mathcal{R}} (\mu_0'+\mu_1,\mu_0''+\mu_1,\mu_0'''+\mu_1,\mu_1,\dots,\mu_1,\mu_0)\\
&\xrightarrow{\mathcal{R}} (\mu_0''+\mu_1+\mu_1',\mu_0'''+\mu_1+\mu_1',\mu_1+\mu_1',\dots,\mu_1+\mu_1',\mu_0+\mu_1',\mu_0'+\mu_1)\\
&\xrightarrow{\mathcal{R}} (\mu_0'''+\mu_1+\mu_1'+\mu_1'',\mu_1+\mu_1'+\mu_1'',\dots,\mu_1+\mu_1'+\mu_1'',\mu_0+\mu_1'+\mu_1'',\mu_0'+\mu_1+\mu_1'',\mu_0''+\mu_1+\mu_1')\\
&\xrightarrow{\mathcal{R}} (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_1,\mu_0+\boldsymbol{\mu}_1-\mu_1,\mu_0'+\boldsymbol{\mu}_1-\mu_1',\mu_0''+\boldsymbol{\mu}_1-\mu_1'',\mu_0'''+\boldsymbol{\mu}_1-\mu_1''')
\end{aligned}
$$

with $\boldsymbol{\mu}_1 = \mu_1+\mu_1'+\mu_1''+\mu_1'''$.

As previously, we require that $\boldsymbol{\mu}_1 = 0$. This happens with probability 2^{-n}, and results in an iterative truncated characteristic $(0,\dots,0,*,*,*,*) \xrightarrow{\mathcal{R}^t} (0,\dots,0,*,*,*,*)$.

As in the previous attack, we build an initial state with special relations in order to control the first t rounds with probability one:

$$X_0 = (\alpha_1,\dots,\alpha_{t-4},x,f(x),y,f(y)).$$

This ensures that the state at Round $(2t-4)$ is of the form:

$$X_{2t-4} = (x'-(x'+\mathsf{RC}_{t-1})^3+\delta_1, -x'+\delta_2, y'-(y'+\mathsf{RC}_{t-1})^3+\delta_3, -y'+\delta_4, \delta_5,\dots,\delta_t).$$

Instead of considering two different states with this shape (with four unknown in total), we will consider one variable state, and one fixed state with $(x,y)=(0,0)$. When we consider the state at Round $(2t)$, we have

$$
\begin{aligned}
&\Delta_{2t} = X_{2t} - X_{2t}(0,0) =\\
&(\boldsymbol{\mu}_2,\dots,\boldsymbol{\mu}_2,\mu_1+\boldsymbol{\mu}_2-\mu_2,\mu_1'+\boldsymbol{\mu}_2-\mu_2',\mu_1''+\boldsymbol{\mu}_2-\mu_2'',\mu_1'''+\boldsymbol{\mu}_2-\mu_2''')
\end{aligned}
$$

where $(\mu_1,\mu_1',\mu_1'',\mu_1''')$ are polynomials of degree 3, 1, 3, and 1 respectively (as seen in X_{2t-4}), and $(\mu_2,\mu_2',\mu_2'',\mu_2''')$ are polynomials of degree 9, 27, 81, and 243, with $\boldsymbol{\mu}_2 = \mu_2+\mu_2'+\mu_2''+\mu_2'''$. All polynomials have variables x and x', and $X_{2t}(0,0)$ is a vector of constants. We now require $\boldsymbol{\mu}_2(x,x')=0$, and we can simplify the state using this assumption:

$$X_{2t} = X_{2t}(0,0) + (0,\dots,0,\mu_1-\mu_2,\mu_1'-\mu_2',\mu_1''-\mu_2'',\mu_1'''+\mu_2+\mu_2'+\mu_2'').$$

We obtain an expression of degree $(0,\dots,0,9,27,81,81)$.

When we focus on Round $(3t)$, we can now express the condition of the differential as a polynomial of degree 729. Therefore, we have a system of two equations of degree 243 and 729 in two variables. This can be solved efficiently by computing the resultant of the two bivariate polynomials, and finding its roots (the resultant is a univariate polynomial of degree 19683). Finding the roots is the bottleneck, with an asymptotic complexity of roughly $d^{1.5}$ field operations for a polynomial of degree d [KU08].

Any solution with $(x,y)\neq(0,0)$ defines a state such that the pair $(X(x,y),X(0,0))$ satisfies the differential characteristic up to Round $(4t-4)$, because Rounds $(4t)$ to $(4t-4)$ are satisfied with probability one.

Extending the differentials. All these attacks can be extended probabilistically by finding about q different input pairs that satisfy the differential characteristic (each pair is found by choosing different constants α_i in the initial state). With high probability, one of these input pairs will also satisfy the next differential transitions, and follow the characteristic for t more rounds.

4.6 Reduced-round collision attacks

We can build collisions on a reduced number of rounds by using the same ideas as for the previous structural or algebraic differential distinguishers. The additional constraint that we have now compared to distinguishers is that any value that needs to be chosen must be assigned to the rate part, *i.e.* the 8 left-most words in GMiMC-128-d, and the capacity part, *i.e.* the 4 right-most words in GMiMC-128-d, will be fixed to a known value we cannot choose.

Building collisions with structures. We won't use the 3-word differential but the 2-word one, as already using the full $2n$ structure from the 2-word one would imply a complexity equivalent to that of a generic collision attack. Instead of having $t = 12$ free rounds at the beginning, we will have only 8, due to the 4 words reserved for the capacity. With a cost of $2^{r \cdot n}$ we can then go through $r \cdot t$ rounds maintaining the same differential. Finally, we can freely add $(t-2)$ rounds that will keep the differences at the rate part, and, consequently, can be finally cancelled:

$$\begin{aligned}(0, \ldots, 0, \alpha, \alpha', 0, 0, 0, 0) &\xrightarrow{\mathcal{R}^{t-6}} (\alpha, \alpha', 0 \ldots, 0) \\ &\xrightarrow{\mathcal{R}^{r \cdot t}} (\beta, \beta', 0 \ldots, 0)\ .\end{aligned}$$

This differential has a probability of $2^{-r \cdot t}$, and would allow to build collisions up to $3t - 6$ rounds, so for 30 rounds for GMiMC. If we use structures we can improve this: if we build a structure of size 2^x, with the cost of the structure we can verify a probability up to 2^{-2x}. If we choose structures of size $2^{3n/2}$, we can consider $r = 3$. This would provide collisions for $4t - 6$ rounds. For GMiMC-128-d this implies collisions on 42 rounds with a cost of 2^{92}, and for GMiMC-256 it implies collisions on 50 rounds with a complexity of 2^{187}.

Building collisions with algebraically controlled techniques. In order to use the algebraically controller techniques in a collision attack, we must not use any difference in the inner part of the sponge. As we said, in the case of GMiMC-128-d, we have $c = 4$, therefore, we start from a state

$$X_0 = (\alpha_1, \ldots, \alpha_4, x, f(x), y, f(y), \alpha_9, \ldots, \alpha_{12})$$

and we have a characteristic over $4t - 4 - c = 40$ rounds. The system can be solved algebraically, by computing the resultant of the two bivariate polynomials, and finding its roots. In MAGMA, this takes a few minutes using less than 3GB of RAM. We give an example of conforming pair in Figure 8, where all the α constants have been set to zero.

This attack can be extended to t more rounds probabilistically, with a complexity of $q \cdot 19683^{1.5} \approx 2^{83}$.

4.7 Summary

Table 5 summarizes all results we have obtained on GMiMC with the considered parameters. It includes a practical collision attack covering 40 (out of 101) rounds of GMiMC. For the full version (i.e. with the 101 rounds), all exhibited weaknesses are *distinguishers for the inner permutation*. The existence of such distinguishers does not imply the existence of a concrete collision or preimage attack on the corresponding sponge. However, since some of these distinguishers have a complexity which is much lower than the claimed security level, they clearly invalidate the security arguments deduced from the indifferentiability proof of the sponge construction. Moreover, the existence of several distinguishers, of different types, which are more powerful than expected by the designers, gives very little confidence in the security of GMiMC with these parameters. It is worth noticing that a better choice of the parameters may increase the security.

```
load('GMiMC_erf.sage') # https://starkware.co/hash-challenge/
S128d_40 = GMiMCParams(field=F61, r=8, c=4, num_rounds=40)

x = vector(F61, [0, 0, 0, 0,                    0, 1265014881285225376,
                      0, 1323963633845726391, 0, 0, 0, 0])
y = vector(F61, [0, 0, 0, 0, 1687869230625042828, 1678073603247747657,
    1246244071391540901, 1919915214622971772, 0, 0, 0, 0])

print ("Input diff : "+" ".join(["{:20}".format(u.lift()) for u in y-x]))
x = erf_feistel_permutation(x, S128d_40)
y = erf_feistel_permutation(y, S128d_40)
print ("Output diff: "+" ".join(["{:20}".format(u.lift()) for u in y-x]))
```

Figure 8: Sagemath code verifying a pair of inputs following the characteristic for a 40-round collision attack of GMiMC-128-d.

Table 5: Distinguishers on the GMiMC permutation and attacks on the corresponding sponge hash functions. The variant aiming at 128-bit security operates on $t = 12$ elements in $\mathbb{F}_q$ with $q = 2^{61} + 20 \times 2^{32} + 1$. The variant aiming at 256-bit security operates on $t = 14$ elements in $\mathbb{F}_q$ with $q = 2^{125} + 266 \times 2^{64} + 1$.

Primitive (security)	Rounds		Attack			
			Type	Rounds	Cost	Sect.
GMiMC (128 bits)	101	permutation	integral distinguisher	70	2^{61}	4.2.3
			ZS distinguisher	102	2^{48}	4.2.5
			ZS distinguisher	128	2^{122}	4.2.4
			diff. distinguisher	64	2^{123}	4.4
			diff. distinguisher	66	practical	4.5
		hash function	collisions	40	practical	4.6
			collisions	42	2^{92}	4.6
			collisions	52	2^{83}	4.6
GMiMC (256 bits)	186	permutation	integral distinguisher	116	2^{125}	4.2.3
			ZS distinguisher	186	2^{108}	4.2.5
			ZS distinguisher	206	2^{125}	4.2.4
			ZS distinguisher	218	2^{250}	4.2.4
		hash function	collisions	50	2^{187}	4.6

5 HadesMiMC (Starkad and Poseidon)

5.1 Description

HadesMiMC is a family of permutations described by Grassi *et al.* in [GLR+19] which follows a new construction for block ciphers called HADES. The aim of the HADES construction is to decrease the number of Sboxes used in a Substitution-Permutation Network with a guarantee that the cipher resists all known attacks, including differential and linear cryptanalysis and algebraic attacks. Reducing the number of Sboxes is especially important in many applications, for instance when the cipher needs to be protected against side-channel attacks [GGNS13] or if it is combined with an FHE-scheme [ARS+15]. This was traditionally achieved by using a partial substitution-layer, i.e., an Sbox layer which

does not operate on the whole internal state. However, several attacks on this type of constructions, e.g. [BDD+15, DLMW15, DEM16, RST18] show that it is much more difficult to estimate the security level of these constructions than that of classical SPNs. The basic principle of the HADES construction is then to combine both aspects: the inner rounds in the cipher have a partial Sbox layer in order to increase the resistance to algebraic attacks at a reduced implementation cost, while the outer rounds consist of traditional SPN rounds, with a full Sbox layer. The resistance against statistical attacks is then analyzed by removing the inner rounds. Indeed, the cipher without the inner rounds is a classical SPN and its resistance to differential and linear cryptanalysis can be estimated with well-studied methods. On the other hand, the resistance to algebraic attacks, e.g. the evolution of the algebraic degree over the cipher, involves the inner rounds.

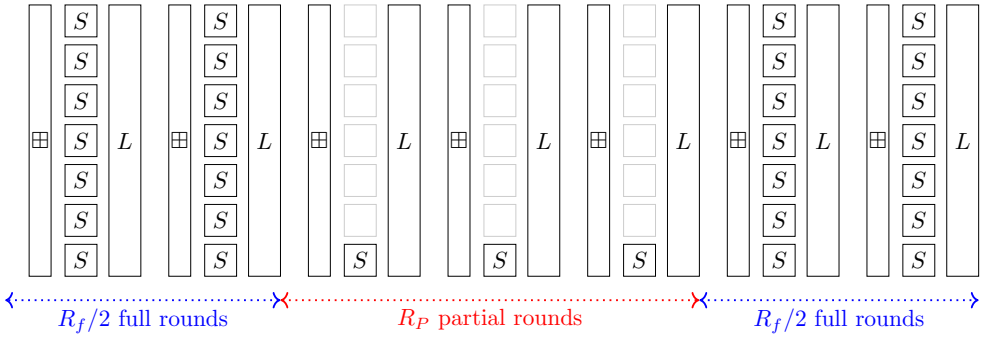


Figure 9: The HADES construction with $t = 6$.

HadesMiMC [GLR+19, Section 3] is then a keyed permutation following the HADES construction dedicated to MPC applications or to STARK-proof systems, where the Sbox is defined by the cube mapping over a finite field and the linear layer L corresponds to a $(t \times t)$-MDS matrix. Two concrete instantiations of HadesMiMC are then detailed by Grassi *et al.* in [GKK+19], namely:

- Starkad operates on t elements in a binary field with odd dimension (which guarantees that the cube mapping is bijective);
- Poseidon operates on t elements in a prime field $\mathbb{F}_p$ with $p \bmod 3 \neq 1$.

For the parameters we focus on, i.e., $t = 12$ and $q \simeq 2^{64}$, the number of rounds is equal to:

- for Starkad, $R_f = 8$ full rounds, and $R_p = 43$ partial rounds, with $q = 2^{63}$.
- for Poseidon, $R_f = 8$ full rounds, and $R_p = 40$ partial rounds, with $q = p = 2^{61} + 20 \times 2^{32} + 1$.

It is worth noticing that, since the cube mapping has differential uniformity 2 over $\mathbb{F}_{2^n}$ (n odd) and over $\mathbb{F}_p$, $R_f = 6$ rounds are enough to guarantee that there is no *differential characteristic* over the full rounds with probability higher than 2^{-tn}.

The following sections describe two types of attacks against HadesMiMC, which both exploit the propagation of affine subspaces over the partial rounds. The first one is an integral distinguisher covering all rounds except the first two rounds for most sets of parameters. The second one is a preimage attack on the full function which applies when the MDS matrix defining the linear layer has, up to multiplication by a scalar, a low multiplicative order. It is worth noticing that, while the designers of HadesMiMC do not mention any requirements on this MDS matrix, they provide several suggestions.

Algorithm 2 HadesMiMC with block-size t, $2R_f$ full rounds and R_P partial rounds.

Input: $(x_1, \ldots, x_t) \in \mathbb{F}^t$
  for r from 0 to $(R_f/2 - 1)$ **do**
    for i from 1 to t **do**
      $y_i \leftarrow (x_i + \mathsf{RC}_{r,i})^3$
    end for
    $(x_1, \ldots, x_t) \leftarrow L(y_1, \ldots, y_t)$
  end for
  for r from 0 to $(R_P - 1)$ **do**
    $y_t \leftarrow (x_t + \mathsf{RC}'_{r,t})^3$
    $(x_1, \ldots, x_t) \leftarrow L(x_1 \ldots, x_{t-1}, y_t)$
  end for
  for r from 0 to $(R_f/2 - 1)$ **do**
    for i from 1 to t **do**
      $y_i \leftarrow (x_i + \mathsf{RC}''_{r,i})^3$
    end for
    $(x_1, \ldots, x_t) \leftarrow L(y_1, \ldots, y_t)$
  end for
Output: $(x_1, \ldots, x_t)$

For Starkad and Poseidon, Cauchy matrices are used [GKK+19]. In Appendix A, we identify weak instances from this class of matrices. Alternatively, the HadesMiMC authors propose [GLR+19, Appendix B] the use of a matrix of the form $A \times B^{-1}$ where both A and B are Vandermonde matrices with generating elements a_i and b_i. In this case, if $a_i = b_i + r$ for some $r \in \mathbb{F}_q$, then the resulting MDS matrix will be an involution for $\mathbb{F}_q$ of characteristic two [SDMO12]. Similarly, in characteristic $p \neq 2$, one obtains an involution whenever $a_i = -b_i$.

5.2 Integral distinguishers

The main threats for all variants of MiMC seem to be attacks of an algebraic nature exploiting the fact that the only nonlinear operation is the cube mapping over a finite field, which has a low degree compared to the size of the field. This is why most of these permutations require a large number of rounds compared to usual ciphers like the AES.

Multivariate polynomial over $\mathbb{F}_q$. In HadesMiMC, the number of rounds has been chosen by the designers in such a way that, when each coordinate of the output is expressed as a polynomial in t variables in

$$\mathbb{F}_q[x_1, \ldots, x_t] / \left(x_1^q - x_1, \ldots, x_t^q - x_t\right),$$

i.e., as

$$\sum_{(i_1, \ldots i_t)} c_{i_1, \ldots i_t} x_1^{i_1} x_2^{i_2} \ldots x_t^{i_t} \text{ with } c_{i_1, \ldots i_t} \in \mathbb{F}_q$$

then the degree of this polynomial *in each input* is close to $(q-1)$, which is the behaviour expected for a randomly chosen permutation. Assuming that the degree in each variable grows as 3^r for r rounds (which is an upper bound), then 39 rounds (resp. 40 rounds) are enough for Poseidon (resp. for Starkad) to achieve maximal degree in each variable. Then, as analyzed by the designers in [GLR+19, Lemma 1], once we get a polynomial of degree $(q-1)$ in each input, $\lceil \log_3(t) \rceil$ rounds are enough to get a polynomial of total degree $(q-1)t$.

It is worth noticing that, among the parameters proposed in the StarkWare challenges, the variants operating over a field $\mathbb{F}_q$ with $\log_2 q \simeq 256$ (namely variants 128-b and 128-e) do not satisfy this requirement since they have at most 96 rounds implying that the degree of the coordinates in each input is at most 2^{152} which is much smaller than the field size. Exploiting this property in an attack with data complexity less than the claimed security level is still an open problem. However, the fact that the degree of the permutation is much lower than expected for an ideal permutation does not seem to be a suitable property. Also, this analysis assumes that the degree over the partial rounds increases exactly as over the full rounds, which seems to be a strong hypothesis.

An integral property. Our idea to improve upon the trivial bound above by a few partial rounds is to choose a specific subspace of inputs. Indeed, we are going to construct a one-dimensional subspace V such that $t-1$ partial rounds will map any coset $V+v_0$ onto a coset of another one-dimensional subspace W. Adding at most $\lfloor \log_3(q-2) \rfloor$ rounds (either full or partial), ensures that the conditions of Corollary 1 are satisfied and thus the outputs sum to zero.

$$V + v_0 \xrightarrow{\mathcal{R}_p^{t-1}} W + w_0 \xrightarrow{\deg < q-1} \text{zero sum.}$$

Let us denote by V a linear subspace of internal states after the Sbox layer of the last of the first $R_f/2$ full rounds (see Figure 10). Then, this subspace leads to an affine subspace at the input of the first partial round, which is a coset of $L(V)$. The following lemma guarantees the existence of a nontrivial vector space $L(V)$ such that any coset of $L(V)$ is mapped to a coset of $W = L^t(V)$ after $t-1$ partial rounds.

Lemma 1. *Let $F : \mathbb{F}_q^t \to \mathbb{F}_q^t$ denote a permutation obtained from $r \geq 1$ partial* HadesMiMC *rounds instantiated with linear layer L. If L has multiplicative order h, then there exists a vector space V with $\dim V \geq t - \min\{h, r\}$ such that $F(x+V) \subseteq F(x) + L(V)$ for all $x \in \mathbb{F}_q^t$.*

Proof. Let $V = \langle \delta_t, L^T(\delta_t), \ldots, (L^T)^{r-1}(\delta_t) \rangle^\perp$ where $\delta_t = (0, \ldots, 0, 1)$. Clearly, $\dim V$ satisfies the desired lower bound. It suffices to show that for all $x \in \mathbb{F}_q^t$ and $v \in V$, $F(x+v) = F(x) + v$. Let $F = R_r \circ \cdots \circ R_1$. Since the last coordinate of any v in V is zero, i.e. $v \perp \delta_t$, the image of $x + V$ by the partial Sbox layer is a coset of V. It follows that $R_1(x+v) = R_1(x) + L(v)$. Similarly, for Round $i = 2, \ldots, r$, it holds that $R_i(x_i + L^{i-1}(v)) = R_i(x_i) + L^i(v)$ if $L^{i-1}(v) \perp \delta_t$ or equivalently $v \perp (L^\top)^{i-1}(\delta_t)$. □

Let us consider any coordinate y of the output of the permutation after adding r additional (partial or full) rounds. When z_0 varies in V, these output words correspond to the images by the additional rounds of the elements z_1 in a coset of $W = L^t(V)$, which we denote by $\gamma + W$ (see Figure 10). As the polynomial corresponding to the r additional rounds has degree at most 3^r, it then follows using Corollary 1 that

$$\sum_{z_0 \in V} y(z_0) = \sum_{z_1 \in \gamma + W} y(z_1) = \sum_{x \in \mathbb{F}_q} P(x) = 0 \,,$$

as long as r is at most $\lfloor \log_3(q-2) \rfloor$.

Thus, in total this covers $(t-1) + \lfloor \log_3(q-2) \rfloor$ rounds, starting after the first full rounds. For most sets of concrete parameters, this actually exceeds the recommended number of rounds in the forward direction for both Poseidon and Starkad. Furthermore, Lemma 1 implies that if the linear layer L has multiplicative order less than $t-1$, then the distinguisher covers an arbitrary number of partial rounds.

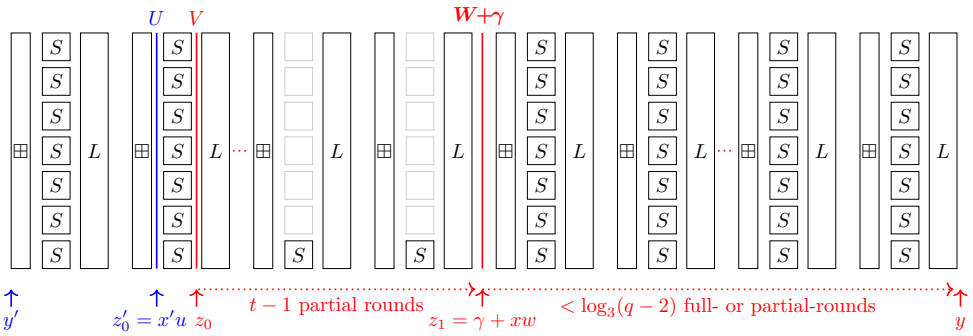


Figure 10: Zero-sum distinguisher against POSEIDON and STARKAD covering $(2+4)$ full rounds and all partial rounds.

Zero-sum distinguishers over $\mathbb{F}_q$. By extending the above-mentioned approach in the backwards direction, we can construct a zero-sum distinguisher with a (slightly) extended number of rounds as depicted on Figure 10. The problem is that, contrary to the case of GMiMC, the inverse round function in HADESMIMC is very different from the round function itself, and it has a much higher degree. Indeed, the inverse of the cube mapping over $\mathbb{F}_q$ is a power function $S^{-1} : x \mapsto x^s$ whose exponent is given by

$$s = \frac{2q-1}{3} .$$

By using classical bounds on the degree, we cannot guarantee a degree lower than $(q-2)$ for more than one single round backwards.

However, V being one-dimensional allows to overcome one additional layer of Sboxes, and thus one additional round. Namely, as V is a one-dimensional space there exists a vector $v = (v_1, \ldots, v_t) \in \mathbb{F}_q^t$ such that

$$V = \{(x\,v_1, x\,v_2, \ldots, x\,v_t) \mid x \in \mathbb{F}_q\}.$$

The image of V under the inverse of the full Sbox layer consists of all the vectors in $\mathbb{F}_q^t$ of the form

$$\left((x\,v_1)^{1/3}, \ldots, (x\,v_t)^{1/3}\right) = x^{1/3}\left(v_1^{1/3}, \ldots, v_t^{1/3}\right).$$

As a consequence, this image is again a one-dimensional vector space having the same form, namely $U = \{x'\,(u_1, \ldots, u_t) \mid x' \in \mathbb{F}_q\}$ where $u_i = v_i^{1/3}$ for all $0 \le i < t$. It is worth noticing that this particular structure does not propagate over more rounds because of the addition of a round constant. Then, any coordinate at the input of the previous round y' is the image of an element $z'_0 = x'u$ in U by an affine layer, followed by the inverse of Sbox, i.e., by $x \mapsto x^{1/3}$ (see Figure 10). We can then consider this mapping as a function of $x' \in \mathbb{F}_q$, and express it as a polynomial Q with coefficients in $\mathbb{F}_q$. Since the degree of this polynomial is the degree of the inverse Sbox, it does not exceed $(q-2)$. Using the notation from Figure 10, we then have

$$\sum_{z_0 \in V} y'(z_0) = \sum_{z'_0 \in U} y'(z'_0) = \sum_{x' \in \mathbb{F}_q} Q(x) = 0 .$$

For most sets of proposed parameters, this provides a zero-sum distinguisher with data complexity q on HADESMIMC for all but the two initial rounds, i.e. for $2+4$ full rounds (2 at the beginning and 4 at the end), and all partial rounds, as detailed in Table 6. Again, for instantiations of HADESMIMC with a linear layer of multiplicative order less than $t-1$, the distinguisher covers an arbitrary number of partial rounds.

Table 6: Number of rounds of HadesMiMC covered by the zero-sum distinguisher of complexity q.

		Poseidon			Starkad		
security level	t	$\log_2 q$	proposed R_f, R_P	nb of rounds of the ZS	$\log_2 q$	proposed R_f, R_P	nb of rounds of the ZS
128 bits	12	61	8, 40	2+4, 45	63	8, 43	2+4, 46
	4	125	8, 81	2+4, 77	125	8, 85	2+4, 77
	12	125	8, 83	2+4, 85	125	8, 86	2+4, 85
	3	253	8, 83	2+4, 157	255	8, 85	2+4, 158
	12	253	8, 85	2+4, 165	255	8, 88	2+4, 166
256 bits	8	125	8, 82	2+4, 81	125	8, 86	2+4, 81
	14	125	8, 83	2+4, 87	125	8, 83	2+4, 87

Multivariate polynomial over $\mathbb{F}_2$ for Starkad. In the case of Starkad, the permutation can also be seen as a permutation over another algebraic structure, namely $\mathbb{K}^{mt}$ where $\mathbb{K}$ is a subfield of $\mathbb{F}_{2^n}$. The simplest setting obviously corresponds to $\mathbb{K} = \mathbb{F}_2$. Then, each bit of the output of the permutation can be expressed as a Boolean function of nt variables, whose *algebraic normal form* is a polynomial in

$$\mathbb{F}_2[x_1, \ldots, x_{nt}] / \left(x_1^2 - x_1, \ldots, x_{nt}^2 - x_{nt}\right).$$

For instance, in the same setting as before for finding a zero-sum distinguisher, we consider internal states in an $\mathbb{F}_{2^n}$-affine space V, which can also be seen as an affine space in $\mathbb{F}_2^{nt}$. Then, any mapping from V into $\mathbb{F}_{2^n}$ can be expressed both as a univariate polynomial P in $\mathbb{F}_{2^n}[x]$ and as a collection of n multivariate polynomials $p_1, \ldots, p_n$ in $\mathbb{F}_2[x_1, \ldots, x_n] / \left(x_1^2 - x_1, \ldots, x_n^2 - x_n\right)$ (i.e., algebraic normal forms of n Boolean functions). There is a well-known relation between the degree of the univariate polynomial P and the degree of the multivariate polynomials p_i (see e.g. [Can16, Prop. 2.5]): for each i,

$$\deg p_i = \max\{wt(u) : 0 \leq u < 2^n \text{ and } c_u \neq 0\}$$

where $P(x) = \sum_{u=0}^{2^n-1} c_u x^u$ and $wt(u)$ denotes the number of ones in the binary decomposition of the integer u. It then follows that the cube mapping has binary degree 2 (since $3 = 2^1 + 2^0$) and its inverse has degree $(n+1)/2$ since

$$\frac{2^{n+1} - 1}{3} = \sum_{i=0}^{(n-1)/2} 2^{2i}$$

see e.g. [Nyb94, Prop. 5]. It implies that the binary degree of r rounds in the forward direction cannot exceed the number of binary digits of an exponent u which belongs to $\{0, \ldots, 3^r\}$, i.e.,

$$\lfloor \log_2(3^r) \rfloor = \lfloor r \log_2(3) \rfloor .$$

We then deduce that 39 rounds in the forward direction have degree at most $(n-1)$ which guarantees that, when the inputs vary in a subspace of dimension n or more, the corresponding output bits sum to zero.

But we are able to extend the previous zero-sum distinguisher by one round backwards by exploiting the binary degree. Indeed, the full rounds correspond to a classical SPN with an Sbox σ of binary degree $d = \frac{n+1}{2}$, and it is well-known that the binary degree of an SPN does not grow with the number of rounds r as d^r, but a bit slower. This property has been used on several hash functions (e.g. [BCD11]). We can then apply the best known

bound on the degree of an SPN, which is Theorem 3.2 in [BC13]. In our case, this theorem states that the degree of two rounds backwards is upper-bounded by

$$nt - \frac{nt - d}{\gamma(\sigma)} \tag{4}$$

where $\gamma(\sigma)$ is a parameter which depends on the Sbox only, namely

$$\gamma(\sigma) = \max_{1 \le i \le n-1} \frac{n-i}{n - \delta_i(\sigma)}$$

and $\delta_i(\sigma)$ denotes the maximal binary degree of the product of any i (or fewer) Boolean coordinates of σ. It appears that, in the specific case of S^{-1}, the inverse of the cube mapping, the value of $\gamma(S^{-1})$ is much smaller that the degree of S^{-1}. This comes from the fact that the binary degree of S is only 2. Therefore, by applying the relation between $\delta_k(S^{-1})$ and $\delta_\ell(S)$ given in [BC13, Theorem 3.1], we obtain that

$$\delta_{n-2^\ell-1}(S^{-1}) < n - \ell$$

for any $1 \le \ell \le 5$. Based on these bounds, we deduce that, for $t = 12$ and $\log_2 q = 63$,

$$\gamma(S^{-1}) \le \frac{n-2}{6}$$

as detailed in Table 7.

Table 7: Computing the value of $\gamma(S^{-1})$ in [BC13, Theorem 3.1] for $S^{-1} : x \mapsto x^{1/3}$ over $\mathbb{F}_{2^n}$ with $n = 63$.

i	1	2	...	31	...	47	...	55
bound on $\delta_i(S^{-1})$	$\frac{n+1}{2}$	$n-6$		$n-5$		$n-4$		$n-3$
bound on $\frac{n-i}{n-\delta_i(S^{-1})}$	2	$\frac{n-2}{6}$		$\frac{n-31}{5}$		$\frac{n-47}{4}$		$\frac{n-55}{3}$

i	...	$n-4$	$n-3$	$n-2$	$n-1$
bound on $\delta_i(S^{-1})$		$n-2$	$n-2$	$n-1$	$n-1$
bound on $\frac{n-i}{n-\delta_i(S^{-1})}$		2	$\frac{3}{2}$	2	1

By applying (4) with $\gamma(\sigma) = \frac{n-2}{6}$, we deduce that two full rounds backwards in Starkad have degree at most 684. It is worth noticing that the same bound applied to 3 rounds backwards leads to an upper-bound equal to 748, which is very close to the block-size.

However, since the degree of two rounds backwards is strictly less than $n(t-1)$, we can exhibit a zero-sum distinguisher on Starkad covering $(3+4)$ full rounds and 35 partial rounds, as depicted on Figure 11. We here choose the internal states after the Sbox layer of the third round of the permutation in the linear space of $\mathbb{F}_{2^n}^t$ of dimension $(t-1)$

$$V = \{(x_1, \ldots, x_{t-1}, 0), x_i \in \mathbb{F}_{2^n}\} .$$

Then, the forward direction corresponds to the linear layer of the second round, followed by 39 rounds, which have binary degree at most $(n-1)$ in the bits of each x_i. This implies that the output bits of the permutation sum to zero when all inner states in V are considered.

In the backward direction, the structure of V is obviously preserved by the inverse of the Sbox layer. Thus, the output of the second round of the permutation is an affine

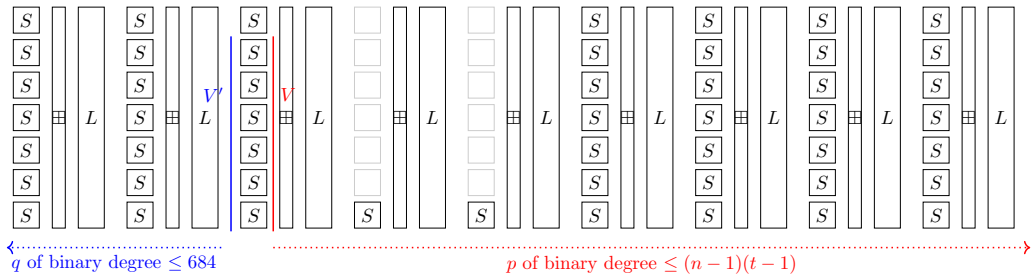


Figure 11: Zero-sum distinguisher against STARKAD with $(3+4)$ full rounds and 35 partial rounds based on the binary degree.

subspace $V' \subset \mathbb{F}_2^{nt}$ of dimension $n(t-1) = 693$. The input bits are then the images of the elements of this subspace by two rounds backwards, which corresponds to a Boolean function of degree at most 684. The sum of any input bit when the intermediate states vary in V then corresponds to the value of a derivative of order 693 of a function of degree at most 684. This sum is then always equal to zero. However, extending the integral property by one round backwards with this technique has a huge cost since the complexity of this distinguisher is now $2^{n(t-1)} = 2^{693}$ which is much higher than the target security level.

Univariate degree. An important observation is that we analyzed here the algebraic properties of the coordinates of the permutation seen as multivariate polynomials only. But the whole permutation can also be expressed as a univariate polynomial with coefficients in $\mathbb{F}_{q^t}$ where the vector space $\mathbb{F}_q^t$ is identified with the field $\mathbb{F}_{q^t}$. This is, for instance, the situation of the original MIMC cipher whose Sbox alphabet corresponds to the whole block-size. But determining the degree of this univariate polynomial in the case of HADESMIMC remains an open question.

5.3 Finding preimages by linearization of the partial rounds

This section shows that, when the linear layer in HADESMIMC has a low multiplicative order, the propagation of linear subspaces through all partial rounds leads to a much more powerful attack. Indeed, we now show that the existence of perfect linear approximations over the partial rounds of HADESMIMC, as detailed in Lemma 2, can be used to setup a simplified system of equations for finding preimages, leading to a full-round preimage attack.

Lemma 2. *Let $F : \mathbb{F}_q^t \to \mathbb{F}_q^t$ denote a permutation obtained from $r \geq 1$ partial* HADESMIMC *rounds instantiated with linear layer L and round constants $c_1, \ldots, c_r$. Let $V \subset \mathbb{F}_q^t$ be the vector space $V = \langle L(\delta_t), L^2(\delta_t), \ldots, L^r(\delta_t)\rangle^\perp$, where $\delta_t = (0, \ldots, 0, 1)$. Then, for all $x \in \mathbb{F}_q^t$ and $v \in V$,*

$$v \cdot F(x) = v \cdot L^r(x) + \sum_{i=1}^{r} v \cdot L^{r+1-i}(c_i),$$

where $u \cdot v$ denotes the usual scalar product in $\mathbb{F}_q^t$. Furthermore, if L has multiplicative order h, then $\dim V \geq t - \min\{h, r\}$.

Proof. Let $F_r = R_r \circ R_{r-1} \circ \cdots \circ R_1$, where R_i denotes the ith partial HADESMIMC round, namely $R_i(x) = L \circ S(x + c_i)$. We proceed by induction on r. For $r = 1$, we have, for any v and x,

$$v \cdot R_1(x) = L^T(v) \cdot S(x + c_1) = L^T(v) \cdot (x + c_1) = v \cdot L(x) + v \cdot L(c_1)$$

if the last coordinate of $L^T(v)$ is zero, or equivalently $L^T(v) \cdot \delta_t = v \cdot L(\delta_t) = 0$.

Let us now consider Round r and $v \in \langle L(\delta_t), L^2(\delta_t), \ldots, L^r(\delta_t)\rangle^\perp$. For any $y \in \mathbb{F}_q^t$, we have

$$v \cdot R_r(y) = L^T(v) \cdot S(y + c_r) = L^T(v) \cdot (y + c_r)$$

since $L^T(v) \cdot \delta_t = v \cdot L(\delta_t) = 0$. Letting $y = F_{r-1}(x)$, it follows that

$$v \cdot F_r(x) = L^T(v) \cdot F_{r-1}(x) + L^T(v) \cdot c_r = L^T(v) \cdot L^{r-1}(x) + \sum_{i=1}^{r-1} L^T(v) \cdot L^{r-i}(c_i) + L^t(v) \cdot c_r$$

where the last equality is deduced from the induction hypothesis using that $L^T(v)$ belongs to $\langle L(\delta_t), \ldots, L^{r-1}(\delta_t)\rangle^\perp$. Finally, it is easy to see that the dimension of $V^\perp$ can be upper bounded as $\dim V^\perp \leq \min\{h, r, t\}$. Hence, $\dim V \geq t - \min\{h, r\}$. □

Suppose that L is such that the vector space V from Lemma 2 is of dimension d. It will be shown that, if d is sufficiently large, such an instantiation of HadesMiMC is vulnerable to preimage attacks for some choices of the rate and capacity parameters of the sponge construction. In particular, when the MDS matrix L in an involution, we obtain $d = t - 2$.

By Lemma 2, there exists a matrix $U_1 \in \mathbb{F}_q^{d \times t}$ such that $U_1 F(x) = U_1(L^r(x) + a)$ for a known constant a. Indeed, let the rows of U_1 be a basis for V. Furthermore, let $U_2 \in \mathbb{F}_q^{(t-d)\times t}$ be a matrix with row space complementary to the row space of U_1. For each x, it holds that

$$\begin{aligned} U_1 y &= U_1(L^r(x) + \textstyle\sum_{i=1}^{r} L^{r+1-i}(c_i)) \\ U_2 y &= U_2 F(x). \end{aligned} \tag{5}$$

Consider a HadesMiMC permutation in a sponge construction with rate k and capacity $c = t - k$. Computing preimages of a one-block message $(y_1, \ldots, y_k) \in \mathbb{F}_q^k$ then corresponds to solving the system of equations $[F(x \| \mathsf{IV})]_i = y_i$, $i = 1, \ldots, k$ in the unknowns $x_1, \ldots, x_k$.

The idea of the attack is simple: for each guess of $U_2 F(x) \in \mathbb{F}_q^{t-d}$, replace the equations for the partial rounds by the affine relations (5) and solve the resulting system of equations. In order to ensure that the ideal generated by these equations is zero-dimensional, we should have $k \leq d$, which always holds when L is an involution, unless $c = 1$. Note that we focus on the case where the number of output elements is equal to the rate. This is the most challenging setting. Indeed, if the output size is smaller than the rate – as in some of the StarkWare challenges – then the preimage problem will typically have many solutions. This allows the attacker to partially or completely avoid the guessing phase. If further degrees of freedom remain after fixing $U_2 F(x)$ completely, one or more input elements may be fixed to an arbitrary value.

Complexity analysis. We now show that the total time cost of the attack can be estimated as

$$2\gamma\,(2\pi)^{-\omega/2}\,k^{2-\omega/2}\,e^{\omega k}\,3^{(\omega k+1)(R_F-1)}\,q^{t-d}.$$

Recall that the cost of solving the system of equations using Gröbner basis techniques is dominated by two steps:

1. Computing a Gröbner basis with respect to a total degree term order such as the degree reverse lexicographic (degrevlex) order. For standard reduction algorithms such as Faugère's F4 and F5, the time required for this step can be upper bounded by [BFS15]

$$T_{\text{gb}} = \mathcal{O}\left(\binom{D+k}{D}^{\omega}\right),$$

for k variables and with D an upper bound on the degree of the Gröbner basis elements. Here, ω is the asymptotic exponent of the time complexity of matrix multiplication.

2. Converting the degrevlex Gröbner basis to a Gröbner basis with respect to a lexicographic order. For the FGLM algorithm, the cost of this step can be estimated as [FGLM93]
$$T_{\text{fglm}} = \mathcal{O}(k \dim(\mathbb{F}_q[x_1, \ldots, x_k]/\mathcal{I})^\omega),$$
where $\mathcal{I}$ is the ideal corresponding the equations.

The time required to factor the univariate polynomials in the lexicographic Gröbner basis can be assumed to be negligible. Hence, the time cost of the attack is dominated by $q^{t-d}\,(T_{\text{gb}} + T_{\text{fglm}})$.

To set up a system of preimage equations for HadesMiMC, two diametrical approaches may be considered. In the first strategy, one attempts to minimize the number of variables by setting up a system of high-degree polynomials relating the input and output of the permutation. In the second approach, intermediate variables are introduced at every round, leading to a system of many low-degree equations. The latter strategy is usually preferred, as it leads to a lower degree D. However, a routine calculation shows that reducing the number of variables is more important for the present attack. Hence, we opt for the former approach.

Clearly, the Sbox layer of the first round may be ignored in the analysis. Furthermore, since the HadesMiMC design strategy states that the last linear layer can be omitted, the last round could also be ignored. Nevertheless, this is not the case for Starkad and Poseidon, so we do not take this into account in the analysis.

For each guess of $U_2F(x)$, the outputs $y_1, \ldots y_k$ may be expressed directly as a polynomial in the input (after the first Sbox layer) of degree 3^{R_F-1}. In general, bounding D is highly nontrivial. However, for regular systems, Macaulay's bound [BFS15, Mac02] yields
$$D \leq (3^{R_F-1} - 1)k + 1.$$
Furthermore, small-scale experiments suggest that this bound is tight for this particular system of equations. It is hard to obtain theoretical estimates of $\dim(\mathbb{F}_q[x_1, \ldots, x_k]/\mathcal{I})$, but small-scale experiments suggest that it scales as $\sim 3^{k(R_F-1)}$, which is consistent with recent results obtained by Faugère and Perret [FP19]. Since the FGLM algorithm is able to exploit sparse linear algebra methods [FM11], it is reasonable to assume that $T_{\text{fglm}} \lessapprox T_{\text{gb}}$.

Suppose that $3^{R_F-1} \gg k$. Following the reasoning in [BFS15, §1.3], it holds that
$$T_{\text{gb}} \leq \gamma\, k\, (D - 3^{R_F-1} + 1) \binom{k + D - 1}{D}^\omega \lessapprox \gamma\, k^2\, 3^{R_F-1} \binom{k + D - 1}{D}^\omega.$$
In the above, the parameters γ and ω are such that the computational cost of computing the row-reduced echelon form of an $m \times n$ matrix is γmn^ω. Stirling's approximation yields the estimate
$$\log \binom{k + D - 1}{D} = \log \binom{k3^{R_F-1}}{k} \approx k + k(R_F - 1)\log 3 - \log \sqrt{2\pi k}.$$
It follows that
$$T_{\text{gb}} \lessapprox \gamma\, (2\pi)^{-\omega/2}\, k^{2-\omega/2}\, e^{\omega k}\, 3^{(\omega k+1)(R_F-1)},$$
assuming that computing the reduced row-echelon form of an $m \times n$ matrix takes time γmn^ω. As discussed above, the total computational cost of the attack is then at most
$$2\gamma\, (2\pi)^{-\omega/2}\, k^{2-\omega/2}\, e^{\omega k}\, 3^{(\omega k+1)(R_F-1)}\, q^{t-d}. \tag{6}$$

Suppose that $2\gamma(2\pi)^{-\omega/2}k^{2-\omega/2} < 3C$ for some absolute constant C. For the total cost (6) to be below the security level $q^{\min\{k, c/2\}}$, it suffices that
$$\log_3 C + R_F + \omega k(\log_3 e + R_F - 1) + (t - d)\log_3 q \leq \min\{k, c/2\} \log_3 q.$$

Assuming $q \geq 3^{\omega R_F}$ we deduce the following lower bound for k:

$$k \geq \frac{(t-d)\log_3 q + R_F + \log_3 C}{\log_3 q - \omega\,(R_F + \log_3 e - 1)}.$$

Since $c = t - k$ we also obtain

$$\log_3 C + R_F + k[\omega(\log_3 e + R_F - 1) + (\log_3 q)/2] \leq (d - t/2)\log_3 q.$$

From this, we deduce the upper bound

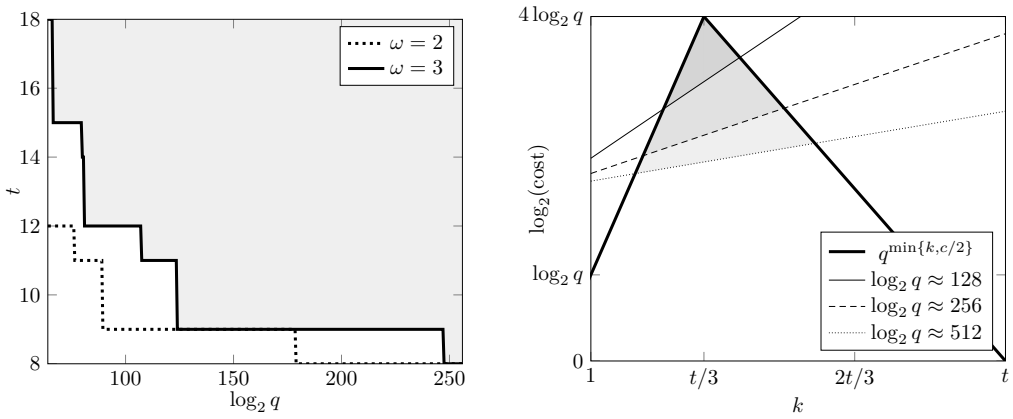
$$k \leq \frac{(d-t/2)\log_3 q - R_F - \log_3 C}{1/2\,\log_3 q + \omega(R_F + \log_3 e - 1)}.$$

We conclude that the preimage attack improves over the $q^{\min\{c/2,k\}}$ security level whenever

$$\frac{(t-d)\log_3 q + R_F + \log_3 C}{\log_3 q - \omega\,(R_F + \log_3 e - 1)} \leq k \leq \frac{(d-t/2)\log_3 q - R_F - \log_3 C}{1/2\,\log_3 q + \omega(R_F + \log_3 e - 1)},$$

where C is a constant close to one. If $k \leq 20$, one can take $C = 3$.

For example, for an involutive L, $R_F = 8$ and an arbitrary number of partial rounds, Figure 12a shows for which choices of q and t an improvement over the generic security of the sponge construction is obtained. The insecure instances are shaded in gray. Note that this domain corresponds to a conservative estimate for the cost of row-echelon reduction, i.e. $\omega = 3$ and $\gamma = 3/2$. The cost itself is shown in Figure 12b. We stress that these figures correspond to the most challenging case, i.e. assuming that the hash output is of length k and no shorter.



(a) Minimum t such that the cost is better than generic for some choice of k.

(b) Cost for different values of the rate k with $t = 12$ and $\omega = 3$.

Figure 12: Cost analysis of the preimage attack on HadesMiMC with an involutive linear layer and $R_F = 8$. The shaded areas correspond to parameters for which the attack improves over the $q^{\min\{k,c/2\}}$ security level.

For the concrete Starkad and Poseidon instances specified in Table 1, we obtain better-than-generic attacks on some variants assuming that the hash output has length $c \leq k$. Indeed, provided that $c \leq d/2 = t/2 - 1$, a sufficiently large number of preimages is likely to exist so that it is no longer necessary to guess $U_2F(x)$. In addition, input variables may be fixed until only c free variables remain. This leads to a computational

cost of $2\gamma\,(2\pi)^{-\omega/2}\,c^{2-\omega/2}\,e^{\omega c}\,3^{(\omega c+1)(R_F-1)}$. Note that, for these instances, we do not obtain relevant preimage attacks when the output size exceeds $t/2-1$.

Table 8: Overview of the computational cost (measured in $\mathbb{F}_q$ operations) of the preimage attack on different instances of Poseidon and Starkad, assuming an involutive linear layer. These estimates assume that the hash output has length c. For the variants 128-a, 128-b and 128-d, the attack does not improve over the generic security level of the sponge.

Variant	c	Computational cost	
		$\omega \approx 2.8$	$\omega = 3$
128-c	2	$2^{80.0}$	$2^{84.3}$
128-e	1	$2^{44.2}$	$2^{46.3}$
256-a	4	$2^{150.9}$	$2^{160.3}$
256-b	4	$2^{150.9}$	$2^{160.3}$

5.4 Remarks on algebraic distinguisher related to the CICO problem

The so-called *Constrained-Input Constrained-Output (CICO) problem* has been analyzed in the "Report on Algebraic attacks" [FP19] which studies in detail the resistance of STARK-friendly primitives to algebraic attacks. Here, we focus on the most interesting setting of the CICO problem in our context, which corresponds to a preimage-attack scenario. In this case, the number of fixed elements in the input and output of the permutation, denoted by k in [FP19], corresponds to the capacity of the sponge. Therefore, with the parameters we consider, the number of fixed inputs is $k = 4$.

The variant of the CICO problem corresponding to a preimage attack is then the following (Algorithm 3), where π denotes the inner permutation in the sponge construction.

Algorithm 3 CICO problem in a preimage-attack scenario.

Input: $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k) \in \mathbb{F}_q^k$
Find $x_1, \ldots, x_{t-k} \in \mathbb{F}_q^{t-k}$ such that
$\pi(a_1, \ldots, a_k, x_1, \ldots, x_{t-k}) = (b_1, \ldots, b_k, y_1, \ldots y_{t-k})$ for some $y_1, \ldots y_{t-k} \in \mathbb{F}_q^{t-k}$

However, for the parameters we consider, $t > 2k$, which implies that the corresponding algebraic system is underdefined. Since we need a single solution, we will instead consider the following variant (Algorithm 4) with k free variables only.

Algorithm 4 Alternative CICO problem in a preimage-attack scenario.

Input: $(a_1, \ldots, a_{t-k}) \in \mathbb{F}_q^{t-k}$ and $(b_1, \ldots, b_k) \in \mathbb{F}_q^k$
Find $x_1, \ldots, x_k \in \mathbb{F}_q^k$ such that
$\pi(a_1, \ldots, a_{t-k}, x_1, \ldots, x_k) = (b_1, \ldots, b_k, y_1, \ldots y_{t-k})$ for some $y_1, \ldots y_{t-k} \in \mathbb{F}_q^{t-k}$

The main result in [FP19] concerning the complexity of the previous CICO problem for Starkad and Poseidon is Theorem 10, stating that the degree of the univariate polynomial of the lexicographical Gröbner basis of the algebraic system corresponding to the CICO problem is

$$D = 3^{kR_f + R_P - 2k + 1}\,. \tag{7}$$

Most notably, an important property is that this degree D is independent from t, i.e. from the block-size of the permutation which is considered.

A first question raised by this result is the determination of the time complexity required for solving the corresponding CICO problem. It is claimed in [FP19] that this complexity is roughly D^2, but that it would be reduced to D when $k \leq 2$. However, it remains unclear whether this complexity is not under-estimated and whether the first step in the algorithm for solving the system (which involves the degree of regularity of the system) is not the bottleneck in this specific case.

Moreover, a more general question arises on the cryptographic significance of the value of the degree D determined by (7). Actually, when π is a randomly chosen permutation of $\mathbb{F}_q^t$, do we expect that the degree of the univariate polynomial of the lexicographical Gröbner basis of the algebraic system corresponding to the CICO problem is close to q^k? In other words, we wonder whether the fact that $D < q^k$ can be seen as a distinguishing property for π (even if the complexity for solving the system is not lower than the security level). For instance, for the versions of Starkad and Poseidon aiming at 256-bit security, which operate on $t = 14$ elements in a field of size close to 2^{128}, we get that $D \simeq 2^{170}$ for $k = 4$. It seems that this property could be considered as a relevant distinguisher.

5.5 Summary

Table 9 summarizes the previous results on Poseidon and Starkad. None of the distinguishers we have exhibited covers the total number of rounds proposed in the StarkWare challenges. However, the security margin does not seem to be very high. Especially, it appears that some instances of HadesMiMC, like the ones with an MDS matrix having a low order, are weaker than expected in the sense that the partial rounds can be bypassed by the attacker.

Table 9: Distinguishers on the HadesMiMC permutations and attacks on the corresponding sponge hash functions. The variant aiming at 128-bit security operate on $t = 12$ elements in $\mathbb{F}_q$ with $q = 2^{61} + 20 \times 2^{32} + 1$ or $q = 2^{63}$. The variant aiming at 256-bit security operate on $t = 14$ elements in $\mathbb{F}_q$ with $q = 2^{125} + 266 \times 2^{64} + 1$ or $q = 2^{125}$. The last attack (*) only applies when the linear layer has a low multiplicative order.

Primitive (security)	Rounds		Attack			
			Type	Rounds	Cost	Sect.
Poseidon (128 bits)	8+40	permutation	ZS distinguisher	6+45	2^{61}	5.2
Starkad (128 bits)	8+43	permutation	ZS distinguisher	6+46	2^{61}	5.2
Poseidon (256 bits)	8+83	permutation	ZS distinguisher	6+87	2^{125}	5.2
		hash function*	preimages	8+any	2^{160}	5.3
Starkad (256 bits)	8+83	permutation	ZS distinguisher	6+87	2^{125}	5.2
		hash function*	preimages	8+any	2^{160}	5.3

6 Marvellous (Vision and Rescue)

6.1 Description

Marvellous includes two families of block ciphers proposed by Aly *et al.* [AAB+19] very recently. These two ciphers share the same structure: they both follow the classical

SPN construction (with full Sbox layers) and operate on an internal state composed of t elements in a large finite field $\mathbb{F}_q$. Their linear layer consists of a $t \times t$ MDS matrix which guarantees a fast diffusion. However, the Sbox layers are very different in the two ciphers.

- In Vision, which is the primitive defined over a binary field $\mathbb{F}_{2^n}$, the Sbox is composed of the inversion in the field $\mathbb{F}_{2^n}$, and of an affine transformation B over $\mathbb{F}_2^n$ which is represented by a univariate polynomial of degree 4 over $\mathbb{F}_{2^n}$, namely $B(X) = B_0 + B_1X + B_2X^2 + B_4X^4$ with $B_i \in \mathbb{F}_{2^n}$. At each odd round[4], the overall Sbox consists of $x \mapsto x^{-1}$ followed by B, while at each even round, it consists of $x \mapsto x^{-1}$ followed by B^{-1}.
- In Rescue, which is the primitive defined over a prime field $\mathbb{F}_p$, the Sbox is derived from the cube mapping in $\mathbb{F}_p$, with $p \bmod 3 \neq 1$. At each odd round, the Sbox corresponds to $x \mapsto x^3$, while at each even round, it corresponds to its inverse. A specificity of Rescue is that this power function is *not* composed with an affine layer (contrary to most ciphers, like the AES, which use an Sbox defined by a power function).

It is worth noticing that all Sboxes used in these permutations, namely x^3, $x^{1/3}$ and x^{-1} have differential uniformity 2 (since $q = 2^n$ with n odd in Vision and $q = p > 3$ in Rescue), which is optimal [HRS99, Theorem 3]. Exactly as for the full rounds in HadesMiMC, any *differential characteristic* over 6 rounds or more has probability lower than 2^{-tn}.

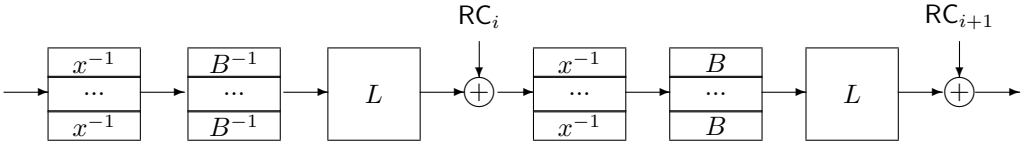


Figure 13: Two rounds of Vision.

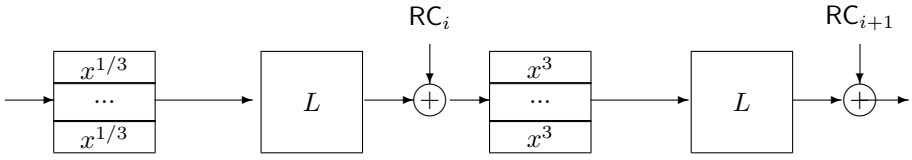


Figure 14: Two rounds of Rescue.

For the parameters we focus on, i.e., $t = 12$ and $q \simeq 2^{64}$, the number of rounds of both primitives is $R = 20$, which means that the permutation contains ten even rounds and ten odd rounds.

6.2 Degree of the permutation

Since the number of rounds in these two primitives is much smaller than for the previous ones, we need to analyze carefully how the degree of Vision and Rescue increases with the number of rounds.

[4]We assume that the rounds are numbered from 0 to $(R-1)$.

Algorithm 5 VISION with block-size t and R rounds (with R even).

Input: $(x_0, \ldots, x_{t-1}) \in \mathbb{F}_{2^n}^t$
  for r from 0 to $(R/2-1)$ **do**
    for i from 0 to $(t-1)$ **do**
      $y_i \leftarrow B^{-1}\left((x_i + \mathsf{RC}_{2r,i})^{-1}\right)$
    end for
    $(x_0, \ldots, x_{t-1}) \leftarrow L(y_0, \ldots, y_{t-1})$
    for i from 0 to $(t-1)$ **do**
      $y_i \leftarrow B\left((x_i + \mathsf{RC}_{2r+1,i})^{-1}\right)$
    end for
    $(x_0, \ldots, x_{t-1}) \leftarrow L(y_0, \ldots, y_{t-1})$
  end for
Output: $(x_0, \ldots, x_{t-1})$

Algorithm 6 RESCUE with block-size t and R rounds (with R even).

Input: $(x_0, \ldots, x_{t-1}) \in \mathbb{F}_p^t$
  for r from 0 to $(R/2-1)$ **do**
    for i from 0 to $(t-1)$ **do**
      $y_i \leftarrow (x_i + \mathsf{RC}_{2r,i})^{1/3}$
    end for
    $(x_0, \ldots, x_{t-1}) \leftarrow L(y_0, \ldots, y_{t-1})$
    for i from 0 to $(t-1)$ **do**
      $y_i \leftarrow (x_i + \mathsf{RC}_{2r+1,i})^3$
    end for
    $(x_0, \ldots, x_{t-1}) \leftarrow L(y_0, \ldots, y_{t-1})$
  end for
Output: $(x_0, \ldots, x_{t-1})$

VISION. As previously detailed, any output coordinate in VISION can be seen as a multivariate polynomial in t variables over $\mathbb{F}_{2^n}$ or as a collection of n binary polynomials in nt variables. As a polynomial over $\mathbb{F}_{2^n}$, the inverse mapping corresponds to $x \mapsto x^{2^n-2}$ and has then the highest possible degree for a permutation over $\mathbb{F}_{2^n}$. This ensures that the degree of output coordinates in each input variable is maximal after a single round. Then, the MDS layer guarantees that different variables are multiplied together in the next round, which implies that the maximal degree for the multivariate polynomial is reached after a few rounds only.

As a binary function, i.e. as a permutation of $\mathbb{F}_2^n$, the inverse mapping has degree $(n-1)$ (which is the number of ones in the binary representation of the integer (2^n-2)). This corresponds to the binary degree of the two Sboxes (since the affine transformations B and B^{-1} do not change the binary degree). If we apply the best known bound on the degree of an SPN, which is Theorem 3.2 in [BC13], with the parameters we focus on, we get that the degree of the algebraic normal form of any output bit of the permutation after two rounds of VISION is upper bounded by 744. But the upper bound for three rounds reaches the highest possible degree for a permutation of $\mathbb{F}_2^{nt}$, which is 755 in our case. It is worth noticing that, in both cases, i.e. over $\mathbb{F}_{2^n}$ and over $\mathbb{F}_2$, the degree increases even faster than for other primitives based on the inverse mapping like the AES or RIJNDAEL. For instance, it has been proved in [BC13, Section V.B] that six rounds of RIJNDAEL-256 are not enough to achieve the maximal degree. The reason is that the diffusion in RIJNDAEL and in the AES is slower than in VISION because the MDS matrix in RIJNDAEL operates on the columns of the internal state, while it operates on the full state in VISION. In other

words, the so-called *Superbox view* [DR06] does not apply to Vision. This explains why the growth of the binary degree in Vision is very fast.

Rescue. Studying the properties of the polynomial representation of the permutation in the case of Rescue is important, for instance to ensure that the polynomial does not simplify because of the interaction between the two Sboxes which are inverse to each other. But this analysis is more difficult than for Vision, first because it involves both the cube mapping and its inverse, $x^{\frac{2p-1}{3}}$. Also, since the field characteristic is very large, the resulting polynomials are much denser than in the binary case (where many terms cancel modulo 2). Nevertheless, we can study the degree of the coordinates of the output of Rescue after two rounds (i.e., after one double-round). Let us activate a single input coordinate and denote by x the value of this input added to the corresponding coordinate of RC_0. Then, the coordinates of the output of the first round can be expressed as $(\lambda x^{1/3} + \mu)$ where μ depends on the other inputs of the permutation. Therefore, after the 2nd-round Sbox layer, each output coordinate can be expressed as

$$(\lambda x^{1/3} + \mu)^3$$

which is a univariate polynomial which contains only the monomials $x^{1/3}$, $x^{2/3}$, x and a constant term. The same property obviously holds for each coordinate of the output of the second round. A simple observation now is that $x^{2/3}$ corresponds to the monomial x^s with

$$s = 2 \times \frac{2p-1}{3} = p - 1 + \frac{p+1}{3} = \frac{p+1}{3} \bmod (p-1) \, .$$

Therefore, $x^{2/3} = x^{\frac{p+1}{3}}$, implying that any coordinate of the output of the second round of Rescue is a univariate polynomial in x which involves monomials of degree 0, 1, $\frac{p+1}{3}$ and $\frac{2p-1}{3}$. Its degree in x is then equal to $\frac{2p-1}{3}$, and is the same as the degree in x of the first round.

Consequently, if we express any coordinate of the output of the second round of Rescue as a multivariate polynomial in

$$\mathbb{F}_p[x_1, \ldots, x_t] / \left(x_1^p - x_1, \ldots, x_t^p - x_t\right),$$

then this polynomial involves all monomials of the form

$$x_i^{a/3} x_j^{b/3} x_k^{c/3} \text{ with } a + b + c \leq 3 \, .$$

It follows that this polynomial has multivariate degree at most $3 \times \frac{2p-1}{3}$.

This shows that the degree in each input variable does not increase between the first and the second round of Rescue. However, this does not apply when more rounds are considered. Indeed, our simulations tend to show that, when k inputs are activated, then the output of the fourth round has degree $k(p-1)$, which would imply that mounting an integral attack similar to the one against GMiMC is not possible on more than four rounds.

6.3 Algebraic distinguisher on Rescue related to the CICO problem

The complexity of the CICO problem corresponding to the inner permutation of Rescue has not been analyzed in the "Report on Algebraic Attacks" [FP19]. The situation in Rescue is quite similar to the one in Poseidon with only 20 full rounds. The only difference is that every even round uses $x \mapsto x^{1/3}$ as an Sbox, instead of $x \mapsto x^3$. However, the *relations* derived for these rounds in the backward direction also have univariate degree 3.

Then, it would be interesting to have an analogous result for Rescue of Theorem 10 in [FP19]. Our intuition is that, for Rescue, the existence of a relation of univariate degree 3 for every round could also be exploited. But it seems that, in this case, the degree D would depend on t in the case of Rescue, and not only on k as for Poseidon.

6.4 Summary

We have not been able so far to exhibit any relevant weakness on Vision and on Rescue, even if we tried to exploit some properties of their polynomial representation, or the fact that the Sbox is a power mapping in Rescue. Our conclusion is then that Vision cannot be easily broken with the state-of-the-art cryptanalytic techniques. The situation is different for Rescue since only a few attacks apply to symmetric primitives defined over a prime field and a much deeper and longer effort is needed to assess the security of this new type of primitives.

7 Conclusions

Remark 1. The notion of symmetric primitives over a prime field has been introduced too recently to be able to provide a rigorous assessment on its security. While decades of research have produced efficient cryptanalytic tools and security criteria for primitives defined over $\mathbb{F}_2$, we do not have the right tools to analyze primitives over $\mathbb{F}_p$. As an illustration of this situation, the notion of linear cryptanalysis and its variants is not clearly established in odd characteristic (see e.g. [BSV07] for a first discussion). Also, while higher-order differential attacks seemed to be relevant in the binary case only, we have shown in this report that similar methods apply in any characteristic, and can even be extended to multiplicative subgroups of $\mathbb{F}_p^\times$.

Our preliminary analysis allowed us to establish a clear hierarchy between the security levels offered by the five primitives we considered:

- GMiMC, with the parameters we focused on, suffers from many weaknesses which were not expected by its designers. The existence of distinguishers with complexity 2^{48} over the full permutation invalidates the security argument provided by the indifferentiability proof of the sponge construction, which assumes that the inner permutation behaves as a randomly chosen permutation. Even when the presented weaknesses are distinguishers for the permutation only, we expect that related properties can be exploited in the future for mounting attacks on the corresponding hash function. Indeed, we have already exhibited a practical collision attack covering 40 rounds (out of 101) of the hash function. For all these reasons, our opinion is that this version of GMiMC should not be used in practice.
- We did not find any distinguisher on the full versions of Starkad and Poseidon for the parameters we focused on, but we exhibited some distinguishers on round-reduced versions with a number of rounds which is not very far from the total number. Most notably, we have shown that, in some specific cases (which do *not* include the version proposed in the StarkWare challenge), all inner rounds of the permutation can be bypassed by the attacker. This leads to a preimage attack on the full hash function for some of the parameters proposed in the StarkWare challenges. In light of this observation, the fact that the security evaluation of HadesMiMC against attacks of algebraic nature often assumes that the partial rounds offer the same security as the full rounds may appear as too optimistic a hypothesis. Therefore, our opinion is that HadesMiMC may not have a large enough security margin to avoid the existence of distinguishers with complexity below the claimed security level.

- We did not find any weakness on RESCUE so far. For the reason mentioned in Remark 1, it is clear that a fuller assessment of the security of RESCUE requires a much longer cryptanalytic effort, including the development of new tools for analyzing symmetric primitives in odd characteristic. Also, it is important to note that the security of RESCUE with respect to attacks based on Gröbner bases and polynomial system solving has not been studied in [FP19]. The results on POSEIDON in [FP19] raise several issues which must be addressed to estimate the resistance of RESCUE to this type of attacks. We therefore recommend to extend research into the algebraic cryptanalysis and general symmetric cryptanalysis of this construction.

- VISION is by far the primitive we trust the most. We did not find any weakness on VISION. This seems rather expected since VISION looks very much like the AES. We then believe that it would be rather surprising that a concrete collision or preimage attack on the sponge using VISION, with the parameters we focused on, would appear in the next three years. If that were the case, then it would probably also have consequences on the security of other primitives, like the AES.

In other words, for the parameters we studied, we see VISION and RESCUE as the two most secure options, for binary and prime fields respectively.

It is also worth noticing that the previously detailed hierarchy between the five SFH-functions might be related to the number of nonlinear operations performed within the function, as detailed in Table 10. Indeed, for the considered parameters, VISION is the permutation with the largest number of Sboxes, and it also uses a stronger Sbox than the other ones (in the sense that it has a much higher degree both in $\mathbb{F}_{2^n}$ and in $\mathbb{F}_2$). Its counterpart over a prime field, RESCUE, has the same number of Sboxes, but half of these Sboxes correspond to x^3 (which is weaker than x^{-1}) and the other half to $x^{1/3}$ (which seems to offer a better security than x^3 against some attacks). Finally, GMIMC is the permutation with the smallest number of Sboxes, and all its Sboxes correspond to the cube mapping. The fact that it offers the lowest security level among these five functions is then not surprising.

Table 10: Number of Sboxes over $\mathbb{F}_q$ for each permutation. The Sbox corresponds to the cube mapping for GMIMC and HADESMIMC, to the inversion for VISION, to the cube mapping or its inverse for RESCUE.

Primitive	Number of Sboxes	For Variant 128-d
GMIMC	R	101
POSEIDON	$tR_f + R_P$	136
STARKAD	$tR_f + R_P$	139
VISION and RESCUE	tR	240

References

[AAB+19] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. Cryptology ePrint Archive, Report 2019/426, 2019. https://eprint.iacr.org/2019/426.

[ACG+19] Martin R. Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic

cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 371–397. Springer, Heidelberg, December 2019.

[AD18] Tomer Ashur and Siemen Dhooghe. MARVELlous: a STARK-friendly family of cryptographic primitives. Cryptology ePrint Archive, Report 2018/1098, 2018. https://eprint.iacr.org/2018/1098.

[AGP+19a] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, Heidelberg, September 2019.

[AGP+19b] Martin R. Albrecht, Lorenzo Grassi, Leo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. Cryptology ePrint Archive, Report 2019/397, 2019. https://eprint.iacr.org/2019/397.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219. Springer, Heidelberg, December 2016.

[AKK+10] Jean-Philippe Aumasson, Emilia Käsper, Lars R. Knudsen, Krystian Matusiewicz, Rune Steinsmo Ødegård, Thomas Peyrin, and Martin Schläffer. Distinguishers for the compression function and output transformation of Hamsi-256. In Ron Steinfeld and Philip Hawkes, editors, *ACISP 10*, volume 6168 of *LNCS*, pages 87–103. Springer, Heidelberg, July 2010.

[AM09] Jean-Philippe Aumasson and Willi Meier. Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi. Presented at the rump session of Cryptographic Hardware and Embedded Systems – CHES 2009, 2009.

[AMPH14] Jean-Philippe Aumasson, Willi Meier, Raphael C.-W. Phan, and Luca Henzen. *The Hash Function BLAKE*. Information Security and Cryptography. Springer, 2014.

[AP11] Andrea Agnesse and Marco Pedicini. Cube attack in finite fields of higher order. In Colin Boyd and Josef Pieprzyk, editors, *AISC 20111*, volume 116 of *CRPIT*, pages 9–14. Australian Computer Society, 2011.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 430–454. Springer, Heidelberg, April 2015.

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) s-boxes to optimize probability of differential characteristics. *IACR Transactions on Symmetric Cryptology*, 2017(4):99–129, 2017.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. https://eprint.iacr.org/2018/046.

[BC13] Christina Boura and Anne Canteaut. 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.

[BCC+09] Emmanuel Bresson, Anne Canteaut, Benoît Chevallier-Mames, Christophe Clavier, Thomas Fuhr, Aline Gouget, Thomas Icart, Jean-Francois Misarsky, Maria Naya-Plasencia, Pascal Paillier, Thomas Pornin, Jean-Rene Reinhard, Celine Thuillet, and Marion Videau. Indifferentiability with distinguishers: Why Shabal does not require ideal ciphers. Cryptology ePrint Archive, Report 2009/199, 2009. http://eprint.iacr.org/2009/199.

[BCD11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and Luffa. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 252–269. Springer, Heidelberg, February 2011.

[BDD+15] Achiya Bar-On, Itai Dinur, Orr Dunkelman, Virginie Lallemand, Nathan Keller, and Boaz Tsaban. Cryptanalysis of SP networks with partial non-linear layers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 315–342. Springer, Heidelberg, April 2015.

[BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT Hash Workshop*, 2007. https://keccak.team/files/SpongeFunctions.pdf.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, Heidelberg, April 2008.

[BDPV09] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak sponge function family - main document. Submission to NIST, 2009. https://keccak.team/obsolete/Keccak-main-2.0.pdf.

[BFL11] Charles Bouillaguet, Pierre-Alain Fouque, and Gaëtan Leurent. Security analysis of SIMD. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *LNCS*, pages 351–368. Springer, Heidelberg, August 2011.

[BFS15] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 gröbner basis algorithm. *Journal of Symbolic Computation*, 70:49–70, 2015.

[Bon19] Xavier Bonnetain. Collisions on Feistel-MiMC and univariate GMiMC. Cryptology ePrint Archive, Report 2019/951, 2019. https://eprint.iacr.org/2019/951.

[BSV07] Thomas Baignères, Jacques Stern, and Serge Vaudenay. Linear cryptanalysis of non binary ciphers. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *SAC 2007*, volume 4876 of *LNCS*, pages 184–211. Springer, Heidelberg, August 2007.

[Can16] Anne Canteaut. Lecture notes on cryptographic boolean functions. Lecture Notes. Available at https://www.rocq.inria.fr/secret/Anne.Canteaut/poly.pdf, 2016.

[CCF+18] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology*, 31(3):885–916, July 2018.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A cipher with low ANDdepth and few ANDs per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 662–692. Springer, Heidelberg, August 2018.

[DEM16] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Higher-order cryptanalysis of LowMC. In Soonhak Kwon and Aaram Yun, editors, *ICISC 15*, volume 9558 of *LNCS*, pages 87–101. Springer, Heidelberg, November 2016.

[DLMW15] Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang. Optimized interpolation attacks on LowMC. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 535–560. Springer, Heidelberg, November / December 2015.

[DR06] Joan Daemen and Vincent Rijmen. Understanding two-round differentials in AES. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 78–94. Springer, Heidelberg, September 2006.

[FGLM93] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[FM11] Jean-Charles Faugère and Chenqi Mou. Fast algorithm for change of ordering of zero-dimensional gröbner bases with sparse multiplication matrices. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, pages 115–122, 2011.

[FP19] Jean-Charles Faugère and Ludovic Perret. Algebraic attacks against STARK-Friendly Ciphers. Personal communication, 2019.

[FSK10] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. *Cryptography Engineering - Design Principles and Practical Applications*. Wiley, 2010.

[GGNS13] Benoît Gérard, Vincent Grosso, María Naya-Plasencia, and François-Xavier Standaert. Block ciphers that are easier to mask: How far can we go? In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *LNCS*, pages 383–399. Springer, Heidelberg, August 2013.

[GKK+19] Lorenzo Grassi, Daniel Kales, Dmitry Khovratovich, Arnab Roy, Christian Rechberger, and Markus Schofnegger. Starkad and Poseidon: New hash functions for zero knowledge proof systems. Cryptology ePrint Archive, Report 2019/458, 2019. https://eprint.iacr.org/2019/458.

[GLR+19] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. Cryptology ePrint Archive, Report 2019/1107, 2019. https://eprint.iacr.org/2019/1107.

[HRS99] Tor Helleseth, Chunming Rong, and Daniel Sandberg. New families of almost perfect nonlinear power mappings. *IEEE Trans. Information Theory*, 45(2):475–485, 1999.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 315–324. Springer, Heidelberg, December 2007.

[KU08] Kiran S. Kedlaya and Christopher Umans. Fast modular composition in any characteristic. In *49th FOCS*, pages 146–155. IEEE Computer Society Press, October 2008.

[KW02] Lars R. Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *FSE 2002*, volume 2365 of *LNCS*, pages 112–127. Springer, Heidelberg, February 2002.

[Lai94] Xuejia Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday*. Kluwer Academic Publishers, 1994.

[Mac02] Francis Sowerby Macaulay. Some formulae in elimination. *Proceedings of the London Mathematical Society*, 1(1):3–27, 1902.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 311–343. Springer, Heidelberg, May 2016.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76. Springer, 2011.

[Nyb94] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 55–64. Springer, Heidelberg, May 1994.

[RST18] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of low-data instances of full LowMCv2. *IACR Trans. Symm. Cryptol.*, 2018(3):163–181, 2018.

[SDMO12] Mahdi Sajadieh, Mohammad Dakhilalian, Hamid Mala, and Behnaz Omoomi. On construction of involutory MDS matrices from Vandermonde matrices in GF(2^q). *Designs, Codes and Cryptography*, 64(3):287–308, Sep 2012.

[SWMSP17] Ana Sălăgean, R. Winter, Matei Mandache-Sălăgean, and Raphael C.-W. Phan. Higher order differentiation over finite fields with applications to generalising the cube attack. *Designs, Codes and Cryptography*, 84(3):425–449, 2017.

[Udo19] Alexei Udovenko. Personal communication, November 2019.

[YMT97] AM Youssef, S Mister, and SE Tavares. On the design of linear transformations for substitution permutation encryption networks. In *Workshop on Selected Areas of Cryptography (SAC96)*, pages 40–48, 1997.

A Weak Cauchy matrices

The linear layers of Starkad and Poseidon are chosen such that $L_{i,j} = 1/(x_i + x_j + a)$ where $x_1, \ldots, x_t$ are distinct elements of $\mathbb{F}_q$ [GKK+19]. The following result shows that, for Starkad instances with t a power of two, there exist weak choices of $x_1, \ldots, x_t$ that enable the preimage attack from Section 5.3.

Theorem 1. *Let $G = \{x_1, \ldots, x_t\}$ be an additive subgroup of $\mathbb{F}_{2^n}$ of order t and let $a \in \mathbb{F}_{2^n} \setminus G$. For the Cauchy matrix $L \in \mathbb{F}_{2^n}^{t \times t}$ defined by $L_{i,j} = 1/(x_i + x_j + a)$, it holds that $L^2 = b^2 I$ with $b = \sum_{i=1}^{t} 1/(x_i + a)$.*

Proof. Observe that

$$(L^2)_{i,j} = \sum_{k=1}^{t} \frac{1}{x_i + x_k + a} \times \frac{1}{x_j + x_k + a} = \sum_{x \in a+G} \frac{1}{x(x + x_i + x_j)}.$$

For $i = j$, the result is clear. It suffices to prove that $(L^2)_{i,j} = 0$ for $i \neq j$. Since $x_i \neq x_j$ for $i \neq j$, we have $g = x_i + x_j \in G \setminus \{0\}$. Finally, it holds that

$$(L^2)_{i,j} = \sum_{x \in a+G} \frac{1}{x(x+g)} = \frac{1}{g} \sum_{x \in a+G} \left(\frac{1}{x} + \frac{1}{x+g} \right) = 0.$$

□

A special case of Theorem 1 is discussed by Youssef *et al.* [YMT97, §3.2]. For an extension $\mathbb{F}_2(\zeta) \supset \mathbb{F}_2$ of degree n, they show that the choice $x_i = \sum_{j=1}^{\log_2 t} \mathtt{i}_j \, \zeta^{j-1}$ with $\mathtt{i}_1, \ldots, \mathtt{i}_{\log_2 t}$ the binary digits of i results in a Cauchy matrix L such that $L^2 = b^2 I$ for some $b \in \mathbb{F}_2(\zeta)$.