New Hash Function Designs for Modern Proof Systems

No Author Given

No Institute Given

Abstract. The are of practical proof systems, like SNARKs, STARKs, or Bulletproofs, is seeing a very dynamic development. Many use-cases of such systems involve, often as their most expensive apart, proving the knowledge of a preimage under a certain cryptographic hash function.

In this paper we present a modular framework and concrete instances of cryptographic hash functions which either work natively with GF(p) objects or on binary strings.

Performance analysis shows substantial benefits in all use-cases.

Table of Contents

1 Introduction

2 Starkad-, Persephone- and Poseidon-Hash Functions

In the following we propose three hash functions:

- the hash function[1] Starkad-Hash for the *binary case* is constructed by instantiating a sponge construction [7] with Starkad-Permutation – denoted by Starkad^{π};
- the hash function[2] Persephone-Hash for the *prime case* is constructed by instantiating a sponge construction [7] with Persephone-Permutation – denoted by Persephone^{π};
- the hash function[3] Poseidon-Hash for the *prime case* is constructed by instantiating a sponge construction [7] with Poseidon-Permutation – denoted by Poseidon^{π}.

Both permutations are variants of HadesMiMC, a block cipher proposed in [30] instantiated by a fixed key, e.g. 0^{κ}.

We recall that when the internal permutation $\mathcal{P}$ of an N-bit sponge function (composed of c-bit capacity and r-bit bitrate: $N = c + r$) is modeled as a randomly chosen permutation, it has been proven by Bertoni *et al.* [7] to be indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$. In other words, a sponge with a capacity of c provides $2^{c/2}$ collision and $2^{c/2}$ (second) preimage resistance. Given a permutation of size N and a desired security level s, we can hash $r = N - 2s$ bits per call to the permutation. Following this design strategy, we choose the number of rounds of the inner permutations Persephone^{π} and Starkad^{π} in order to ensure that such permutation does *not* exhibits non-generic property[4].

As usual, the message is first padded according to the sponge specification so that the number of message blocks is a multiple of r, where r is the rate in the sponge mode. In our case, we use Persephone^{π} or Starkad^{π} or Poseidon^{π} permutation with a fixed key, where $N \geq 4 \cdot s$ (s is the security level). For Persephone-Hash-256 (analogous for Starkad-Hash-256 and Poseidon-Hash-256), we thus use Persephone permutation with $N = n \cdot t \geq 1024$. The rate and

[1] *About the name:* Starkad was a legendary hero in Norse mythology. connection with ???

[2] *About the name:* Persephone was the consort of Hades. When Persephone is down in the Underworld with her husband, the winter falls upon the earth.

[3] *About the name:* Poseidon – brother of Zeus and Hades – was god of the Sea and other waters, of earthquakes and of horses.

[4] In other words, such permutation can not be distinguished from a randomly-drawn permutation.

the capacity are chosen as 512. This choice allows e.g. for processing the same amount of input bits as SHA-256 (512 bits) while at the same time offering collision security and (second) preimage security of 256 bits. Similar considerations hold as well for Persephone-Hash-128 and/or Starkad-Hash-128 and/or Poseidon-Hash-128.

2.1 Description of the Hades Strategy and Hades-like Permutation

2.2 Hades Strategy

(Cryptographic) Permutations are typically designed by iterating an efficiently implementable round function many times in the hope that the resulting composition behaves like a randomly drawn permutation. In general, *the same round function is iterated enough times to make sure that any symmetries and structural properties that might exist in the round function vanish.*

Instead of considering the same round function in order to construct the cipher (to be more precise, the same non-linear layer for all rounds), in [30] authors propose to consider *a variable number of S-Boxes per round*, that is, to use different S-Box layers in the round functions.

Similar to any other SPN design, each round of a cipher based on Hades is composed of three steps:

1. *Add-Round Key* - denoted by $ARK(\cdot)$;
2. *SubWords* operation - denoted by S-Box$(\cdot)$;
3. *MixLayer* - denoted by $M(\cdot)$.

A final round key addition is then performed, and the final MixLayer operation can be omitted (we sometimes include it in this description for simplicity):

$$\underbrace{ARK \to \text{ S-Box} \to M}_{1st \text{ round}} \to \ldots \to \underbrace{ARK \to \text{ S-Box} \to M}_{(R-1)\text{-}th \text{ round}} \to \underbrace{ARK \to \text{ S-Box}}_{R\text{-}th \text{ round}} \to ARK$$

The crucial property of HadesCubic is that *the number of S-Boxes per round is not the same for every round*:

– a certain number of rounds - denoted by R_F - has a *full* S-Box layer, i.e., t S-Box functions;

– a certain number of rounds - denoted by R_P - has a *partial* S-Box layer, i.e., $1 \leq s < t$ S-Boxes and $(t - s)$ identity functions.

In the following, we limit to consider only the case $s = 1$, that is, R_P rounds have a single S-Box per round and $t - 1$ identity functions.

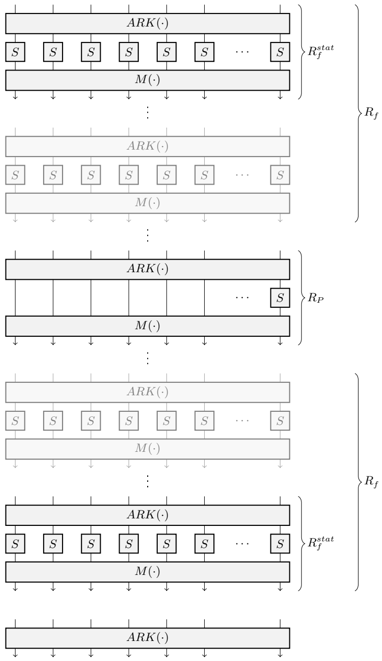

Fig. 1: Construction of Hades (the final matrix multiplication can be omitted).

In more details, assume $R_F = 2 \cdot R_f$ is an even number[5]. Then

- the first R_f rounds have a full S-Box layer,
- the middle R_P rounds have a partial S-Box layer (i.e., 1 S-Box layer),
- the last R_f rounds have a full S-Box layer.

Figure 1 shows the strategy Hades. Note that the rounds with a partial S-Box layer are "masked" by the rounds with a full S-Box layer, which means that an attacker should not (directly) take advantage of the rounds with a partial S-Box layer.

Behind Hades Strategy. The crucial point of our design is that it contains *both rounds with full S-Box layers and rounds with partial S-Box layers.* This allows to provide *simpler argumentation about the security against statistical attacks* than the one proposed for P-SPN ciphers.

[5] $R_F = 2 \cdot R_f$ is even in order to have a "symmetric" permutation. Note that some attacks – like the statistical ones – have the same performance both in the forward and in the backward direction. Thus a "symmetric" permutation with $R_F = 2 \cdot R_f$ guarantees the same security against these attacks both in the chosen-/known-"plaintext" scenario and in the chosen-/known-"ciphertext" one.

In more details, a certain number of rounds $R_F^{stat} = 2 \cdot R_f^{stat}$ with full S-Box layer situated at the beginning and the end guarantee security against statistical attacks. Indeed, even without the middle part, they are sufficient in order to apply the "Wide-Trail" strategy, in a way that we are going to show in the following. Security against all algebraic attacks is achieved working both with rounds $R_F = R_F^{stat} + R'_F \geq R_F^{stat}$ with full S-Box layer and rounds $R_P \geq 0$ with partial S-Box layer. Even if few (even one) S-Boxes per round are potentially sufficient to increase the degree of the encryption/decryption function (which mainly influences the cost of an algebraic attack), other factors can play a crucial role on the cost of such attacks (e.g. a Gröbner basis attack depends also on the number of non-linear equation to solve).

With this in mind, the idea is to construct "something in the middle" between an SPN and a P-SPN cipher. Moreover, since we aim to have the same security w.r.t. chosen-plaintext and chosen-ciphertext attacks, we consider a cipher which is "symmetric": in other words, the same number of rounds with full non-linear layer are applied at the beginning and at the end, where the rounds with partial non-linear layers are in the middle and they are "masked" by the rounds with full non-linear layers. As a result, depending on the cost metric that one aims to minimize (e.g. the total number of non-linear operations) and on the size of the S-Box, in the following we provide the *best ratio* between the number of rounds with full S-Box layer and with partial ones in order to both achieve security and minimize the cost metric.

For more details about Hades strategy, we refer to [30].

What about the choice of the linear and of the non-linear layer? This strategy does not pose any restriction/constriction on the choice of the linear layer and/or on the choice of the S-Box. The idea is to *consider a "traditional" SPN cipher based on the wide trail strategy, and then to replace a certain number of rounds with full S-Box layer with the same number of rounds with partial S-Box layer* in order to minimize the number of non-linear operations, but without affecting the security. The Hades strategy has a huge impact especially in the case of ciphers with low-degree S-Box, since in this case a large number of rounds is required to guarantee security against algebraic attacks.

2.3 The Block Ciphers HadesCubic & HadesInverse and the Permutations Starkad$^\pi$, Persephone$^\pi$ & Poseidon$^\pi$

HadesMiMC is a block cipher constructed using the strategy just proposed, hence it is both an SPN and a Partial-SPN cipher. Roughly speaking, HadesMiMC is obtained by applying the Hades strategy to the cipher Shark [17], proposed by Daemen *et al.* in 1997 and based on the wide trail strategy.

HadesMiMC works with texts of $t \geq 2$ words[6] in $\mathbb{F}_p$ or $\mathbb{F}_{2^n}$, where p is a prime of size $p \approx 2^n$.

As for Shark, the MixLayer of HadesMiMC is simply defined by a multiplication with a fixed $t \times t$ MDS matrix or near-MDS matrix. The number of rounds $R = 2 \cdot R_f + R_P$ depends on the choice of the S-Box and of the parameters n and t. For the applications that we have in mind, we focus on

- the cubic S-Box S-Box$(x) = x^3$ – remember that the cubic S-Box is a bijection in $GF(2^n)$ iff n is odd and it is a bijection in $GF(p)$ iff $p = 2 \mod 3$; in the following, we call this case as HadesCubic;
- the inverse one S-Box$(x) = x^{-1}$; in the following, we call this case as HadesInverse.

In the following:

- *for the prime case* and cubic S-Box, Persephone$^\pi$ is obtained by fixing the key of HadesCubic with a random/chosen value (e.g. zero);
- *for the prime case* and inverse S-Box, Poseidon$^\pi$ is obtained by fixing the key of HadesInverse with a random/chosen value (e.g. zero);
- *for the binary case* and cubic S-Box, Starkad$^\pi$ is obtained by fixing the key of HadesCubic with a random/chosen value (e.g. zero).

About the MDS Matrix. A $t \times t$ MDS matrix[7] with elements in $GF(2^n)$ (or $GF(p)$ where $p \approx 2^n$) exists if the condition (see [27] for details)

$$\log_2(2t + 1) \leq n$$

(or equivalently $t \cdot \log_2(2t + 1) \leq N$) is satisfied.

Given n and t, there are several ways to construct an MDS matrix. One of them is using Cauchy Matrix [35], which we recall here briefly. Let $x_i, y_i \in \mathbb{F}_{2^n}$ for $i = 1, ..., t$ s.t.

- $\forall i \neq j : \qquad x_i \neq x_j, \quad y_i \neq y_j,$
- for $1 \leq i \leq t$ and $1 \leq j \leq t$: $\qquad x_i \oplus y_j \neq 0.$

To fulfill these conditions, one can simply consider x_i s.t. the $t - \log_2(t)$ most significant bits are zero. Then, choosing $r \in \mathbb{F}_{2^n}$ s.t. the $t - \log_2(t)$ most significant bits are non zero, let $y_i = x_i \oplus r$. Let A be the Cauchy matrix defined by

$$a_{i,j} = \frac{1}{x_i \oplus y_j}.$$

[6] The case $t = 1$ corresponds to MiMC [2].

[7] A matrix $M \in \mathbb{F}^{t \times t}$ is called *Maximum Distance Separable* (MDS) matrix iff it has branch number $\mathcal{B}(M)$ equal to $\mathcal{B}(M) = t + 1$. The branch number of M is defined as $\mathcal{B}(M) = \min_{x \in \mathbb{F}^t}\{wt(x) + wt(M(x))\}$, where wt is the hamming weight. Equivalently, a matrix M is MDS iff every submatrix of M is non-singular.

It follows that A is MDS. A similar construction works for $\mathbb{F}_p$.

Efficient Implementation. We refer to App. A for a complete description about possible strategies for efficient HadesCubic implementations.

3 Security Analysis of HadesCubic-Hash

As for any new design, it is paramount to present a concrete security analysis. In the following, we provide an in-depth analysis of the security of HadesCubic-Hash. Due to a lack of any method to ensure that an hash function based on a sponge construction is secure against all possible attacks, we base our argumentation on the following consideration. As we just recalled in the previous section, when the internal permutation $\mathcal{P}$ of an $N = c + r$ bit sponge function is modeled as a randomly chosen permutation, the sponge hash function is indifferentiable from a random oracle up to $2^{c/2}$ calls to $\mathcal{P}$. Thus, we choose *the numbers of rounds of the inner permutation case in order to guarantee security against any (secret-/known-/chosen-) distinguisher which is independent of the key. Equivalently, this means that such number of rounds guarantee that $\mathcal{P}$ does not present any non-random/structural property (among the ones known in the literature*[8]*).*

Since the cryptanalysis of HadesCubic$^\pi$ is close to the one provided in [30] for HadesCubic, we limit ourselves here to do some considerations, and we refer to Supplementary Material for a detailed cryptanalysis of HadesCubic$^\pi$ instantiated by S-Box$(x) = x^3$ in $\mathbb{F}_{2^n}$ or/and $\mathbb{F}_p$. We remark that many attacks are independent of the details of the S-Box. Moreover, for our applications we discuss the security up to $2^M \le 2^N$, that is *we work in a scenario where the data/computational costs of the attacker are limited to* $2^M \le 2^N$.

1st) Remark. Before going on, we remark that the fact that $\mathcal{P}$ presents a non-random/structural property does not imply an attack on the hash sponge function instantiated by $\mathcal{P}$. To have a concrete example, consider Keccak (SHA-3). A zero-sum distinguisher can be set up for the *full* 24-round internal permutation that defines it - see for example [14,18]. In other words, the internal permutation that defines Keccak presents a non-random property, that is it does not look like a randomly-drawn permutation[9]. On the other hands, the best practical collision attack cover ("only") up to 6 rounds Keccak [33], which is still far from threatening the security of the full 24-round Keccak family.

2nd) Remark. Roughly speaking, the fact that a permutation obtained by HadesCubic with a fixed (e.g. known/chosen) key does not present any non-random property (i.e. the fact that it is indistinguishable from a randomly-drawn permutation) corresponds to the impossibility to set up a known-/chosen-key distinguisher on the corresponding permutation HadesCubic with a fixed

[8] We do not exclude that a non-random property can be discovered in the future.

[9] We also refer to [8] for a detailed discussion about this topic.

known/chosen key. In the so-called known-key [24] or chosen-key models, the attacker can have access or even choose the key(s) used, and the goal is to find some input/output pairs having a certain property with a complexity lower than what is expected for randomly chosen permutation(s).

3rd) Remark. The analysis of HadesInverse is analogous of the one provided in the following for HadesCubic. We limit ourselves to emphasize the main difference in App. C.

3.1 Main Points of Our Cryptanalysis Results

Here we would like to list/highlight the main points of our cryptanalysis results (which is given in details in the appendix). The number of rounds we can break depends on the security level M and the number of S-boxes t, which we specify for each concrete hash function instance in the next section.

$\mathbb{F}_p^t$ *versus* $\mathbb{F}_{2^n}^t$. From the point of view of the designer, the prime field version $\mathbb{F}_p^t$ is always stronger than the binary field version $\mathbb{F}_{2^n}^t$, since fewer attacks apply. In particular, the designer must be taken into account the higher-order differential attack when he determines the number of rounds in order to guarantee security in $\mathbb{F}_{2^n}^t$. Vice-versa, this attack does not apply (or better, it is much less powerful) in $\mathbb{F}_p^t$ (due to the fact that the only subspaces of $\mathbb{F}_p$ are $\{0\}$ and the entire space).

Statistical Attacks. As we show in the following, the best statistical attacks (differential, linear, truncated/impossible differential attacks, rebound attack) cover at most 5 rounds with full S-Box layer both for the case S-Box$(x) = x^3$.

Algebraic Attacks. In order to estimate the security against algebraic attacks, we evaluate the degree of the reduced-round permutations and their inverses. At this regards, it is important to make an important clarification: even if all algebraic attacks depend on the degree[10], the number of rounds necessary to protect from this attack is not always the same. This is motivated by the different "relation" between the algebraic attack and the degree.

Roughly speaking, our results can be summarized as following (where $n \simeq \log_2(p)$):

Interpolation Attack. The interpolation attack depends on the number of different monomials of the interpolation polynomial, where (an upper/lower bound of) the number of different monomials can be estimated given the degree of the function. The idea of such attack is to construct an interpolation polynomial that describes the function. If the number of monomials is too big, then such polynomial can not be constructed faster than via a brute force attack. We show

[10] Let us briefly discuss the number of rounds necessary for the permutation output function to be a polynomial of degree D in F_p (or F_{2^n}) over its inputs. The number of rounds needed to achieve degree D for full-round x^3 S-boxes is estimated as $\log_3 D$.

that when the polynomial is dense, the attack complexity is $O(D^t)$, where a certain number of rounds ($\log_2(t)$ in $\mathbb{F}_{2^n}$ and $\log_3(t)$ in $\mathbb{F}_p$) must be added in order to guarantee that the polynomial becomes dense. Therefore for security level M bits $\log_3 2^{\min\{n,M\}} \approx 0.63 \min\{n, M\} + \alpha \cdot \log_2 t$ (for a certain $\log_3(2) \le \alpha \le 1$) rounds can be attacked.

Gröbner Basis. In a Gröbner basis attack, one tries to solve a system of non-linear equations that describe the function. The cost of such attack depend obviously on the degree of the equations, but also on the number of equations and on the number of variables. We show that the attack complexity is about $O(D^{2t})$, therefore for security level M bits the attack works at most on $\log_3 2^{\min\{n/2,M/2\}}$ rounds, which is smaller than for the interpolation attack. If a partial S-Box layer is used, it could become more efficient to consider degree-3 equations for single S-Boxes. In this case, more rounds can be necessary to guarantee security against this attack.

Higher-Order Differential. The higher-order differential attack depends on the *boolean degree*, where the boolean degree δ of a function $f(x) = x^d$ is given by $\delta = hw(d)$ where $hw(\cdot)$ is the hamming weight. The idea of such attack is based on the property that given a function $f(\cdot)$ of boolean degree δ, then $\bigoplus_{x \in V \oplus \phi} f(x) = 0$ if the dimension of the subspace V satisfies $dim(V) \ge \delta + 1$. If the boolean degree is sufficiently high, then the attack does not work. Working with rounds with full S-Box layer, the boolean degree grows at least as 2^{R_F} (where R_F denotes rounds with full S-Box layer) as long as $\delta < N/2$, which for $M < N/2$ yields $\log_2 M$ as the maximum number of rounds to be attacked. For the permutation inverse, the boolean degree grows by the factor $(n + 1)/2$ each round, which exceeds M after 2 rounds as $n/4 > t$ and $M < t \cdot n/2$. The result is similar working with rounds with partial S-Box layer. Similar results hold for the case of rounds with partial S-Box layer. *Zero-Sum Partition.* The zero-sum partition distinguisher can be applied for $q = q_1 + q_2$ rounds as long as the boolean degree in the forward direction for q_1 and in the backward direction for q_2 does not exceed M. This allows attacking $\log_2 M + 2$ rounds.

Security Margin. Given the *minimum* number of rounds necessary to guarantee security against all attacks known in the literature, we *arbitrary* decided by adding:

- two more rounds with full S-Box layer (+2 R_F);
- 7.5% more rounds with partial S-Box layer (+7.5% R_P).

4 Number of Rounds Needed for Security – HadesCubic-Hash

The design goal is to offer an hash function optimized for schemes whose performance critically depends on the MULTdepth/ANDdepth, the number of MULTs/ANDs, or the number of MULTs/ANDs per bit. We thus try to be as close to the number of rounds needed for security as possible.

Besides the possibility to choose the size of the S-Box, one of the strengths of our design is the freedom to choose the ratio between the number of rounds R_F with full S-Box layer and the number of rounds R_P with partial S-Box layer. For the applications that we have in mind, here we limit ourselves to optimize HadesCubic-Hash w.r.t. two different metrics:

- *Number of Multiplications/S-Box;*
- *Number of Multiplications/S-Box* × *Field Size.*

Remark. *In the following, we discuss together the cases $\mathbb{F}_{2^n}$ and $\mathbb{F}_p$, assuming $n \simeq \log_2(p)$.*

Preliminary. HadesCubic results secure if – *for every attack* – (at least) one of the following inequality is satisfied

$$R_F \geq \Phi^F(N, M, t) \quad \text{or} \quad R_P + \varphi(t) \cdot R_F \geq \Phi^P(N, M, t) \quad \text{or} \quad R_F \geq \Phi(N, M, t, R_P)$$

where $\Phi(N, M, t, R_P), \Phi^F(N, M, t), \Phi^P(N, M, t)$ and $\varphi(t)$ are functions that depends on the attack (in our cases, $\varphi(t)$ can only be equal to $\varphi(t) = \alpha$ or $\varphi(t) = \alpha \cdot t$ or $\varphi(t) = \alpha \cdot \log_2(t)$ for some constant $\alpha \neq 0$).

In our design strategy, we always exploit the "Wide-Trail" strategy in order to guarantee security against statistical attacks. In other words, for this class of attacks, we limit to work with rounds with full S-Box layer in order to guarantee security. HadesCubic results secure against statistical attacks if

$$R_F^{stat} \geq \Phi^{stat}(N, M, t) = 6.$$

Thus, given

$$R_F = R_F^{stat} + R'_F \geq R_F^{stat},$$

we are actually looking for *the best ratio between R'_F and R_P that minimizes the total number of S-Boxes.*

4.1 Minimize "Number of S-Boxes"

Here we mainly focus on minimizing the number of S-Boxes, since this is the metric that best describes the cost of the applications that we have in mind. In other words, for given n and t, the goal is to find the best ratio between R_P and R_F that minimizes the total number of S-Boxes, given by

$$\textit{minimum number of S-Boxes } = t \cdot R_F + R_P \tag{1}$$

where $t \geq 2$ and where the number of non-linear operations is proportional to the number of S-Boxes.

Remark. As we just said, due to our design strategy, we limit to use rounds with full S-Box layer in order to guarantee security against statistical attacks. For the particular case of S-Box$(x) = x^3$, it is possible to use rounds with full S-Box layer or rounds with partial S-Box layer in order to guarantee security against Gröbner basis attacks and/or Higher-Order diff. attacks (in $\mathbb{F}_{2^n}$).

In the following, we consider these two cases separately. Moreover, we assume N, M and t (and so, n or $\log_2(p)$) fixed. For every concrete instantiations, the idea is to choose the recommendation that minimizes the metric cost (in this case, the number of S-Boxes). *As supplementary material, we provide a script that given in input N, M and t, returns the best ratio between R_P and R_F that minimizes the total number of S-Boxes.*

Finally, let us briefly discuss the case in which t is not fixed in advance (assume just N and M are fixed in advance). In this case, *for each possible value of t* where $2 \leq t \leq \frac{N}{\log_2\{[(2N)/(\log_2(N+1)+1)]+1\}}$ (where this upper bound guarantees the existence of an MDS matrix), one finds the best ratio between R_P and R_F that minimize the number of S-Boxes for that particular t, using the strategy just proposed. Then, one simply looks for the best value of t that minimizes the total number of non-linear operations. Also for this case, *as supplementary material, we provide a script that given in input N, returns the best t and the best ratio between R_P and R_F that minimizes the total number of S-Boxes.*

1*st*) Recommendation. First of all, we consider the case in which the security against Gröbner basis attack is provided by rounds with full S-Box layer. For t and n fixed, let

$$R_F \geq \max\{R_F^{stat}(N,M,t) \equiv 6, R_F^{ZS}(M), R_F^{Grob}(N,M,t)\}$$

where R_F^{stat} is the minimum number of rounds with full S-Box layer necessary to prevent statistical attacks, R_F^{ZS} is the minimum number of rounds with full S-Box layer necessary to prevent zero-sum attacks, while R_F^{Grob} is the minimum (even) number of rounds necessary to prevent Gröbner basis attacks (they both depend on t and N). In order to be secure, HadesCubic must satisfy another condition related to the interpolation attack:

$$R_P + R_F \geq R^{inter}(N,t). \tag{2}$$

Note that every R_P and R_F that satisfy the previous inequalities guarantee security. Obviously, in order to minimize the number of S-Boxes, it makes sense to choose R_P and R_F for which the previous inequality is minimized, that is $R_P + R_F = R^{inter}(N,t)$.

What is the best ratio between R_P and R_F that minimizes the total number of S-Boxes? By combining eq. (3) (that is, the minimum number of S-Boxes) and eq. (2), we get

$$t \cdot R_F + R_P \bigg|_{R_P+R_F\geq R^{inter}} \leq R_F(t-1) + R^{inter}$$

which is minimized by taking the minimum value of R_F (where note that R^{inter} is fixed for t and N fixed).

Assuming $M \leq N/2$ and $n \simeq \log 2(p)$, it follows that the best choice is given by

$$\begin{aligned} R_F &= \max\bigl\{R_F^{stat} \equiv 6, R_F^{ZS}(M), R_F^{Grob}(N,M,t)\bigr\} \\ R_P &= \max\bigl\{0; \underbrace{1 + \lceil \log_3(2) \cdot \min\{n; M\} \rceil + \Phi(t)}_{\equiv R^{inter}(N,M,t)} - R_F\bigr\}. \end{aligned}$$

where $N \approx t \cdot \log_2 p$ and where[11]

$$\begin{aligned} R_F^{Grob}(N,M,t) &= 2 \cdot \left\lceil \frac{1}{2} + \frac{\log_3(2)}{2} \cdot \left(\frac{\min\{n; M\}}{2} + \log_2(t) \right) \right\rceil \\ R_F^{ZS}(M) &= \begin{cases} 2 + 2 \cdot \left\lceil \frac{1+\log_2(M)}{2} \right\rceil & \text{working over } \mathbb{F}_{2^n} \\ 0 & \text{working over } \mathbb{F}_p \end{cases} \\ \Phi(t) &= \begin{cases} \lceil \log_2(t) \rceil & \text{working over } \mathbb{F}_{2^n} \\ \lceil \log_3(t) \rceil & \text{working over } \mathbb{F}_p \end{cases} \end{aligned}$$

***2nd*) Recommendation.** The second possibility would be to guarantee security against Gröbner basis attack by using rounds with partial S-Box layer. For this particular case, HadesCubic results secure if R_F and R_P satisfy (at least) one of the two following system of inequalities:

$$\begin{cases} R_F \geq \max\bigl\{R_F^{stat} \equiv 6; R_F^{ZS}(M)\bigr\} \qquad \text{and} \qquad R_P \geq 0; \\ R_P + R_F \geq \Psi^{(1)}(N,M,t) \equiv \max\bigl\{R^{inter}(N,M,t); R^{1st-Grob}(N,M,t)\bigr\} \\ R_P + t \cdot R_F \geq \Psi^{(t)}(N,M,t) \equiv R^{2nd-Grob}(N,M,t) \\ R_F \geq R^{3rd-Grob}(N,M,t,R_P) \end{cases}$$

or

$$\begin{cases} R_F \geq \max R_F^{stat} \equiv 6 \qquad \text{and} \qquad R_P \geq 0; \\ R_P + R_F \geq \Psi^{(1)}(N,M,t) \equiv \max\bigl\{R^{inter}(N,M,t); R^{1st-Grob}(N,M,t); R_P^{ZS}(N,M,t)\bigr\} \\ R_P + t \cdot R_F \geq \Psi^{(t)}(N,M,t) \equiv R^{2nd-Grob}(N,M,t) \\ R_F \geq R^{3rd-Grob}(N,M,t,R_P) \end{cases}$$

[11] *Remark:* for the case "HadesCubic-Hash instantiated over $\mathbb{F}_p$", we set $R_F^{ZS}(M) = 0$ since the number of rounds necessary to guarantee security against the interpolation attack are sufficient to prevent higher-order differential attacks as well.

where $R^{inter}(N,M,t)$ is defined as before, and where $R_P^{ZS}(N,M,t)$, $R^{1st-Grob}(N,M,t)$, $R^{2nd-Grob}(N,M,t)$ and $R^{3rd-Grob}(N,M,t,R_P)$ are defined as

$$R_P^{ZS}(N,M,t) = \begin{cases} 5 + \lceil \log_2(n-1) \rceil + \left\lceil \frac{3 \cdot M}{n-1} \right\rceil & \text{working over } \mathbb{F}_{2^n} \\ 0 & \text{working over } \mathbb{F}_p \end{cases}$$

$$R^{1st-Grob}(N,M,t) = 1 + \left\lceil \frac{\min\{n; M\} + 2 \cdot \log_2(t)}{2 \cdot \log_2(3)} \right\rceil$$

$$R^{2nd-Grob}(N,M,t) = 1 + \left\lceil \frac{M}{2 \cdot (\log_2(27) - 2)} \right\rceil$$

$$R^{3rd-Grob}(N,M,t,R_P) = 1 + \log_3(2) \cdot \left(\frac{M}{2t + R_P} + 2 \cdot \log_2(2t + R_P) - 2 \cdot \log_2(2t) \right)$$

where $n \simeq \log_2(p)$. Here $R_P^{ZS}(N,M,t)$ is the minimum number of rounds with partial S-Box layer necessary to guarantee security against higher-order differential attack, while $R^{1st-Grob}(N,M,t)$, $R^{2nd-Grob}(N,M,t)$ and $R^{3rd-Grob}(N,M,t,R_P)$ are all conditions necessary to prevent Gröbner basis attacks (each one of them guarantees security against a particular implementation of Gröbner basis attack).

As before, the goal is to find the best ratio between R'_F (where $R_F = R_F^{stat} + R'_F \geq R_F^{stat}$) and R_P that minimizes the total number of S-Boxes, where both $\Psi^{(1)}(N,M,t)$ and $\Psi^{(t)}(N,M,t)$ are fixed (since N, M and t are fixed).

Results via Script. A complete analysis on how to set up the script – in order to guarantee security and to find the best ratio between R_P and R_F – for this case has been proposed in [30]. For this reason, we refer to [30], and we limit ourselves here to report the minimum number of rounds necessary to guarantee security.

For completeness, we mention that the simplest way to set up the script is to test (e.g. by brute force) all possible values R_P and R_F that guarantee security (equivalently, for which previous inequalities are satisfied), and finds the ones that minimize the (metric) cost.

4.2 Minimize "Number of S-Boxes × Field Size"

Secondly, we consider the metric given by "number of S-Boxes × field size", which well describes the cost of the Picnic PQ-Signature Scheme – where HadesCubic instantiated over $\mathbb{F}_{2^n}$ (low-data case). In this case, for each N and t, the goal is to find the best ratio of R_P and R'_F (where $R_F = R_F^{stat} + R'_F \geq R_F$) for which the following cost is minimized

$$n \times \big(t \cdot R_F + R_P\big) = N \cdot R_F + n \cdot R_P. \tag{3}$$

If both n and t are fixed, this metric is proportional to the one given before (that is, it is equal to the one given in (3) times a factor n). Thus, the results given in the previous section hold also for this metric. As before, for the case in which t is not fixed, we provide *a script that takes in input N and returns the best t and the best ratio between R_F and R_P that minimizes the metric given in (3).*

5 Concrete Instantiations – PERSEPHONE$^{\pi}$ and STARKAD$^{\pi}$

For our applications, we are interested in the cases:

- texts size: $N = 1\,536 = 3 \cdot 2^9$ (where $N = n \cdot t \simeq t \cdot \log_2 p$);
- security level: $M = 128$ and/or 256.

All our MDS matrices are Cauchy matrices, and the method to construct them is further described in Section 2.3. We use ascending sequences of integers (or elements in $\mathbb{F}_{2^n}$) for the construction.

The round constants are generated using the Grain LFSR [?] in a self-shrinking mode:

- Initialize the state with 80 bits $b_0, b_1, \ldots, b_{79}$ set to 1.
- Update the bits using $b_{i+80} = b_{i+62} \oplus b_{i+51} \oplus b_{i+38} \oplus b_{i+23} \oplus b_{i+13} \oplus b_i$.
- Discard the first 160 bits.
- Evaluate bits in pairs: If the first bit is a 1, output the second bit. If it is a 0, discard the second bit.

If a randomly sampled integer is not in $\mathbb{F}_p$, we discard this value and take the next one. Note that cryptographically strong randomness is not needed for the round constants, and other methods can also be used. We give both the matrices and the round constants in an auxilliary file for two example instantiations:

- PERSEPHONE-Permutation in $\mathbb{F}_p$ with $p = 2^{64} - 2^8 - 1$, $n = 64$, $t = 24$, $N = 1536$,
- POSEIDON-Permutation in $\mathbb{F}_p$ with ???,
- STARKAD-Permutation in $\mathbb{F}_{2^n}$ with $p(x) = x^{63} + x + 1$, $n = 63$, $t = 25$, $N = 1575$.

We also include a reference implementation for the first of these two instantiations.

Table 1: A range of different parameter sets for Starkad$^\pi$ and Persephone$^\pi$ instantiated by S-Box$(x) = x^3$, where the security level is $M = 256$. check formulas/script, some results are strange here...

Security	**Text Size**	**S-Box Size**	**# S-Boxes**	**$\mathbf{R_F}$**	**$\mathbf{R_P}$**	**Field**	**Cost 1**	**Cost 2**
M	$N = n \times t$	(n or $\log_2 p$)	(t)				Sect. 4.1	Sect. 4.2
128	1536	768	2	10	73	$\mathbb{F}_p$	93	71424
128	1536	384	4	8	76	$\mathbb{F}_p$	108	41472
128	1536	256	6	8	76	$\mathbb{F}_p$	124	31744
128	1536	192	8	8	76	$\mathbb{F}_p$	140	26880
128	1536	96	16	6	59	$\mathbb{F}_p$	155	14880
128	1551	33	47	6	22	$\mathbb{F}_{2^n}$	304	10032
128	1575	63	25	6	40	$\mathbb{F}_{2^n}$	190	11970
128	1581	31	51	6	21	$\mathbb{F}_{2^n}$	327	10137
256	1536	768	2	10	154	$\mathbb{F}_p$	174	133632
256	1536	384	4	10	155	$\mathbb{F}_p$	195	74880
256	1536	256	6	8	157	$\mathbb{F}_p$	205	52480
256	1536	192	8	8	117	$\mathbb{F}_p$	181	34752
256	1536	96	16	6	59	$\mathbb{F}_p$	155	14880
256	1551	33	47	6	28	$\mathbb{F}_{2^n}$	310	10230
256	1575	63	25	6	40	$\mathbb{F}_{2^n}$	190	11970
256	1581	31	51	6	30	$\mathbb{F}_{2^n}$	336	10416

6 SNARKs Application via Persephone$^\pi$ and Poseidon$^\pi$

ZK-SNARKs and Bulletproofs are powerful proof systems to prove the computational integrity of very complex program executions. They are helpful in cryptographic protocols where Prover proves the knowledge of a hash function preimage or an opening in a Merkle tree. Such protocols are popular in cryptocurrencies where they make possible to hide the transaction origin or amount by only proving it had been earlier included to a Merkle tree. Both SNARKs and Bulletproofs work with programs represented as arithmetic circuits over some prime field $GF(p)$. In SNARKs , the prime field is typically the scalar field of some point on a pairing-friendly elliptic curve, whereas in Bulletproofs the curve does not have to be pairing-friendly (thus fast curves such as Curve25519 are a popular choice). The primitive Persephone$^\pi$ can be represented as such circuit with reasonably few gates, but the parameters of Persephone$^\pi$ must have been determined first by p. Concretely, after p is fixed, we first check if x^3 or x^5 are bijections in $GF(p)$, which is true if $p \bmod 3 \neq 1$ (resp., $p \bmod 5 \neq 1$). If both inequalities are not satisfied, we have to use the inverse S-box and thus the Poseidon design.

The SNARK prover complexity is $O(s)$ where s is the number of rank-1 constraints – quadratic equations of form $(\sum_i u_i X_i)(\sum_i v_i X_i) = \sum_i w_i X_i$ where

u_i, v_i, w_i are field elements and X_i are program variables. It is easy to see that the S-box x^3 is represented by 2 constraints, the S-box x^5 by 3 constraints, and the S-box $1/x$ by 3 constraints (1 for non-zero case, and two more for the zero case). Thus in total we have

$$2tR_F + 2R_P \text{ constraints for } x^3\text{-based } \textsc{Persephone}^\pi; \tag{4}$$

$$3tR_F + 3R_P \text{ constraints for } x^5\text{-based } \textsc{Persephone}^\pi; \tag{5}$$

$$3tR_F + 3R_P \text{ constraints for } 1/x\text{-based } \textsc{Poseidon}^\pi. \tag{6}$$

It requires a bit more effort to see that we do not need more constraints as the linear layers and round constants can be incorporated into these ones. However, it is necessary to do some preprocessing. For example, in the $\textsc{Persephone}^\pi$ setting the full S-Box layers are followed by linear transformation $M = (M_{i,j})$. Each round with full S-Box can be represented by the following constraints in the SNARK setting.

$$\left(\sum_j M_{i,j}x_{i,j}\right) \cdot \left(\sum_j M_{i,j}x_{i,j}\right) = y_i \;\; 1 \leq i \leq t \tag{7}$$

$$y_i \cdot \left(\sum_j M_{i,j}z_{i,j}\right) = z_i \tag{8}$$

where $M = I_{t\times t}$ for the first round. However, in a round with partial S-Box layer we will have only one such constraint for $j = 1$. For the rest of the $t-1$ variables we will have linear constraints of the form

$$\sum_j M_{i,j}x_{i,j} = u_i \;\; \text{where} 2 \leq i \leq t.$$

Since the linear constrains have no role in the SNARK, in the following partial S-Box rounds the linear constraints can be composed with (from the previous round(s)) using following equation

$$\sum_k M_{i,k}\left(\sum_j M_{i,j}x_{i,j}\right) = v_k \;\; 2 \leq k \leq t$$

We can now calculate the number of constraints for the sponge-based hash functions and Merkle trees. In sponges, the $2M$ bits are reserved for the capacity, so $N - 2M$ bits are fed with message. Therefore, we get

- $\frac{2tR_F+2R_P}{N-2M}$ constraints per bit for x^3-based $\textsc{Persephone}^\pi$;
- $\frac{3tR_F+3R_P}{N-2M}$ constraints per bit for x^5-based $\textsc{Persephone}^\pi$;
- $\frac{3tR_F+3R_P}{N-2M}$ constraints per bit for $1/x$-based $\textsc{Poseidon}^\pi$.

Similarly we obtain that the Merkle tree based on such a sponge function has branching $\frac{N}{2M}-1$. Based on that we can calculate how many constraints we need to prove the opening in a Merkle tree of, for example, 2^{32} elements (the recent ZCash setting). The tree will have $32\log_{-1+N/2M} 2$ levels with the number of constraints in each according to the above.

7 STARKs Application via Starkad$^\pi$

For STARKs, we focus on the case of GF(2^n), with $n \approx 64$, since currently available high-performance implementation of STARKs also focus on the binary field case. However, the authors of [**?**] explicitly mention that STARKs also work over prime fields. We chose the binary field size to be close to 64 bits to be able to efficiently utilize the carry-less multiplication (CLMUL) instruction-set available in recent CPUs to speed up finite field operations.

The main costs metrics for creating a STARK are dependant on the AET (Algebraic Execution Trace) and the associated constraint system of the computation to be proven. For the concrete instance proposed for the 128-bit security level, the constraint system consists of 240 degree-3 constraint polynomials, ensuring the integrity of the S-Box computations.

should we add any concrete complexity estimates based on the equations from the short STARK overview paper? For that we would need to build an approximate AET.

References

1. Abdelraheem, M.A., Ågren, M., Beelen, P., Leander, G.: On the Distribution of Linear Biases: Three Instructive Examples. In: CRYPTO 2012. LNCS, vol. 7417, pp. 50–67 (2012)
2. Albrecht, M.R., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: ASIACRYPT 2016. LNCS, vol. 10031, pp. 191–219 (2016)
3. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454 (2015)
4. Aumasson, J.P., Meier, W.: Zero-sum distinguishers for reduced Keccak-f and for the core functions of Luffa and Hamsi (2009), presented at the Rump Session of CHES 2009, https://131002.net/data/papers/AM09.pdf
5. Bardet, M., Faugere, J., Salvy, B., Yang, B.: Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In: The Effective Methods in Algebraic Geometry Conference (MEGA). pp. 1–14 (2005)
6. Beierle, C., Canteaut, A., Leander, G., Rotella, Y.: Proving Resistance Against Invariant Attacks: How to Choose the Round Constants. In: CRYPTO 2017. LNCS, vol. 10402, pp. 647–678 (2017)
7. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: On the Indifferentiability of the Sponge Construction. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197 (2008)
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Note on zero-sum distinguishers of Keccak-f, http://keccak.noekeon.org/NoteZeroSum.pdf
9. Bettale, L., Faugere, J.C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology **3**(3), 177–197 (2009)
10. Bettale, L., Faugère, J., Perret, L.: Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: International Symposium on Symbolic and Algebraic Computation - ISSAC 2012. pp. 67–74. ACM (2012)
11. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology **4**(1), 3–72 (1991)
12. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer (1993)
13. Boura, C., Canteaut, A.: On the Influence of the Algebraic Degree of F^{-1} on the Algebraic Degree of $\mathrm{G} \circ \mathrm{F}$. IEEE Trans. Information Theory **59**(1), 691–702 (2013)
14. Boura, C., Canteaut, A., De Cannière, C.: Higher-Order Differential Properties of Keccak and *Luffa*. In: FSE 2011. LNCS, vol. 6733, pp. 252–269 (2011)
15. Carlet, C., Charpin, P., Zinoviev, V.A.: Codes, Bent Functions and Permutations Suitable For DES-like Cryptosystems. Designs, Codes Cryptography **15**(2), 125–156 (1998)
16. Cox, D.A., Little, J., O'Shea, D.: Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.). Undergraduate texts in mathematics, Springer (1997)
17. Daemen, J., Knudsen, L.R., Rijmen, V.: The Block Cipher Square. In: FSE 1997. LNCS, vol. 1267, pp. 149–165 (1997)
18. Duan, M., Lai, X.: Improved zero-sum distinguisher for full round Keccak-f permutation. Chinese Science Bulletin **57**(6), 694–697 (2012)
19. Grassi, L.: Mixture Differential Cryptanalysis: a New Approach to Distinguishers and Attacks on round-reduced AES. IACR Trans. Symmetric Cryptol. **2018**(2), 133–160 (2018)

20. Grassi, L., Rechberger, C., Rønjom, S.: A New Structural-Differential Property of 5-Round AES. In: EUROCRYPT 2017. LNCS, vol. 10211, pp. 289–317 (2017)
21. Jakobsen, T., Knudsen, L.R.: The Interpolation Attack on Block Ciphers. In: FSE 1997. LNCS, vol. 1267, pp. 28–40 (1997)
22. Jean, J., Naya-Plasencia, M., Peyrin, T.: Multiple Limited-Birthday Distinguishers and Applications. In: SAC 2013. LNCS, vol. 8282, pp. 533–550 (2013)
23. Knudsen, L.R.: Truncated and Higher Order Differentials. In: FSE 1994. LNCS, vol. 1008, pp. 196–211 (1994)
24. Knudsen, L.R., Rijmen, V.: Known-Key Distinguishers for Some Block Ciphers. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324 (2007)
25. Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schläffer, M.: Rebound Distinguishers: Results on the Full Whirlpool Compression Function. In: ASIACRYPT 2009. LNCS, vol. 5912, pp. 126–143 (2009)
26. Leander, G., Abdelraheem, M.A., AlKhzaimi, H., Zenner, E.: A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In: CRYPTO 2011. LNCS, vol. 6841, pp. 206–221 (2011)
27. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. Northholland Publishing Company (1978)
28. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397 (1993)
29. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In: FSE 2009. LNCS, vol. 5665, pp. 260–276 (2009)
30. *No Authors Given*: The HADES Design Strategy and Instantiations, in Submission
31. Nyberg, K.: Differentially uniform mappings for cryptography. In: EUROCRYPT 1993. LNCS, vol. 765, pp. 55–64 (1994)
32. Nyberg, K., Knudsen, L.R.: Provable Security Against Differential Cryptanalysis. In: CRYPTO 1992. LNCS, vol. 740, pp. 566–574 (1992)
33. Qiao, K., Song, L., Liu, M., Guo, J.: New Collision Attacks on Round-Reduced Keccak. In: Advances in Cryptology – EUROCRYPT 2017. LNCS, vol. 10212, pp. 216–243 (2017)
34. Todo, Y.: Structural Evaluation by Generalized Integral Property. In: EUROCRYPT 2015. LNCS, vol. 9056, pp. 287–314 (2015)
35. Youssef, A.M., Mister, S., Tavares, S.E.: On the Design of Linear Transformations for Substitution Permutation Encryption Networks. In: School of Computer Science, Carleton University. pp. 40–48 (1997)

SUPPLEMENTARY MATERIAL

A Efficient Implementation

Like for LowMC, the fact that the non-linear layer is partial in R_P rounds can be used to reduce the size of the round constants required in each round R_P. Referring to [?], we recall here an equivalent representation of an SPN with partial non-linear layer for an efficient implementation.

Round Constants. In the description of an SPN, it is possible to swap the order of the linear layer and the round constant addition as both operations are linear. The round constant then needs to be exchanged with an equivalent one. For round constant $c^{(i)}$, the equivalent one can be written as $\hat{c}^{(i)} = MC^{-1}(c^{(i)})$, where MC is the linear layer in the i-th round. If one works with partial non-linear layers, it is possible to use this property to move parts of the original round constants from the last round all the way through the permutation to the whitening key. To arrive at such a reduced variant, we work as following:

- First, we find an equivalent round constants that is applied before the affine layer by moving the round key through the affine layer.
- Then we split the round constants in two parts, one that applies to the S-Box part of the non-linear layer and one that applies to the identity part of the non-linear layer. The constant part that only applies to the non-linear layer part can now move further up where it is merged with the previous round key.
- Working in this way for all round constants, we finally end up with an equivalent representation in which round constants are only added to the output of the S-Boxes apart from one constant which is applied to the entire state after the first R_f rounds.

This simplified representation can in certain cases also reduce the implementation cost of an SPN permutation with a partial non-linear layer. For instance, the standard representation of HadesCubic requires constants matrices of total size $t \cdot n \cdot (R + 1)$, where $R = R_P + R_F$ is the number of rounds. The optimized representation only requires $t \cdot n \cdot (R_F + 1) + n \cdot R_P$, thus potentially greatly reducing the amount of needed memory and calculation to produce the round constants.

Linear Layer. For our design the situation is simpler than for LowMC, since we can guarantee the existence of invertible sub matrices. Hence, a similar trick can be used also for the matrix multiplication.

Focusing on the rounds with a single S-Box, let M be the $t \times t$ MDS matrix of the linear layer:

$$M = \left[\begin{array}{c|ccccc} M_{0,0} & M_{0,1} & M_{0,2} & \cdots & M_{0,t-1} & M_{0,t} \\ \hline M_{1,0} & & & & & \\ M_{2,0} & & & & & \\ \vdots & & & \hat{M} & & \\ M_{t-1,0} & & & & & \\ M_{t,0} & & & & & \end{array}\right] \equiv \left[\begin{array}{c|c} M_{0,0} & v \\ \hline w & \hat{M} \end{array}\right]$$

where $\hat{M}$ is a $(t-1) \times (t-1)$ MDS matrix (note that since M is MDS, every submatrix of M is also MDS), v is a $1 \times (t-1)$ matrix and w is a $(t-1) \times 1$ vector. By simple computation, the following equivalence holds:

$$M = \underbrace{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{M} \end{array}\right]}_{M'} \times \underbrace{\left[\begin{array}{c|c} M_{0,0} & v \\ \hline \hat{w} & I \end{array}\right]}_{M''}, \qquad (9)$$

where

$$\hat{w} = \hat{M}^{-1} \times w$$

and I is the $(t-1) \times (t-1)$ identity matrix. Note that both M' and $M^{''}$ are two invertible matrices[12].

As for the round constants discussed previously, it is possible to use the equivalence (9) in order *to swap the S-Box layer (formed by a single S-Box and $t-1$ identity functions) and the matrix multiplication with the matrix M'*. As a result, each linear part in the R_P rounds is defined only by a multiplication with a matrix of the form $M^{''}$, which is a *sparse matrix*, since $(t-1)^2 - (t-1) = t^2 - 3t + 2$ coefficients of $M^{''}$ are equal to zero (moreover, $t-1$ coefficients of $M^{''}$ are equal to one). It follows that this optimized representation – potentially – greatly reduces the amount of needed memory and calculation to compute the linear layer multiplication.

B Security Analysis – HadesCubic with S-Box(x) $= x^3$ in $GF(2^n)$

B.1 Security Analysis - Statistical Attacks

Differential Cryptanalysis. Differential cryptanalysis [11,12] and its variations are the most widely used techniques to analyze symmetric-key primitives. The differential probability of any function over the finite field $\mathbb{F}_{2^n}$ is defined as

$$Prob[\alpha \to \beta] := |\{x : f(x) \oplus f(x \oplus \alpha) = \beta\}|/(2^n).$$

[12] First of all, $\det(M') = \det(\hat{M}) \neq 0$ since $\hat{M}$ is an MDS matrix, and so it is invertible. Secondly, $\det(M) = \det(M') \cdot \det(M^{''})$. Since $\det(M) \neq 0$ and $\det(M') \neq 0$, it follows that $\det(M^{''}) \neq 0$.

Since the cubic function $f(x) = x^3$ is an almost perfect non-linear permutation (APN) [32,31], it has an optimal differential probability over a prime field or $\mathbb{F}_{2^n}$ (where n is odd). In other words, for this function the probability is bounded above by $2/2^n$ or $2/|\mathbb{F}_p|$.

As largely done in the literature, we claim that HadesCubic is secure against differential cryptanalysis if each characteristic has probability at most 2^{-N}.

In order to compute the minimum number of rounds to guarantee this, we work only with the rounds with full S-Box layers. In other words, we limit ourselves to work with a "weaker" version of the permutation defined as

$$R^{R_f} \circ L \circ R^{R_f}(\cdot), \tag{10}$$

where

- L is an *invertible linear layer* (which is the "*weakest*" possible assumption),
- $R(\cdot) = M \circ \text{S-Box} \circ ARK(\cdot)$ where S-Box$(\cdot)$ is a full S-Box layer (remember that M is an MDS matrix).

We are going to show that this "weaker" permutation is secure against differential cryptanalysis for $R_F = 2R_f = 6$ if $t+2 < 2n$, and $R_F = 8$ otherwise. As a result, it follows that also HadesCubic (instantiated with R_F rounds with full S-Box layers) is secure against such an attack. Indeed, if the linear layer L (which we only assume to be invertible) is replaced by R_P rounds of HadesCubic, its security cannot decrease. *The same strategy is exploited in the following in order to prove security against all attacks in this subsection.*

In order to prove the result just given, we need a lower bound on the number of minimum number of active S-Boxes. Observe that the minimum number of "active" S-Boxes in the permutation

$$R^s \circ L \circ R^r(\cdot) \equiv SB \circ \underbrace{M \circ SB}_{s-1 \text{ times}} \circ \underbrace{L'}_{\equiv L \circ M(\cdot)} \circ SB \circ \underbrace{M \circ SB}_{r-1 \text{ times}}(\cdot)$$

(where $s, r \geq 1$, $R(\cdot)$ is a round with full S-Box layer and where L' is an invertible linear layer) are at least[13]

$$\text{number } \textit{active} \text{ S-Boxes} \geq \underbrace{\big(\lfloor s/2 \rfloor + \lfloor r/2 \rfloor\big) \cdot (t+1)}_{\text{due to final/initial rounds}} + \big(s \bmod 2\big) + \big(r \bmod 2\big).$$

We emphasize that the (middle) linear $L'(\cdot) \equiv L \circ M(\cdot)$ plays *no* role in the computation of the previous number. Since at least $2 \cdot (t+1) + 1$ S-Boxes are active in the 5 middle rounds of $R^r \circ L \circ R^{5-r}(\cdot)$ for $1 \leq r \leq 4$, and since the maximum differential probability of the cubic S-Box is $DP_{max} = 2^{-n+1}$, each characteristic has probability at most

$$(2^{-n+1})^{2\cdot(t+1)+1} = 2^{-N} \cdot 2^{-N-3n+2t+3} < 2^{-N},$$

[13] If $s = 2 \cdot s'$ is even, then the minimum number of active S-Boxes over $R^s(\cdot)$ rounds with full S-Box layer is $\lfloor s/2 \rfloor \cdot (t+1)$. Instead, if $s = 2 \cdot s' + 1$ is odd, then the minimum number of active S-Boxes over $R^s(\cdot)$ rounds with full S-Box layer is $\lfloor s/2 \rfloor \cdot (t+1) + 1$.

since $[N + 3n = n \cdot (t + 3)] > [2t + 3 = 2 \cdot (t + 3/2)]$, where $t + 3 > t + 3/2$ and $n \geq 3$. Finally, 1 more round guarantees that no differential attack can be set up.

Similarly, in the case in which $t + 2 < 2n$, it is sufficient to consider the 3 middle rounds to guarantee security against differential cryptanalysis. Indeed, each characteristic has probability $(2^{-n+1})^{t+2} = 2^{-N} \cdot 2^{-2n+t+2} < 2^{-N}$, since at least $t+2$ S-Boxes are active. Again, 1 more round guarantees that no differential attack can be set up.

Security up to $2^M \leq 2^N$. For completeness, we present the number of rounds necessary to provide security up to 2^M (that is, data and computational cost of the attacker upper bounded by 2^M). Using the same analysis as before, it turns out that

$$R_F = \begin{cases} 4 & \text{if } t + 2 < N + 2n - M \\ 6 & \text{if } t + 2 \geq N + 2n - M \end{cases}$$

guarantees that no differential attack can be set up.

Linear Cryptanalysis. Similar to differential attacks, linear attacks [28] pose no threat to the HadesCubic family of permutations instantiated with the same number of rounds previously defined for classical differential cryptanalysis. This follows from the fact that the cubic function is almost bent (AB), which means that its maximum square correlation is limited to 2^{-n+1} (see [1] for details). As a result, it offers the best possible resistance against linear cryptanalysis much like an APN function provides optimal resistance against differential cryptanalysis.

For completeness, we remember a function $f(\cdot)$ is AB and/or APN if and only if its inverse $f^{-1}(\cdot)$ is AB and/or APN [15]. As a result, both the forward and the inverse permutation are secure against linear and differential cryptanalysis[14].

Truncated Differential. A variant of classical differential cryptanalysis is the truncated differential one [23], in which the attacker can specify only part of the difference between pairs of texts.

We consider the "weaker" permutation described in (10) again. Focusing only on active/passive bytes (and not on the actual differences), there exist several differentials with probability 1 for a maximum of 1 round of HadesCubic, e.g.

$$[\alpha, 0, ..., 0]^T \xrightarrow{R(\cdot)} M \times [\beta, 0, ..., 0]^T$$

where α, β denote non-zero differences. Due to the next S-Box layer, the linear relations given by $M \times (\beta, 0, ..., 0)^T$ are destroyed in the next round. As a result, no probability-one truncated differential covers more than a single round.

[14] Remember that if a matrix M is MDS, then also M^{-1} is MDS.

Since no linear relation survives the S-Box layer, it seems hard to set up a truncated differential which is independent of the secret key for more than 2 rounds. As a result, it turns out that 4 rounds with full S-Box layer makes HadesCubic$^\pi$ secure against this attack.

Rebound Attacks. The rebound attacks [25,29] have much improved the best known attacks on many hash functions, especially for AES-based schemes. The goal of this attack is to find two (input, output) pairs (p^1, c^1) and (p^2, c^2) such that the two inputs satisfy a certain (truncated) input difference and the corresponding outputs satisfy a certain (truncated) output difference.

The rebound attack consists of two phases, called *inbound* and *outbound* phase. According to these phases, the internal permutation of the hash function is split into three sub-parts. Let f be the permutation, then we get $f = f_{fw} \circ f_{in} \circ f_{bw}$. The part of the inbound phase is placed in the middle of the permutation and the two parts of the outbound phase are placed next to the inbound part. In the outbound phase, two high-probability (truncated) differential trails are constructed, which are then connected in the inbound phase. Since the rebound attack is a differential attack, as first thing an attacker needs to construct a *"good" (truncated) differential trail*. A good trail used for a rebound attack should have a high probability in the outbound phases and can have a rather low probability in the inbound phase. In the first phase, the attacker uses the knowledge of the key to find pairs of texts that satisfy the middle rounds of the truncated differential trail. In the second one, they propagate the solutions found in the first phase in the forward and in the backward directions, and check if at least one of them satisfies the entire differential trail.

The best rebound attack on AES proposed in [22] covers 8 rounds. Here we claim that 6 rounds with full S-Box layers are sufficient to protect HadesCubic$^\pi$ from this attack. To support it, note that *(1st)* 1 round of HadesCubic provides full diffusion while 2 rounds of AES are necessary to provide it and *(2nd)* the best truncated differential covers 1 round of HadesCubic$^\pi$ *vs* 3 rounds of AES[15]. Since the best results on AES in the literature cover at most 8 rounds, due to the similarity between AES and HadesCubic$^\pi$ and due to the previous observations, we argue that it is not possible to mount a rebound attack on more than 5 rounds with full S-Box layers of HadesCubic$^\pi$. Hence, 6 rounds of HadesCubic$^\pi$ with full S-Box layers are sufficient to guarantee security against this attack.

Multiple-of-n and Mixed Differential Cryptanalysis. The "Multiple-of-8" distinguisher [20] was proposed at Eurocrypt 2017 by Grassi *et al.* as the first 5-round secret-key distinguisher for AES that exploits a property which is independent of the secret key and of the details of the S-Box. It is based on a new structural property for up to 5 rounds of AES: by appropriate choices of a

[15] The best truncated differential distinguisher with prob. 1 covers 2 rounds of AES.

number of input pairs it is possible to make sure that the number of times that the difference of the resulting output pairs lie in a particular subspace is always a multiple of 8. The input pairs of texts that satisfy a certain output difference are related by linear/differential relations. Such relations are exploited by a variant of such a distinguisher, called the "mixture differential" distinguisher [19] proposed at FSE/ToSC 2019.

Regarding HadesCubic, it is possible to set up such distinguishers on 2 rounds only. In particular, consider a set of texts with $2 \le s \le t$ active words (and $t-s$ constants words). The number of pairs of texts that satisfy an (arbitrary) output truncated differential is always a multiple of 2^{s-1}. Moreover, the relations of the input pairs of texts exploited by mixture differential cryptanalysis are known.

The proofs of these two properties are analogous to the ones proposed in [20] and in [19]. E.g., consider two texts T^1 and T^2 of the form

$$T^1 = C \oplus \left[x_0 \; x_1 \; 0 \; ... \; 0\right]^T, \qquad T^2 = C \oplus \left[y_0 \; y_1 \; 0 \; ... \; 0\right]^T$$

for some constant C and where $x_i \neq y_i$ for $i = 0, 1$. After one round, the difference in each word is of the form

$$M_0 \cdot [\text{S-Box}(x_0 \oplus c_0) \oplus \text{S-Box}(x_1 \oplus c_1)] \oplus M_1 \cdot [\text{S-Box}(y_0 \oplus c_0) \oplus \text{S-Box}(y_1 \oplus c_1)],$$

where M_0, M_1 depend on the MixLayer and c_0, c_1 depend on the secret key. By simple observation, the same output difference is given by the pair of texts

$$\hat{T}^1 = C \oplus \left[y_0 \; x_1 \; 0 \; ... \; 0\right]^T, \qquad \hat{T}^2 = C \oplus \left[x_0 \; y_1 \; 0 \; ... \; 0\right]^T.$$

Combining this result with a 1-round truncated differential with prob. 1, it is possible to set up a multiple-of-n distinguisher (where $n = 2^{s-1}$) and a mixture differential one on 2 rounds of HadesCubic. Using the inside-out approach, it is possible to set up such attack on 4-round of HadesCubic$^\pi$. As a result, it turns out that 6 rounds with full S-Box layers make HadesCubic$^\pi$ secure against these attacks.

Invariant Subspace Attack. The invariant subspace attack [26] makes use of affine subspaces that are invariant under the round function. As the round constant addition translates this invariant subspace [6], random round-constants provides a good protection against such attacks.

Integral/Square Attack. Integral cryptanalysis is a technique first applied on SQUARE [17] and is particularly efficient against designs based on substitution-permutation networks, like AES or HadesCubic.

The idea is to study the propagation of sums of values. For the case of HadesCubic, it is possible to set up an integral distinguisher over two rounds, e.g.

$$\begin{bmatrix}A\\C\\...\\C\end{bmatrix} \xrightarrow{\text{S-Box}(\cdot)} \begin{bmatrix}A\\C\\...\\C\end{bmatrix} \xrightarrow{M(\cdot)} \begin{bmatrix}A\\A\\...\\A\end{bmatrix} \xrightarrow{\text{S-Box}(\cdot)} \begin{bmatrix}A\\A\\...\\A\end{bmatrix} \xrightarrow{M(\cdot)} \begin{bmatrix}B\\B\\...\\B\end{bmatrix}$$

where A denotes an active word, C a constant one and B a balanced one[16]. Using the inside-out approach, it is possible to set up such attack on 4-round of HadesCubic$^{\pi}$. As a result, it turns out that 6 rounds with full S-Box layers make HadesCubic$^{\pi}$ secure against this attack.

B.2 Security Analysis - Algebraic Attacks

Interpolation Attack. One of the most powerful attacks against HadesCubic is the interpolation attack, introduced by Jakobsen and Knudsen [21] in 1997. In the case of a keyed function, the strategy of the attack is to construct a polynomial representation of the function without knowledge of the secret key. If an adversary can construct such a polynomial then it can compute any output without knowing the key, thus enabling forgeries (for MAC settings) and other attacks.

Let $E_k : \mathbb{F}_{2^N} \to \mathbb{F}_{2^N}$ be a keyed function. The interpolation polynomial $P(x)$ representing $E_k(x)$ can be constructed using e.g. the Vandermonde matrix - *cost* approximately of $\mathcal{O}(t^2)$ - or the Lagrange's theorem - *cost* approximately of $\mathcal{O}(t \cdot \log t)$, where x is the indeterminate corresponding to the input.

In more details, each output word of an SPN permutation can be represented as a multivariate polynomial where the variables are the inputs to each S-Box. *Consider a keyed permutation input where χ input words are unknown/variables to us, and the other $t-\chi$ words are known/fixed*:

$$\chi \text{ variables input words} \qquad \text{and} \qquad t-\chi \text{ fixed input words.}$$

Considering HadesCubic and since the S-Box is the cubic function, the degree of each word after r rounds is[17] (at most) 3^{r-1}. In particular, note that *since in one round at least one S-Box is applied and since the affine layer does not change the algebraic degree, the algebraic degree of one round is three as well.* In other words, one S-Box per round (together with an affine layer) is sufficient to increase the degree of each word. For this reason, we consider a *weaker permutation in which each round contains a single S-Box. If such a permutation is secure against*

[16] For completeness, we recall that given a set of texts $\{x_i\}_{i\in I}$, the word x^j is *active* if $x_i^j \neq x_l^j$ for each $i \neq l$, constant if $x_i^j = x_l^j$ for each i, l, and balanced if $\bigoplus_i x_i^j = 0$.

[17] Note that after the first round not all words of degree 3 appears. Indeed, the input of each S-Box in the first round is composed of a single word, which means that after the first round there is no *non-linear* mixing of different words.

an interpolation attack, also our design is secure[18] (more S-Boxes per round do not decrease the security).

A (rough) estimation of the number of monomials of the interpolation polynomial (and so of the complexity of the attack) is given by

$$(3^{r-1}+1)^{\chi} \geq 3^{(r-1)\chi},$$

since after r rounds there are t words each of degree *at least* 3^{r-1}. As a result, by requiring that the number of monomials be close to the number of possible input values $3^{(r-1)\chi} \simeq 2^{\chi n}$ (that is $3^{r-1} \simeq 2^n$), the number of rounds must be at least $r \simeq n \cdot \log_3(2)$.

As showed in [30], the interpolation polynomial is dense when working in $\mathbb{F}_p$. The situation is instead different when working in $\mathbb{F}_{2^n}$, where one needs at least $1 + \lceil \log_3(2^n-1) \rceil + \lceil \log_2(t) \rceil$ rounds in order to guarantee that E_k is *dense*.

Since S-Box$^{-1}(x) = x^{1/3} = x^{(2^{n+1}-1)/3}$ has an higher degree than S-Box$(x) = x^3$, we do not expect the attack performs better when considering the backward direction instead of the forward one.

As a result, the total number of rounds R must satisfy [19]

$$R \geq 1 + \lceil n \cdot \log_3(2) \rceil + \Phi(t)$$

to thwart the interpolation attack where

$$\Phi(t) = \begin{cases} \log_2(t) & \text{working in } \mathbb{F}_{2^n} \\ \log_3(t) & \text{working in } \mathbb{F}_p \end{cases}$$

Security up to $2^M \leq 2^N$. For completeness, we present the number of rounds necessary to provide security up to 2^M (that is, data and computational cost of the attacker upper bounded by 2^M).

Using the same argumentation given before, the number of rounds must satisfy

$$(3^{r-1}+1)^{\chi} \geq 3^{(r-1)\chi} \approx 2^{\min\{M, n\cdot\chi\}}$$

that is $r \geq 1 + \min\{n, M/\chi\} \cdot \log_3(2)$. The maximum number of attacked rounds is achieved for $\chi = 1$. As a result, we have $R_P + R_F = \big(1 + \lceil \log_3(2) \cdot \min(M, n) \rceil\big) + \Phi(t)$

[18] In the case of partial S-Box layers, it could *potentially* be possible to skip a certain number of rounds by a proper choice of the input texts (e.g. by having no active S-Box). However, we do not care about this property here, since – due to the Hades strategy – at least the first and the last 3 rounds are going to have a full S-Box layer, which guarantees full diffusion.

[19] We emphasize that in this analysis we do not take into account the cost to construct the interpolation polynomial, which is (in general) non-negligible.

Gröbner Basis Attack. The natural generalization of GCDs is the notion of Gröbner basis [16]. The attack proceeds like the GCD attack with the final GCD computation replaced by a Gröbner basis computation. Analogous to the GCD above and the interpolation analysis in the following, 1 S-Box per round is sufficient to prevent this attack (since it basically depends on the degree of the encryption function, which is independent of the number of S-Boxes per round).

For generic systems, the complexity of computing a Gröbner basis for a system of $\mathfrak{N}$ polynomials f_i in $\mathfrak{V}$ variables is $\mathcal{O}\left(\binom{\mathfrak{V}+D_{reg}}{D_{reg}}^{\omega}\right)$ operations over the base field $\mathbb{F}$ [16], where D_{reg} is the *degree of regularity* and $2 \leq \omega < 3$ is the linear algebra constant. We note that the memory requirement of these algorithms is of the same order as the running time. The degree of regularity depends on the degrees of the polynomials d and the number of polynomials $\mathfrak{N}$. When $\mathfrak{V} = \mathfrak{N}$, we have the simple closed form

$$D_{reg} := 1 + \sum_{i=0}^{\mathfrak{N}-1} (d_i - 1), \tag{11}$$

where d_i is the degree of the i-th polynomial f_i in the polynomial system we are trying to solve (see [5] for details). In the over-determined case, i.e., $\mathfrak{V} < \mathfrak{N}$, the degree of regularity can be estimated by developing the Hilbert series of an ideal generated by generic polynomials $\langle f_0, \dots, f_{\mathfrak{N}-1}\rangle$ of degrees d_i (under the assumption that the polynomials behave like generic systems). Closed form formulas for D_{reg} are known for some special cases, but not in general.

Low-Data Case: 1 input/output. In the case in which the attacker has access to a single known plaintext/ciphertext pair – denoted by $p, c \in (\mathbb{F}_{2^n})^t$ where $p \equiv (p_0, ..., p_{t-1})$ and $c \equiv (c_0, ..., c_{t-1})$, the system is described by t equations

$$\forall i = 0, ..., t-1 : \qquad c_i = f_i(p_0, ..., p_{t-1}, k_0, ..., k_{t-1})$$

in t variables $k_0, ..., k_{t-1}$ (note that the key-schedule is linear). Using the formula just given, it follows that $D_{reg} = 1 + t \cdot (d-1) \approx t \cdot 3^r$ where $d \simeq 3^r$ is the (approximately) degree after of each function f_i after r rounds. Thus, the overall complexity becomes $\mathcal{O}(\binom{t+d}{d}^{\omega})$ with the hidden constant ≥ 1. Setting $\omega = 2$, the cost of the attack is

$$\left[\binom{t + t \cdot 3^r}{t \cdot 3^r}\right]^2 \geq \left(\frac{(t \cdot 3^r)^t}{t!}\right)^2 \geq \left(\frac{t \cdot 3^r}{t}\right)^{2t} = (3^r)^{2t}$$

where $n! \leq n^n$ for all $n \geq 1$, and where $\prod_{i=1}^{n}(x+i) \geq x^n$. As a result, $r \geq 2 + \log_3(2)\frac{n}{2}$ are sufficient to prevent the attack.

Generic Case. On the other hand, the cost of the attack (potentially) decreases if the attacker has access to more than a single input/output pair of texts. In this case[20], given at most $2^N - 1$ plaintexts/ciphertexts, the number of equa-

[20] Each new (plaintext, ciphertext) pair provides a new polynomial while keeping the number of unknowns $\mathfrak{V} = t$ constant

tions is at most $t \cdot 2^N$ while the number of variables remains equal to $\mathfrak{V} = t$. Depending on parameter choices, the hybrid approach [9,10] which combines exhaustive search with Gröbner basis computations may lead to a somewhat reduced cost. Following [9,10], fixing $\chi \leq t$ input components leads to a complexity of $\mathcal{O}\left(2^{-\chi\, n} \cdot \binom{t-\chi+D'_{reg}}{D'_{reg}}^{\omega}\right)$, where $D'_{reg} \leq D_{reg}$ is the degree of regularity for the system of equation after substituting κ variables with their guesses. W.r.t. the previous case, there is no closed formula to compute D_{reg} in this over-determined case. However, since there are $\binom{t+d}{d}$ monomials of degree less than or equal to d in t unknowns, the problem reduces to linear system solving of a $\binom{t+d}{d} \times \binom{t+d}{d}$ matrix over $\mathbb{F}$, which implies $D_{reg} = \mathcal{O}(d) \approx d$.

It follows that to prevent Gröbner basis attacks, the minimum number of rounds r must satisfy

$$\binom{t-\chi+D_{reg}}{D_{reg}}^{\omega} \geq 2^{n(t-\chi)},$$

for all $\chi \in \{0, \ldots, t-2\}$ and where the degree of regularity $D_{reg} = \mathcal{O}(d) \approx 3^r$ (i.e. the maximum degree of the polynomials in the polynomial system that we are trying to solve). For our parameter choices, this expression is minimized for $\chi = 0$. By simple computation, we get

$$\binom{t+d}{d} = \frac{1}{t!} \cdot \prod_{i=1}^{t} (d+i) \geq \frac{d^t}{t!} \geq \left(\frac{d}{t}\right)^t = 2^{t \log_2(d/t)}$$

where $n! \leq n^n$ for each $n \geq 1$. Setting $\omega = 2$, we obtain $2t \log_2(d/t) \approx n \cdot t$ and

$$r \geq \log_3(2) \cdot \big(n/2 + \log_2(t)\big).$$

As a result, $R \geq 1 + \big\lceil \log_3(2) \cdot \big(n/2 + \log_2(t)\big) \big\rceil$ rounds are sufficient to protect the permutation from this attack. For the follow-up, we emphasize that the analysis just proposed is independent of the fact that the rounds contain a full or a partial S-Box layer. Moreover, as for the interpolation attack, we do not expect the attack performs better when considering the backward direction instead of the forward one.

Other Strategy. The strategy just described is not the only possible one in order to set up a Gröbner Basis Attack. In particular, each equation of degree 3^r can be re-written in a different way, e.g. as r equations each one of degree 3. *Even if this allows to reduce D_{reg}, the introduction of new intermediate variables does not lead – in general – to a reduced solving time* (remember that the cost of a Gröbner basis depends both on the number of variables and of the degree of the system of equations that we are trying to solve).

We propose a detailed analysis of this strategy in [30]. As we showed there, this second strategy does not outperform the one given before in the case in which one uses rounds with full S-Box layer to guarantee security against Gröbner basis attack. Thus, $R_F^{Grobner} \geq \lceil \log_3(2) \cdot \big(n/2 + \log_2(t)\big)\rceil + 2$ rounds with full S-Box layer are sufficient to provide security against this attack.

The situation is a little different for the case in which one uses rounds with partial S-Box layer. In this case and using an analysis similar to the one just given, the complexity of the attack is approximately given by

$$\left(\frac{27}{4}\right)^{2\cdot(R_F\cdot t+R_P)},$$

which means that

$$R_F \cdot t + R_P \geq R_P^{2nd-Grob} \approx 1 + \frac{N}{2\cdot(\log_2(27)-2)}$$

rounds are necessary to protect HadesCubic from this Gröbner basis attack. Moreover, a variant of this attack can be set up, for which we need

$$R_F \geq R^{3rd-Grob}(N,t,R_P) \equiv 1+\log_3(2)\cdot\left(\frac{N}{2t+R_P}+2\cdot\log_2(2t+R_P)-2\cdot\log_2(2t)\right)$$

rounds are necessary to protect HadesCubic from this Gröbner basis attack. In conclusion, we claim that R_F and R_P rounds s.t.

$$R_P + R_F \geq R^{1st-Grob} \qquad R_P + t\cdot R_F \geq R^{2nd-Grob} \qquad R_F \geq R^{3rd-Grob}(R_P) \tag{12}$$

where $R^{1st-Grob} \equiv 1 + \left\lceil \frac{n+2\cdot\log_2(t)}{2\log_2(3)} \right\rceil$ are sufficient to protect HadesCubic from this attack.

Security up to $2^M \leq 2^N$. For completeness, we present the number of rounds necessary to provide security up to 2^M (that is, data and computational cost of the attacker upper bounded by 2^M).

Using the same argumentation given before, in the first strategy the number of rounds is given by

$$R_P + R_F \geq \max_{\chi=0,\ldots,t-1}\left\lceil \log_3(2)\cdot\left(\min\left\{\frac{n}{2};\frac{M}{2(t-\chi)}\right\}+\log_2(t-\chi)\right)\right\rceil.$$

Taking

$$R_P + R_F \geq 1 + \left\lceil \frac{\log_3(2)}{2}\cdot\Big(\min\{n;M\}+\log_2(t)\Big)\right\rceil$$

satisfies the previous inequality. Using the same approach for the other strategy, it follows that the minimum number of rounds necessary for security are:

$$\begin{aligned}
R^{1st-Grob} &\geq 1 + \left\lceil \frac{\min\{n;M\}+2\cdot\log_2(t)}{2\cdot\log_2(3)}\right\rceil \\
R^{2nd-Grob} &\geq 1 + \left\lceil \frac{M}{2\cdot(\log_2(27)-2)}\right\rceil \\
R^{3rd-Grob}(R_P) &\geq 1 + \log_3(2)\cdot\left(\frac{M}{2t+R_P}+2\cdot\log_2\left(1+\frac{R_P}{2t}\right)\right)
\end{aligned} \tag{13}$$

rounds are necessary to protect the permutation from Gröbner basis attack.

Higher-Order Differential Attack. A well-known result from the theory of Boolean functions is that if the algebraic degree of a vectorial Boolean function $f(\cdot)$ (like a permutation) is d, then the sum over the outputs of the function applied to all elements of a vector space $\mathcal{V}$ of dimension $\geq d+1$ is zero (as is the sum of all inputs, i.e., the elements of the vector space). The same property holds for affine vector spaces of the form $\{v + c \,|\, v \in \mathcal{V}\}$ for arbitrary constant c

$$\bigoplus_{v \in \mathcal{V} \oplus c} v = \bigoplus_{v \in \mathcal{V} \oplus c} f(v) = 0.$$

This is the property exploited by higher-order differential attack [23].

Working at word level, the number of rounds R_P given by the interpolation attack provides security also against higher-order differential attacks. Indeed, for the interpolation attack it is required that the degree d after r rounds satisfies $d \geq 2^N$. Instead, for higher-order differentials (working at word level), it is sufficient that $d \geq N+1$. The conclusion follows immediately.

What happens if one works - instead - on a bit level? To prevent such attacks, ideally we would like to be able to make a statement such as "After r rounds there is no output bit and no input subspace of dimension d' s.t. the derivative of the polynomial representation of the output bit with respect to this subspace is the zero polynomial." To achieve such a goal, we need to estimate the *growth of the algebraic degree*. First of all, the degree of the S-Box $f(x) = x^3$ in its algebraic representation in $\mathbb{F}_{2^n}$ is only 2. Thus, clearly the algebraic degree of the permutation after r rounds is bounded from above by 2^r. It is furthermore generally bounded from above by $N-1$ as it is a permutation.

However, better and more realistic upper bounds (e.g. [14,34]) are given in the literature. As a main result, it turns out that the degree of the function – when it is iterated – grows in a much smoother way than expected when it approaches the number of variables. For instance, the degree of the composition of two functions $G \circ F(\cdot)$ can always be upper-bounded by $\deg(G \circ F) \leq \deg(G) \cdot \deg(F)$. However, this trivial bound is often hardly representative of the true degree of the permutation, in particular if we are trying to estimate the degree after a high number of rounds.

Better suited and certainly more realistic upper bounds are given in the literature. As we are going to recall, *these bounds depend on the number of S-Boxes per round. In the following, we limit ourselves to consider two extreme cases, that is the case of rounds with full S-Boxes and the one with only 1 S-Box.*

Case: Full S-Box Layer. For the case of rounds with full S-Box layer, a better suited and certainly more realistic upper bound was found by Boura, Canteaut, and De Canniére [14]:

Proposition 1 ([14]). *Let F be a function from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$ corresponding to the concatenation of t smaller balanced*[21] *S-Boxes $S_1, ..., S_t$ defined over $\mathbb{F}_2^n$. Then, for any function G from $\mathbb{F}_2^N$ to $\mathbb{F}_2^N$, we have*

$$\deg(G \circ F) \leq \min\left\{\deg(G) \cdot \deg(F), N - \frac{N - \deg(G)}{n'}\right\}, \tag{14}$$

where $n' = n - 1$. For the particular case in which $n \geq 3$ and all S-Boxes have degree at most $n - 2$, $n' = n - 2$.

Note that the growth of the degree is independent of the linear/affine layer.

Differing from [14], this result does not completely apply to HadesCubic, since – in the R_P rounds – the SubBytes layer only partially consists of S-Boxes and partially of the identity mapping. To overcome this problem and referring to a generalized "weaker" version of the permutation $R^{R_f} \circ L \circ R^{R_f}(\cdot)$ – e.g. as the one proposed in (10)), our idea is to choose $R_F = 2 \cdot R_f$ rounds with full S-Box layer in order to provide security against such an attack (note that by replacing the linear layer L with R_P rounds, one increases the security of our permutation).

Working with $R^{R_f} \circ L \circ R^{R_f}(\cdot)$, the number of rounds R_F is given by $R_F \geq 1 + R_{deg}$, where R_{deg} must be chosen in accordance to (14) to ensure a boolean degree higher than $N - 1$. Since the cubic case in $\mathbb{F}_{2^n}$ which has boolean degree equal to 2:

- in the case in which one aims to have a security of 2^N, the minimum number of rounds $R_F = 2 \cdot R_f$ must satisfy

$$R_F \geq 2 + 2 \cdot \left\lceil \frac{1}{2} \cdot \left(\lceil \log_2(N) \rceil + \lceil \log_{n'}(N/2) \rceil \right) \right\rceil \tag{15}$$

  (a complete and detailed computation of such number is provided in [30]);
- in the case in which one aims to have a security of 2^M, the minimum number of rounds $R_F = 2 \cdot R_f$ must be computed using formula proposed in 1 (we do not give here a closed formula).

Case: Full S-Box per Round & Security $M < N/2$. Before going on, we briefly discuss the case in which

$$M \leq N \cdot \left(\frac{1}{2} + \frac{1}{4(n-5)} \right) \approx N/2.$$

Here, the number of rounds necessary to provide security are simply given by

$$R_F \geq 1 + \lceil \log_2(M) \rceil.$$

[21] Any function $f(\cdot)$ from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is said to be *balanced* if each element in $\mathbb{F}_2^m$ has exactly 2^{n-m} preimages under $f(\cdot)$.

Indeed, exploiting the result proposed in [14] and previously recalled, note that $\deg(G) \cdot \deg(F) \leq N - \frac{N-\deg(G)}{n'}$ which corresponds to

$$2 \cdot 2^r \leq N - \frac{N - 2^r}{n'}$$

until

$$2^r \leq \frac{N \cdot (n'-1)}{2n'-1} \leq \frac{N \cdot (n'-1)}{2n'-2}.$$

Thus, if $M < N/2$, $1 + \lceil \log_2(M) \rceil$ is a good estimator of the number of rounds needed for security.

Case: 1 S-Box per Round. Even if the previous result applies also to the case of partial non-linear layer, for completeness we mention that a better bound is given in the literature:

Proposition 2 ([3]). *Let F be a function that corresponds to the parallel application of m balanced n-bit S-Boxes and an identity function of width $l = N - m \cdot n$. Thus, F is a mapping from $\mathbb{F}_2^{n \cdot m + l}$ to $\mathbb{F}_2^{n \cdot m + l}$. Let δ_k be the maximal boolean degree of the product of any k output bits of the S-Box. Then for any function G from $\mathbb{F}_2^{n \cdot m + l}$ to $\mathbb{F}_2^N$, we have*

$$\deg(G \circ F) \leq \min\Big\{\deg(G) \cdot \deg(F), \beta \cdot m + \deg(G)\Big\},$$

where $\beta = \max_{1 \leq i \leq n}(\delta_i - i)$.

As before, this result does not completely apply to HadesCubic. To overcome this problem and referring to a generalized "weaker" version of the permutation $R^P(\cdot)$ – e.g. as the one proposed before for the algebraic attacks, our idea is to choose R_P rounds with 1 S-Box in order to provide security against such an attack (note that by replacing some of these rounds with R_F rounds, one increases the security of our permutation).

Working with rounds with partial non-linear layer, the number of rounds $R_F + R_P$ is given by $R_P \geq 1 + R_{deg} - R_F$, where R_{deg} must be chosen in accordance to (14) to ensure a boolean degree higher than $N-1$ and where a certain number of rounds – namely, R_F rounds – have a full S-Box layer (this provides security both against statistical attack and provide full diffusion, as already observed for the interpolation attack).

For the cubic case in $\mathbb{F}_{2^n}$ which has boolean degree equal to 2, it follows that the minimum number of rounds $R \equiv R_F + R_P$ must satisfy

- in the case in which one aims to have a security of 2^N, the minimum number of rounds R_P must satisfy

$$R \geq 3 + \left\lceil \log_2\left(\frac{n-1}{2}\right)\right\rceil + \left\lceil \frac{2N}{n-1} \right\rceil + \left\lceil \log_{n'}\left(\frac{2N \cdot n' - (n-1)}{2N-2}\right)\right\rceil - \left\lfloor \frac{2n'}{n'-1} \right\rfloor \simeq$$
$$\simeq 1 + \lceil \log_2(n-1) \rceil + \left\lceil \frac{2N}{n-1} \right\rceil \qquad (16)$$

(using an analysis similar to the one provided in [30]);

- in the case in which one aims to have a security of 2^M, the minimum number of rounds R_P can be computed using formula proposed in Prop. 2.

Higher-Order Differential Attacks on $\mathbb{F}_p$. Here we emphasize an important difference between the higher-order differential attack on $\mathbb{F}_{2^n}$ and on $\mathbb{F}_p$. Given a function $f(\cdot)$ of degree d, the sum over the outputs of the function applied to all elements of a vector space $\mathcal{V}$ of dimension $\geq d+1$ is zero.

The crucial point here is that the previous result holds if $\mathcal{V}$ is a (sub)space, and not only a generic set of elements. While $\mathbb{F}_{2^m}$ is always a subspace of $\mathbb{F}_{2^n}$ for each $m \leq n$, the only subspaces of $\mathbb{F}_p$ are $\{0\}$ and $\mathbb{F}_p$. It follows that the biggest subspace of $(\mathbb{F}_p)^t$ has dimension t, with respect to the biggest subspace of $(\mathbb{F}_{2^n})^t$, which has dimension $n \cdot t = N$.

As a result, in the case in which a permutation is instantiated over $\mathbb{F}_p$, a lower degree (and hence a smaller number of rounds) is sufficient to protect it from the higher-order differential attack with respect to the number of rounds for the $\mathbb{F}_{2^n}$ case. In more details, the number of rounds necessary to protect our design against the interpolation attack are sufficient in order to guarantee security against this attack also.

Zero-Sum Distinguishers. The fact that some inner primitive in a hash function has a relatively low degree can often be used to construct higher-order diff. distinguishers, or *zero-sum structures*. This direction has been investigated e.g. in [14] for two SHA-3 candidates, Luffa and Keccak. More generally, a zero-sum structure for a function $f(\cdot)$ is defined as a set Z of inputs z_i that sum to zero, and for which the corresponding outputs $f(z_i)$ also sum to zero, i.e. $\bigoplus_i z_i = \bigoplus_i f(z_i) = 0$. For an iterated function, the existence of zero sums is usually due either to the particular structure of the round function or to a low degree. Since it is expected that a randomly chosen function does not have many zero sums, the existence of several such sets can be seen as a distinguishing property of the internal function.

By using the *inside-out* technique, here we investigate the minimum number of rounds of HadesCubic$^\pi$ sufficient to prevent zero-sum structures.

Definition 1 (Zero-sum Partition [14]). *Let P be a permutation from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$. A zero-sum partition for P of size $K = 2^k \lneqq 2^n$ is a collection of 2^k disjoint sets $\{X_1, X_2, ..., X_k\}$ with the following properties:*

- $X_i = \{x_1^i, ..., x_{2^{n-k}}^i\} \subset \mathbb{F}_{2^n}$ *for each* $i = 1, ..., k$ *and* $\bigcup_{i=1}^{2^{n-k}} X_i = \mathbb{F}_{2^n}$,
- $\forall i = 1, ..., 2^k$: *the set* X_i *satisfies zero-sum* $\bigoplus_{j=1}^{2^k} x_j^i = \bigoplus_{j=1}^{2^k} P(x_j^i) = 0$.

We focus on creating zero-sum partitions of the permutation $P(\cdot)$ of the form $P(\cdot) = R_r \circ ... \circ R_1(\cdot)$, where all R_i are permutations over $\mathbb{F}_2^n$. Remember that for the permutation in a hash function, one can exploit any state starting from an intermediate state. Thus, assume one can find a set of texts $X = \{x_i\}_i$ and a set of texts $Y = \{y_i\}_i$ with the property $\bigoplus_i R_{r-1} \circ ... \circ R_{s+1}(y_i) = 0$ and $\bigoplus_i R_s \circ ... \circ R_1(x_i) = 0$ for a certain s. Working with the intermediate states (remember that there is no secret material), the idea is to choose texts in $X \bigoplus Y$: the inputs p_i are defined as the $(r-s)$-round decryptions of $X \bigoplus Y$, while the corresponding outputs c_i are defined as the s-round encryptions of $X \bigoplus Y$. This results into a zero-sum partition $\{p_i\}$ for the permutation P.

To avoid such an attack, we require that $R_{r-1} \circ ... \circ R_{s+1}(\cdot)$ and $R_s \circ ... \circ R_1(\cdot)$ have maximum degree. About $R_s \circ ... \circ R_1(\cdot)$, one can simply reuse the result already proposed for the higher-order differential discussed in the previous section. Similarly, choosing $r - s = R'_{deg}$ in accordance to Prop. 1 ensures maximum degree of $R_{r-1} \circ ... \circ R_{s+1}(\cdot)$.

In order to compute the following result, we limit to recall here that the algebraic degree of S-Box$(x) = x^{1/3}$ (i.e. the inverse S-Box) is $(n+1)/2$ (see Prop.3).

Case: Full S-Box Layer. If one aims to have a security of 2^N, then - due to the analysis just proposed - it turns out that

$$\begin{aligned} R_F \geq 2 + 2 \cdot \Bigg\lceil \frac{1}{2} \cdot \Bigg\{ & \overbrace{\left\lceil \log_2\left(\frac{N\cdot(n'-1)}{2n'-1}\right)\right\rceil + \left\lceil \log_{n'}\left(\frac{N\cdot n'}{2n'-1}\right)\right\rceil}^{\text{encrypted rounds (S-Box}(x)=x^3)} + \\ & + \underbrace{\left\lceil \log_{(n+1)/2}\left(\frac{2N\cdot(n'-1)}{n'\cdot(n+1)-2}\right)\right\rceil + \left\lceil \log_{n'}\left(\frac{N\cdot n'\cdot(n-1)}{n'\cdot(n+1)-2}\right)\right\rceil}_{\text{decrypted rounds (S-Box}(x)=x^{\frac{1}{3}})} \Bigg\} \Bigg\rceil \simeq \\ \simeq 2 + 2 \cdot \Bigg\lceil \frac{1}{2} \cdot \Bigg\{ & \left\lceil \log_2(N/2) \right\rceil + \left\lceil \log_{n'}(N^2/2) \right\rceil + \left\lceil \log_{(n+1)/2}(2N/n) \right\rceil \Bigg\} \Bigg\rceil \end{aligned} \tag{17}$$

rounds with full S-Box layers prevents this attack (where n' is defined as before). If this is not the case, a zero-sum partition attack can be mounted[22].

As before, if one aims to have a security level of $2^M \leq 2^N$, the minimum number of rounds R_F in order to guarantee security against zero-sum partitions can be computed using Prop. 1. For the particular case $M < N/2$, due to the high degree of S-Box$^{-1}(x) = x^{1/3}$, we conjecture that

$$R_F \geq 3 + \log_2(M)$$

are sufficient for this goal.

[22] While it is known how to construct a zero-sum for a random permutation (see [4,8] for details), there is no way – to the best of our knowledge – to construct a zero-sum partition for a random permutation without using a brute-force approach.

Case: Partial S-Box Layer. If one aims to have a security of 2^N, then - due to the analysis just proposed - it turns out that

$$R_P + R_F \ \geq\ 7 + \left\lceil \log_2 \left(\frac{n-1}{2} \right) \right\rceil + \left\lceil \frac{3N}{n-1} \right\rceil + \left\lceil \log_{n'} \left(\frac{2N \cdot n' - (n-1)}{2N-2} \right) \right\rceil + \\ + \left\lceil \log_{n'} \left(\frac{N \cdot n' - (n-1)}{N-1} \right) \right\rceil - \left\lfloor \frac{3n' \cdot (n+1)}{(n'-1)\cdot(n-1)} \right\rfloor \simeq 5 + \lceil \log_2 (n-1) \rceil + \left\lceil \frac{3N}{n-1} \right\rceil \tag{18}$$

(where $\beta = n-1$ for S-Box$(x) = x^{1/3}$) in order to prevent zero-sum partitions.

As before, if one aims to have a security level of $2^M \leq 2^N$, the minimum number of rounds R_P in order to guarantee security against zero-sum partitions can be computed using Prop. 2 (we do not provide a closed formula for it).

Details – Algebraic Degree of S-Box$^{-1}(x) = x^{1/3}$ in $GF(2^n)$.

Proposition 3. *The algebraic degree of S-Box*$^{-1}(x) = x^{1/3} = x^{(2^{n+1}-1)/3}$ *is* $(n+1)/2$ *(remember that n is odd).*

Proof. We prove this result by induction.

For $n = 3$, it follows that S-Box$^{-1}(x) = x^{1/3} = x^5$. Since $x^5 = x^4 \cdot x$ and since x^4 is a linear operation in $GF(2^n)$, the result follows immediately.

Assume the result is true for $n - 1 = 2n' + 1$. Here we show that it works for $n = 2n' + 3$. Observe that

$$\frac{2^{n+1}-1}{3} = \frac{2^{2n'+4}-1}{3} = \frac{2^{2n'+4} - 2^{2n'+2}}{3} + \frac{2^{2n'+2}-1}{3} = 2^{2n'+2} + \frac{2^{2n'+2}-1}{3}$$

thus

$$x^{\frac{2^{n+1}-1}{3}} = x^{2^{2n'+2}} \cdot x^{\frac{2^{2n'+2}-1}{3}}.$$

Since the exponent of the first term on the l.h.s. is a power of 2, it is linear in $GF(2^n)$. By the induction assumption, the second term has algebraic degree $(n-1)/2$. It follows that the algebraic degree is $(n+1)/2$. □

Lemma 1. *Let S be the cubic S-Box, that is $S : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ s.t. $S(x) = x^3$. Let $\beta = \max_{1 \leq i \leq n}(\delta_i - i)$, where δ_k is the maximal boolean degree of the product of any k output bits of the S-Box. For the cubic S-Box S just defined, $\beta = \frac{n-1}{2}$.*

Proof. Since S is a permutation over $\mathbb{F}_{2^n}$, it follows that $\delta_i \leq n-1$ for each i (see Corollary 3.4 of [13]). Moreover, since the algebraic degree of S is 2, it follows that $\delta_k \leq k \cdot 2$. Thus $\delta_i \leq 2i \leq n-1$ if and only if $i \leq (n-1)/2$. It follows that

$$\beta \leq \max \left\{ \max_{1 \leq i \leq (n-1)/2} (\underbrace{\delta_i}_{=2i} - i),\ \max_{(n-1)/2 < i \leq n} (\underbrace{\delta_i}_{\leq n-1} - i) \right\} = \frac{n-1}{2}.$$

□

C Security Analysis – HadesInverse

Here we propose the security analysis of HadesInverse$^\pi$ instantiated[23] with S-Box$(x) = x^{-1}$ in $GF(p)$. In particular, we focus only on the attacks that *depend* on the details of the S-Box, like differential/linear attacks and the algebraic attacks.

C.1 Statistical Attacks

Since statistical attacks work in the same way in $GF(p)$ and $GF(2^n)$, in this subsection we do not distinguish the two cases.

Differential Attack. For simplicity[24], assume $N \geq 10$, that is, $n \geq 4$. Then, as for HadesInverse with S-Box$(x) = x^3$, 8 rounds with full S-Box layers are largely sufficient to prevent differential and linear attacks. In particular, w.r.t. the cubic function, the inverse function S-Box$(x) = x^{-1}$ in $GF(2^n)$ is not APN but differentially 4-uniform [31]. This means that its differential probability is bounded by 2^{-n+2}.

Thus, consider the "weaker" permutation $R^3 \circ L \circ R^3(\cdot)$ as defined in (10), and focus on the "middle" 5 rounds. Since M is an MDS matrix, at least $2 \cdot (t+1)+1$ S-Boxes are active in the middle 5 rounds of the middle permutation. As a result, each characteristic has probability

$$(2^{-n+2})^{2\cdot(t+1)+1} = 2^{-N} \cdot 2^{-N-3n+4t+6} < 2^{-N}$$

since $[N + 3n = n \cdot (t+3)] > [4t+6]$. For $n \geq 4$, the result follows immediately.

Again, 4 rounds are sufficient if $n > t+2$. Indeed, each characteristic has probability

$$(2^{-n+2})^{t+2} = 2^{-N} \cdot 2^{-2n+2t+4} < 2^{-N}$$

since $n > t+2$.

Security up to $2^M \leq 2^N$. For completeness, we present the number of rounds necessary to provide security up to 2^M (that is, data and computational cost of the attacker upper bounded by 2^M). Using the same analysis as before, it turns out that

$$R_F = \begin{cases} 4 & \text{if } 2t+4 < N+2n-M \\ 6 & \text{if } 2t+4 \geq N+2n-M \end{cases}$$

Linear Cryptanalysis. Similar considerations hold for linear cryptanalysis.

[23] We do not have any practical application of HadesInverse$^\pi$ instantiated in $GF(2^n)$. For this reason, here we limit ourselves to consider the case $GF(p)$.

[24] Due to the MDS assumption, a $t \times t$ MDS matrix with elements in $GF(2^n)$ exists if $2t+1 \geq 2^n$. If $n = 3$, then $t \leq 3$ which implies $N \leq 9$.

Rebound Attacks. Due to the same argumentation in order to provide security of HadesCubic instantiated by S-Box$(x) = x^3$ against the rebound attack, 6 rounds provide security also HadesInverse instantiated by S-Box$(x) = x^{-1}$ against the rebound attack.

C.2 Algebraic Attacks

Higher-Order Diff. Attack. We refer to previous discussion against higher-order diff. attacks over $\mathbb{F}_p$, and we limit ourselves to remember that the number of rounds necessary to guarantee security against the interpolation attack is also also sufficient to guarantee security against higher-order diff. attacks.

Interpolation Attack. As we have already seen, in an interpolation attack [21], the goal is to determine the polynomial representation of a state word. Since the inverse function has high degree, one may think that the interpolation attack can cover only few rounds in this case. However, exploiting the original idea proposed by Jakobsen and Knudsen in [21], it is possible to show that the following:

- for a full S-Box layer, the S-Box $f(x) = x^{-1}$ has the same behavior as the one of a function of algebraic degree t (i.e., the number of words)[25] "from the point of view" of the interpolation attack;
- for a partial S-Box layer (with a single S-Box), the S-Box $f(x) = x^{-1}$ has the same behavior as the one of a function of algebraic degree 2 "from the point of view" of the interpolation attack.

Note that the two previous cases lead to two completely different results, while we emphasize that the two previous cases (full or partial S-Box layer) are equivalent for a cubic S-Box. It follows that the choice to use partial or full S-Box layer in order to protect from algebraic attacks also depend on the details of the S-Box.

Full S-Box Layer. Firstly, consider $t = 1$. In this case, every encryption function can be written as

$$f(x) = \frac{x + A}{B \cdot x + C}$$

for *any* number of rounds and for some constants A, B, C. This means that 4 texts are sufficient to break the permutation.

Consider the case $t = 2$. Let $f_i^r(\cdot) \equiv \frac{Nf_i^r(\cdot)}{Df_i^r(\cdot)}$ (for $i = 0, 1$) be the interpolation polynomial at round r of the i-th word. By simple computation, the i-th word

[25] More precisely, the degree of S-Box$(x) = x^{-1} \equiv x^{p-2}$ "from the point of view of the interpolation attack" is $\min\{t, p-2\}$, where t is due to the fraction representation and $p-2$ is due to the "normal" representation. Since $2t \leq p+1$ in order to guarantee that a $t \times t$ MDS matrix with coefficients in $\mathbb{F}_p$ exists, it follows that $\min\{t, p-2\} = t$.

of the function at round $r+1$ (assuming a full S-Box layer) for $i = 0, 1$ can be written as

$$
\begin{aligned}
f_i^{r+1}(x \equiv [x_0, x_1]) &= \frac{A}{f_0^r(x \equiv [x_0, x_1]) + k_0} + \frac{B}{f_1^r(x \equiv [x_0, x_1]) + k_1} = \\
&= \frac{A \cdot Df_0^r(x)}{Nf_0^r(x) + k_0 \cdot Df_0^r(x)} + \frac{B \cdot Df_1^r(x)}{Nf_1^r(x) + k_1 \cdot Df_1^r(x)} = \\
&= \frac{A \cdot \big[Nf_1^r(x) + k_1 \cdot Df_1^r(x)\big] \times Df_0^r(x) + B \cdot \big[Nf_0^r(x) + k_0 \cdot Df_0^r(x)\big] \times Df_1^r(x)}{\big[Nf_0^r(x) + k_0 \cdot Df_0^r(x)\big] \times \big[Nf_1^r(x) + k_1 \cdot Df_1^r(x)\big]} = \\
&= \frac{Nf_i^{r+1}(x \equiv [x_0, x_1])}{Df_i^{r+1}(x \equiv [x_0, x_1])}
\end{aligned}
$$

for some constants A, B. It follows that the degree of the function increases at most by a factor of 2 (where the degree after the first round is 1). As a result, the number of unknown coefficients after r rounds is at most $2 \cdot (2^{r-1} + 1)^2$, where the degree of the numerator (and so the number of unknown coefficients) is always less or equal than the degree of the denominator.

As a result, the number of unknown coefficients after r rounds for t words is approximately

$$2 \cdot (t^{r-1} + 1)^t.$$

The permutation can be considered secure if $2 \cdot (t^{r-1} + 1)^t \simeq 2^N$, that is, $t^{r-1} \simeq p$, which implies

$$r \geq \log_t(2) \cdot \log_2(p) + 1.$$

As a result, the total number of rounds (with full S-Box layer) must be

$$R_F \geq \log_t(2) \cdot n + 2 = 2 + \frac{\log_2(p)}{\log_2(t)}.$$

Partial S-Box Layer. Referring to the expression of f_i^r given before, it possible to note that all denominators at rounds r (for any r) are in general equal, while all numerators are in general different, that is

$$\forall i, j \in [0, 1, ..., t-1] : \qquad Df_i^r = Df_j^r.$$

This observation seems to have no effect on the complexity of the previous attack. Indeed, since the S-Box are applied at each word and since the numerators are different, it turns out that the denominators of S-Box(f^r) (which correspond to the numerator of f^r) are all different.

However, this has an important effect in the case in which we work with a partial non-linear layer, e.g. a non-linear layer composed of a single S-Box. Consider first the case $t = 2$ assuming the S-Box is applied only on the first word (we use the same notation as before):

$$\begin{aligned}
&f_i^{r+1}(x \equiv [x_0, x_1]) = \frac{A}{f_0^r(x \equiv [x_0, x_1]) + k_0} + B \cdot \big[f_1^r(x \equiv [x_0, x_1]) + k_1\big] = \\
&= \frac{A \cdot Df_0^r(x)}{Nf_0^r(x) + k_0 \cdot Df_0^r(x)} + \frac{B \cdot \big[Nf_1^r(x) + k_1 \cdot Df_1^r(x)\big]}{Df_1^r(x)} = \\
&= \frac{A \cdot Df_0^r(x) \times Df_1^r(x) + B \cdot \big[Nf_1^r(x) + k_1 \cdot Df_1^r(x)\big] \times \big[Nf_0^r(x) + k_0 \cdot Df_0^r(x)\big]}{\big[Nf_0^r(x) + k_0 \cdot Df_0^r(x)\big] \times Df_1^r(x)} = \\
&= \frac{Nf_i^{r+1}(x \equiv [x_0, x_1])}{Df_i^{r+1}(x \equiv [x_0, x_1])}
\end{aligned}$$

In this case, there is no difference w.r.t. the previous case.

Consider now the case $t \geq 3$. By previous observation, it follows that $Df_i^r(x) = Df_j^r(x)$ for each $i, j \geq 1$, which implies that

$$Df_i^{r+1} = \big[Nf_0^r(x) + k_0 \cdot Df_0^r(x)\big] \times Df_1^r(x)$$

also for the case $t \geq 3$. This fact has a huge impact on the number of monomials of the corresponding polynomial at round r. Indeed, the number of unknown coefficients after r rounds for t words is approximately

$$2 \cdot (2^{r-1} + 1)^t,$$

which is much smaller than $2 \cdot (t^{r-1} + 1)^t$ for large t. The permutation can be considered secure if $2 \cdot (2^{r-1} + 1)^t \simeq 2^N$, that is, $2^{r-1} \simeq p$, which implies

$$r \geq \log_2(p) + 1.$$

As a result, the total number of rounds (with full S-Box layer) must be

$$R \equiv R_P + R_F \geq \log_2(p) + 1.$$

Actually, the previous result can be improved. Since at least $R_F \geq 6$ rounds have a full S-Box layer, it follows that the number of unknown coefficients after $R = R_P + R_F$ rounds for t words is approximately

$$2 \cdot (2^{R_P} \cdot t^{R_F - 1} + 1)^t \equiv 2 \cdot (2^{R_P + (R_F - 1) \cdot \log_2(t)} + 1)^t.$$

The permutation can be considered secure if $2 \cdot (2^{R_P + (R_F - 1) \cdot \log_2(t)} + 1)^t \simeq 2^N$, that is

$$R_P + (R_F - 1) \cdot \log_2(t) \geq \log_2(p) + 1.$$

As a result, the total number of rounds (with full S-Box layer) must be

$$R_P + \log_2(t) \cdot R_F \geq R^{inter}(N, t) \geq 2 + \log_2(p) + \log_2(t).$$

Security up to $2^M \leq 2^N$ – 1 S-Box Layer. For completeness, we present the number of rounds necessary to provide security up to 2^M (that is, data and computational cost of the attacker upper bounded by 2^M).

Using the same argumentation given before, the number of rounds must satisfy

$$2 \cdot (2^{R_P + (R_F - 1)\cdot \log_2(t)} + 1)^\chi \approx 2^{\min\{M, \log_2(p)\cdot \chi\}},$$

that is

$$R_P + R_F \cdot \log_2(t) \geq R^{inter}(N, t, M) = 2 + \log_2(t) + \min\{M, \log_2(p)\}$$

where the maximum number of attacked rounds is achieved for $\chi = 1$.

MitM Interpolation Attack. Before going on, we discuss the case of Meet-in-the-Middle version of the interpolation attack. Given (input, output) pairs (p, c), the idea is to construct two polynomials P_1 and P_2 which satisfy

$$p \xrightarrow{R^r(\cdot)} P_1(p) = P_2(c) \xleftarrow{R^{-s}(\cdot)} c,$$

i.e. that "agree" in the middle. Since they can take potentially any possible value in the middle, they can be different from the right interpolation polynomial. Thus, the idea is to check them using other (input, output) pairs.

Even if this strategy works when the goal is to set up a key-recovery attack, it seems rather hard to exploit it in the case of a permutation, e.g. in order to set up a forgery attack. Indeed, given $P_1(\cdot)$ and $P_2(\cdot)$, assume the goal is to find a pair (p', c') that satisfies $P_1(p') = P_2(c')$. Given p', one should invert $P_2(\cdot)$ in order to find the corresponding c'. Since P_2 has in general an high degree, such operation is in general infeasible. As a result, it seems rather hard to exploit the Meet-in-the-Middle version of the interpolation attack in order to set up a forgery attack (or other attack in the case of a permutation).

C.3 Gröbner Basis

Case: Full S-Box Layer. First, let's focus only on the rounds with *full S-Box layer.* After $r \geq 1$ rounds and using the "fraction representation" just proposed for the interpolation attack, the minimum degree of a variable in the output polynomials is t^{r-1}, using the equivalence

$$f(x) \equiv \frac{Nf(x)}{Df(x)} = C \quad \text{if and only if} \quad Nf(x) = C \cdot Df(x).$$

First Strategy. To prevent Gröbner basis attacks, we require

$$p^{\kappa} \cdot \binom{t-\kappa+d-1}{d-1}^{\omega} \geq p^{t}.$$

Using Stirling's approximation of the binomial when $t \ll d$, we approximate $\binom{t+d}{d}$ by $(d/t)^t = 2^{t\log_2(d/t)}$ and, setting $\omega = 2$, obtain

$$2t\log_2(d/t) \approx \log_2(p) \cdot t.$$

Since $d = t^{r-1}$, it follows that

$$r \geq 2 + \frac{\log_2(p)}{2} \cdot \log_t 2.$$

As a result, this implies that this strategy to set up the Gröbner basis attack does not outperform the interpolation attack for any values of $\log_2(p)$ and t (when working only with rounds with full S-Box layer).

Second Strategy. Note that each S-Box is described by an equation of degree 2. Thus, the attack is described by a system of $t \cdot R_F$ variables and $t \cdot R_F$ equations of degree 2, and the cost is given by

$$\left[\binom{1+2t\cdot R_F}{t\cdot R_F}\right]^2 \approx 16^{t\cdot R_F}$$

using Stirling's approximation $x! \approx x^x \cdot e^{-x} \cdot \sqrt{2\pi \cdot x}$.

At a result, the minimum number of rounds to guarantee security is given by

$$t \cdot R_F > \frac{N}{4},$$

which implies

$$R_F \geq 1 + \frac{\log_2(p)}{4}.$$

Security up to $2^M \leq 2^N$. Working in the same way as before, it follows that the minimum number of rounds necessary to guarantee security is given by

$$R_F \geq R^{Grob}(N,t,M) \equiv 1 + \frac{M}{4t}.$$

Case: Partial S-Box Layer. The previous strategies can be set up in a similar way also when one exploits rounds with partial S-Box layer in order to guarantee security against Gröbner basis attack.

First Strategy. In the first case, since the degree of after R rounds is well estimated by

$$d = t^{R_F - 1} \cdot 2^{R_P} = 2^{R_P + \log_2(t) \cdot (R_F - 1)}$$

due to the same argument given for the interpolation attack, it follows that

$$R_P + \log_2 t \cdot R_F \geq 1 + \frac{\log_2 p}{2} + 2 \log_2 t$$

rounds are necessary to guarantee security against this version of the Gröbner basis attack.

Second Strategy. In this second case, the idea is to work at round level. The number of variables and equations of degree 2 (working at S-Box level) is given by $R_P + t \cdot R_F$. It follows that $D_{reg} = 1 + R_P + t \cdot R_F$, and that the cost of the attack is given by

$$\left[\binom{1 + 2(R_P + t \cdot R_F)}{R_P + t \cdot R_F} \right]^2 \approx 16^{R_P + t \cdot R_F}$$

using Stirling's approximation $x! \approx x^x \cdot e^{-x} \cdot \sqrt{2\pi \cdot x}$.

At a result, the minimum number of rounds to guarantee security is given by

$$R_P + t \cdot R_F \geq 1 + \frac{N}{4}.$$

Security up to $2^M \leq 2^N$. Working in the same way as before, it follows that the minimum number of rounds necessary to guarantee security must satisfy the following two equivalences

$$\begin{cases} R_P + \log_2 t \cdot R_F \geq R^{1st-Grob}(n, t, M) \equiv 1 + \frac{\min\{\log_2 p, M\}}{2} + 2 \log_2 t \\ R_P + t \cdot R_F \geq R^{2nd-Grob}(n, t, M) \equiv 1 + \frac{M}{4} \end{cases}$$

C.4 Summary – Security of HadesInverse$^\pi$ in $GF(p)$

As for HadesCubic$^\pi$, we propose two recommendations, together with a script that returns the best ratio between R_P and R_F which minimizes the required metric.

First Recommendation. In this first case, HadesInverse$^\pi$ is secure in $GF(p)$ if R_F and R_P satisfies the following inequalities:

$$\begin{cases} R_F \geq \max\left\{R_F^{stat}(N, M, t) = 6; R^{Grob}(N, t, M)\right\} \\ R_P + R_F \cdot \log_2(t) \geq R^{inter}(N, t, M) = 2 + \log_2(t) + \min\{M, \log_2(p)\} \end{cases}$$

where

$$R^{Grob}(N, t, M) \equiv 1 + \frac{M}{4t}.$$

Second Recommendation. In this first case, HadesInverse$^\pi$ is secure in $GF(p)$ if R_F and R_P satisfies the following inequalities:

$$\begin{cases} R_F \ge R_F^{stat}(N, M, t) = 6 \\ R_P + R_F \cdot \log_2(t) \ge \max\{R^{inter}(N, t, M); R^{1st-Grob}(n, t, M)\} \\ R_P + t \cdot R_F \ge R^{2nd-Grob}(n, t, M) \equiv 1 + \frac{M}{4} \end{cases}$$

where $R^{inter}(N, t, M) \equiv 2 + \log_2(t) + \min(M, \log_2(p))$ as before, and where

$$R^{1st-Grob}(n, t, M) \equiv 1 + \frac{\min\{\log_2 p, M\}}{2} + 2\log_2 t.$$

C.5 Concrete Instantiations

Case:

$$N = 1743 \qquad t = 7$$

and

$$p = 2^{249} - 15145038707218910765482344729778085401$$