

Arion: Arithmetization-Oriented Hashing for Zero-Knowledge Proof Systems

Luca Campa¹ , Arnab Roy¹ , Matthias Johann Steiner²  and
Stefano Trevisani³ 

¹ University of Innsbruck, Innsbruck, Austria

² Independent Researcher, Klagenfurt am Wörthersee, Austria

³ TU Wien, Vienna, Austria

Abstract

We propose the arithmetization-oriented (AO) hash function **Arion**, following a permutation-based design approach. We first define the permutation **Arion- π** over the finite field $\mathbb{F}_p$, where ($p > 2$) is prime. The design of **Arion- π** is based on the recently introduced generalized triangular polynomial system, a novel algebraic framework for constructing cryptographic permutations using polynomials over finite fields. Using this permutation, we define the hash function **Arion** in two modes: the well-established Sponge construction and a feed-forward truncation mode (Trunc). Secure parameter sets are specified for prime fields commonly used in zero-knowledge proof applications, including the scalar fields of the BLS12-381 and BN254 elliptic curves.

We provide an extensive security analysis of **Arion**, with particular emphasis on algebraic techniques — including interpolation and polynomial system solving (PoSSo) based techniques, such as Gröbner basis computations, and resultants — which are especially relevant for cryptographic primitives defined over prime fields. To the best of our knowledge, **Arion** is the first hash function whose security analysis is explicitly based on the algebraic invariant of the underlying ideal - the *quotient ring dimension*. In particular, we explicitly determine the dimension of the quotient ring associated with the CICO problem induced by the hashing modes. Furthermore, our analysis of the CICO- t problem applies to any $t \geq 1$ and covers both the Sponge and feed-forward constructions.

We evaluate the efficiency of **Arion** across several arithmetization frameworks — R1CS, Plonk, and AIR — and compare it with prominent AO hash functions, including POSEIDON, POSEIDON2, **Anemoui**, GRIFFIN, and **Rescue**. Our results show that **Arion** is frequently the best-performing design in the Plonk setting and remains highly competitive, often ranking second, in both R1CS and AIR. In terms of native performance, **Arion** is the only construction based on high-degree power maps that achieves performance comparable to POSEIDON/POSEIDON2. This makes it an attractive choice for applications where both zero-knowledge proving efficiency and native evaluation costs are important considerations.

Keywords: Hash function, Arithmetization-Oriented, Zero-Knowledge, Algebraic Cryptanalysis

Emails: luca.campa@uibk.ac.at (Luca Campa), arnab.roy@uibk.ac.at (Arnab Roy), steiner.matthias@gmx.at (Matthias Johann Steiner), stefano.trevisani@tuwien.ac.at (Stefano Trevisani)

1 Introduction

Hash Functions and Zero-knowledge Proof Systems.

Zero-knowledge proof (ZKP) systems, in particular SNARKs (Succinct Non-interactive ARGument of Knowledge) and STARKs (Transparent ZK-SNARKs), enable efficient verification of the correct execution of a computation. Their efficiency critically depends on the cost of *arithmetizing* the target computation over a finite field.

Cryptographic hash functions play an important role in the security and efficiency of ZKP systems, both because in many applications they make up for a large part of the ZK computation, and because they can be a fundamental component of the ZKP system itself. More specifically, the two most notable use cases of hash functions in ZK-SNARKs are:

- (1) Securely and efficiently proving set membership using a Merkle Tree (MT) hash accumulator, a key component of privacy preserving blockchain applications like Filecoin [PD20] and ZCash [HBHW22]. In this setting, users $\mathcal{U}_i$ can enter their corresponding values $\mathbf{z}_i$ in the MT accumulator, and the MT root value is published. To claim the ownership of $\mathbf{z}_i$, $\mathcal{U}_i$ must prove knowledge of the value. In these proof systems, the multiplicative complexity of the function’s arithmetic circuit over a large prime field $\mathbb{F}_p$, is the dominant cost metric rather than the plain (a.k.a. native) computation time of the underlying hash function.
- (2) In recursive SNARK systems, such as Halo Infinite [BDFG21] or Fractal [COS20], the entire computation is viewed as consecutive applications of some functions $f_1, f_2, \dots, f_m$ on some state $\mathbf{s}_0$. The application of each f_i leads to an intermediate value $\mathbf{x}_i$. The proof of correctness for the computation of f_i is handed over to the next prover, who computes f_{i+1} . The efficiency of such proof systems depends again on the multiplicative complexity of the underlying hash function’s arithmetic circuit over $\mathbb{F}_p$, as well as its plain computation time.

In such settings, the choice of using classical bit-oriented hash functions like SHA-2 [NIS15a] and SHA-3 [NIS15b] leads to a high multiplicative complexity over $\mathbb{F}_p$, slowing down the proof generation process. For this reason, so-called *Arithmetization-Oriented* (AO) hash functions are designed with low multiplicative complexity (over $\mathbb{F}_p$) in mind.

AO Hash Functions. Since the first AO hash function (over a prime field) MiMC [AGR⁺16], AO hash-function design has evolved [AGP⁺19, AAB⁺20, SAD20, AKM⁺22, GKR⁺21, GKS23] from adapting classical block-cipher paradigms to developing permutation structures specifically optimized for low multiplicative complexity.

The permutation corresponding to POSEIDON (and POSEIDON2) is constructed using the partial Substitution-Permutation-Network which was formally proposed for HADES permutation [GLR⁺20]. Recently, two new AO hash functions GRIFFIN [GHR⁺23] and Anemoui [BBC⁺23] were proposed. While GRIFFIN is based on a mixture of SPN and Feistel network, Anemoui exploits the CCZ-equivalent [CCZ98] representation towards lowering multiplicative complexities of the corresponding permutations. Noticeably, POSEIDON and subsequent AO hash functions have introduced novel permutation-design methodologies that depart from, or extend, classical frameworks such as SPNs, Feistel networks, and Lai–Massey schemes.

A separate line of research has been exploring *lookup* table-based strategies [GKL⁺22, GKL⁺24, BGK⁺25a, BGK⁺25b, HHL⁺25] that can be leveraged in specific ZK frameworks [GW20].

In this work, we only consider AO hash functions which do not require dedicated lookup support, thus excluding these particular designs from our study.

Algebraic Cryptanalysis. Algebraic cryptanalysis [RAS20, ABM24, ACG⁺19, YZY25] has emerged as one of the most effective tools for evaluating the security of AO hash functions. More specifically, algebraic cryptanalysis based on polynomial system solving have been most successful in identifying security flaws of these functions. Among the various polynomial system solving (PoSSo) techniques, elimination based methods, such as Gröbner basis (GB) and resultant computation, have been the most effective. Despite the highly algebraic nature of AO hash functions, most proposals provide only limited evidence regarding resistance to GB analysis, often relying on generic complexity estimates such as Macaulay bounds. This is not necessarily due to oversight of the designers, but rather stems from the absence of well-established algebraic results that can be directly applied or utilized to assess the security of AO hash functions. Recent results have shown that rigorous GB cryptanalysis can identify security flaws in AO hash functions. For example, in [BBL⁺24, CR25, BBB⁺25b] suitable polynomial modelling were used to show polynomial complexity of GB computation in the cryptanalysis of GRIFFIN and Anemoui. In [CR25], polynomial structures and sparsity were used to prove a tight complexity of univariate polynomial computation in GB cryptanalysis. Another elimination based technique, namely the resultant computation, was used in [YZY⁺24, BBB⁺25b] to reduce the complexity of polynomial system solving based cryptanalysis. Although the methodology applied in those two elimination based techniques is different, their complexity, in particular the univariate polynomial finding step, relies on the same underlying algebraic property of the polynomial system: the quotient ring dimension. Indeed, the latter one represents an invariant of the ideal generated by the target polynomial system, and it is a key parameter in the complexity of elimination based techniques.

Since most AO hash functions are typically instantiated via Sponge-based [BDPV07] hashing modes, PoSSo-based cryptanalysis naturally targets the resulting CICO (Constrained-Input Constrained-Output) problem rather than the permutation in isolation. In algebraic cryptanalysis, one effectively analyses the constrained polynomial system which arises as a result of using the Sponge mode. Depending on the choice of the hash output size, measured in the number of $\mathbb{F}_p$ elements, the CICO problem is parametrized as CICO- t problem where t denotes the number of field elements constituting the hash output. Another important gap in algebraic cryptanalysis of the new AO hash functions is the lack of analysis of the generic CICO problem for arbitrary (but finite) t .

Polynomial-Based Design and Dedicated Cryptanalysis. The need for efficient arithmetization in ZKP systems naturally motivates algebraically structured designs for AO hash functions. At the same time, the security of such constructions should be supported by rigorous algebraic cryptanalysis rather than generic complexity estimates. Motivated by these observations, we investigate whether structured polynomial systems can simultaneously support efficient arithmetization and enable rigorous algebraic security analysis.

To this end, we design a new AO hash function Arion using the iterative triangular polynomial system proposed in [RS24]. Our proposed ZK-friendly AO hash function over $\mathbb{F}_p$ (where p is a large prime) is accompanied by a precise estimation of the quotient ring dimension of the ideal generated by the polynomial system modelling the CICO- t problem, for $t \geq 1$. From this analysis we derive precise security bounds.

1.1 Overview of Our Results

AO Hash function. In this paper we propose a new AO hash function Arion using the permutation-based design approach [BDPV07, BBC⁺23, GHR⁺23]. We propose Arion- π permutation (Section 2), based on the recently proposed triangular polynomial system for constructing cryptographic permutations [RS24]. Arion- π is defined over $\mathbb{F}_p$ with the target to achieve low multiplicative complexity in a prover circuit for ZK proof systems.

Using the Arion- π we define the hash function Arion in two modes — Sponge [BDPV07] and Trunc [GHR⁺23]. Although Arion- π is well-defined over arbitrary prime fields $\mathbb{F}_p$, the hash function Arion is defined with specific parameter choices for the permutation Arion- π , and its security analysis is given for such parametrizations. Keeping the target ZK proof systems in mind, we choose the primes corresponding to the scalar fields of the BLS12-381 and BN254 elliptic curves.

Cryptanalytic Security. We provide extensive cryptanalysis of Arion against state-of-the-art cryptanalysis techniques to justify its security (Section 3). For the recently proposed AO hash functions like GRIFFIN and Anemoi along with their corresponding feed-forward type modes, dedicated algebraic analyses have been missing. For demonstrating resistance against Gröbner basis (GB) cryptanalysis, loose generic bounds were used. Recent cryptanalysis results [KLR24, BBL⁺24, CR25, BBB⁺25b] have demonstrated that careful GB analysis is essential to rigorously establish the security of AO hash functions.

To the best of our knowledge, Arion is the first hash function (in literature) whose security is supported by dedicated algebraic cryptanalysis using the explicit estimate of the underlying quotient ring dimension, instead of applying generic bounds (that could possibly be loose). More specifically, since the quotient ring dimension determines the complexity of elimination based techniques, we explicitly determine the quotient ring dimension of the ideal generated by the polynomial system modelling the CICO- t problem for Arion, for arbitrary t . The previous algebraic analysis [BBL⁺24, BBB⁺25b] was carried out only for $t = 1$. In our algebraic analysis (see Section 3.1.1), we also considered the constrained polynomial system arising from feed-forward mode.

We stress that we focus our security analysis of Arion particularly for the primes that are relevant to the target ZK proof applications.

Efficiency of Arion. To demonstrate the efficiency of Arion (Section 4), we compared it across several arithmetization techniques, namely Plonk [GWC19], R1CS [BCR⁺19], and AIR [BBHR18], that are utilized in practically deployed ZK proof systems [Gro16, Dus22, Tea22]. Our comparison includes AO hash functions Anemoi, GRIFFIN, POSEIDON/POSEIDON2 and Rescue-Prime [SAD20, GKR⁺21, BBC⁺23, GHR⁺23, GKS23]. For considering a 128-bit security level over the target prime fields, the round numbers of some of the comparable designs — Anemoi and GRIFFIN — had to be increased in light of several new attacks [ABM24, BBL⁺24, CR25, BBB⁺25b].

Our analysis shows that in standard Plonk (Section 4.3) arithmetization, Arion requires between 1/2 and 1/7 the constraints needed by POSEIDON — a widely deployed AO hash function, and anywhere between 10–45% less constraints than Anemoi — a recently proposed hash function. Arion is also one of the best candidates in terms of R1CS and AIR (Sections 4.4 and 4.5) arithmetization complexity (usually ranking in 2nd place), making it a particularly versatile design. Finally, when implemented on an x86-64 CPU (Section 4.6), Arion is the only hash function using high-degree power maps to remain competitive with POSEIDON2 and POSEIDON (and, in some instances, can even beat the latter one), while being anywhere between 2–7 $\times$ faster than the other targets. Hence, in applications where both the ZK prover time and hash function evaluation time are relevant complexity metrics, Arion can be a particularly good choice to improve the overall performance of the system.

2 The Permutation and Hash Function

2.1 The Arion- π Permutation

The Arion hash function is constructed using a permutation-based design approach. The underlying permutation — Arion- π , is an iterative function where each round function uses

the specific triangular polynomial system defined below.

Definition 1 (Triangular polynomial system $\mathcal{F}$). *Let $p \in \mathbb{Z}_{>4}$ be a prime, n a positive integer, and $d_1, d_2, e \in \mathbb{Z}_{>1}$ be integers such that*

- (i) d_1 is the smallest positive integer co-prime with $p-1$,
- (ii) d_2 is an arbitrary integer co-prime with $p-1$, and
- (iii) $e \cdot d_2 \equiv 1 \pmod{p-1}$.

Let $\alpha_{i,1}, \alpha_{i,2}, \beta_i \in \mathbb{F}_p$ be such that $\alpha_{i,1}^2 - 4 \cdot \alpha_{i,2}$ is a quadratic non-residue modulo p . The triangular polynomial system $\mathcal{F} = (f_1, \dots, f_n)$, for constructing Arion- π is defined as

$$\begin{aligned} f_i(x_1, \dots, x_n) &= x_i^{d_1} \cdot g_i(\sigma_{i+1}) + h_i(\sigma_{i+1}) \\ f_n(x_1, \dots, x_n) &= x_n^e, \end{aligned}$$

where $1 \leq i \leq n-1$ and

$$\begin{aligned} g_i(x) &= x^2 + \alpha_{i,1} \cdot x + \alpha_{i,2}, \\ h_i(x) &= x^2 + \beta_i \cdot x, \\ \sigma_{i+1} &= \sum_{j=i+1}^n x_j + f_j(x_1, \dots, x_n). \end{aligned}$$

Although d_2 is not explicitly present in the definitions of f_i , the Conditions (ii) and (iii) in the above definition have an important role in the efficiency of the permutation in ZK setting. We will elaborate this in Section 4.2. Note that $\mathcal{F}$ is an instantiation of the polynomial system used in [RS24] to define a generic iterative permutation. The following lemma follows immediately from the above definition.

Lemma 2 (Invertibility of $\mathcal{F}$). *The map $\mathcal{F}: \mathbb{F}_p^n \rightarrow \mathbb{F}_p^n$ as defined in Definition 1 is invertible.*

Proof. Inversion of $\mathcal{F}(\mathbf{x}) = \mathbf{y}$ is straightforward. First, notice that for all $f_i \in \mathcal{F}$, the first $i-1$ unknowns are purely formal and do not appear in the polynomial. Starting from the last branch, one simply inverts the univariate power permutation $x_n = y_n^{1/e} = y_n^{d_2}$, which works because of Conditions (ii) and (iii) in Definition 1. With x_n , one can then evaluate σ_n , and in turn g_{n-1} and h_n . Since g_{n-1} is irreducible over $\mathbb{F}_p$, it has no zeros and can hence be safely divided by. Overall, inversion of the $(n-1)$ -th component is performed via

$$x_{n-1} = \left(\frac{y_{n-1} - h_{n-1}(\sigma_n)}{g_{n-1}(\sigma_n)} \right)^{\frac{1}{d_1}}$$

Where $1/d_1$ exists because of Condition (i) in Definition 1. The equation above can be immediately extended inductively to any $i < n$. $\square$

Definition 3 (Affine transformation). *Let $\mathbb{F}_p$ be the prime field of order p . Let:*

$$S = \text{swap}(1, \dots, n) = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & n-1 & n \\ 2 & 1 & 3 & 4 & \dots & n-1 & n \\ 1 & 3 & 2 & 4 & \dots & n-1 & n \\ 1 & 2 & 4 & 3 & \dots & n-1 & n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 2 & 3 & 4 & \dots & n & n-1 \end{pmatrix} \in \mathbb{F}_p^{n \times n}.$$

The affine layer of Arion- π is defined as

$$\mathcal{L}_{\mathbf{c}}: \mathbb{F}_p^n \rightarrow \mathbb{F}_p^n, \quad \mathbf{x} \mapsto \text{swap}(1, \dots, n)\mathbf{x} + \mathbf{c},$$

where $\mathbf{c} \in \mathbb{F}_p^n$ is a public constant vector.

Algorithm 1 describes an efficient way to evaluate a matrix-vector product with $\text{swap}(1, \dots, n)$. It also helps to obtain a more efficient Plonk arithmetization of Arion.

Algorithm 1 Efficient evaluation of matrix-vector product for $\text{swap}(1, \dots, n)$.

Input: $\mathbf{v} = (v_1, \dots, v_n)^\top \in \mathbb{F}_p^n$
Output: $\text{swap}(1, \dots, n)\mathbf{v} \in \mathbb{F}_p^n$
1: Initialize $\mathbf{w} = (0, \dots, 0)^\top \in \mathbb{F}_p^n$.
2: Compute $w_1 = \sum_{i=1}^n i \cdot v_i$.
3: **for** $i = 2..n$ **do**
4: Set $w_i = w_1 + v_{i-1} - v_i$.
5: **return** $\mathbf{w}$

Lemma 4 (Invertibility of S). $S = \text{swap}(1, \dots, n) \in \mathbb{F}_p^{n \times n}$ is invertible $\iff \frac{n \cdot (n+1)}{2} \not\equiv 0 \pmod{p}$.

Proof. Let $\mathbf{A}, \mathbf{B} \in \mathbb{Z}^{n \times n}$ be such that

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & \dots & 0 & 0 \\ & & & \ddots & & \\ -1 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 \\ & & & \ddots & & \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Then

$$\mathbf{BAS} = \begin{pmatrix} 1 & 2 & 3 & \dots & n-1 & n \\ 1 & 0 & 0 & \dots & 0 & -1 \\ 0 & 1 & 0 & \dots & 0 & -1 \\ & & & \ddots & & \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}$$

Echelonizing the first row of the right-hand side yields $(0, \dots, 0, \sum_{i=1}^n i)$, which proves the claim. $\square$

The above lemma implies that $\mathcal{L}_{\mathbf{c}}$ can (potentially) be non-invertible only for $n > \sqrt{p}$, which is not a concern in our target applications since $n \lll p$. Finally, we define the permutation

Definition 5 (Arion- π permutation). Let $\mathcal{F}$ and $\mathcal{L}_{\mathbf{c}_i}$ are as defined in Definition 1 and Definition 3 respectively. Then, we define:

$$\text{Arion-}\pi_r := (\mathcal{L}_{\mathbf{c}_i} \circ \mathcal{F})^r \circ \mathcal{L}_{\mathbf{c}_0},$$

where r is a positive integer denoting the number of iterations (rounds), and $\mathbf{c}_i$ are the round constants for $0 \leq i < r$.

Being a composition of permutations, Arion- π_r is clearly a permutation itself for any r . To ease the notation, in the remainder we will leave the number of rounds r implicit.

The following lemma gives the degrees of the polynomials f_i .

Lemma 6. *Let $\mathcal{F} = (f_1, \dots, f_n) \subset \mathbb{F}_p[x_1, \dots, x_n]$ be the triangular polynomial system of Arion (as in Definition 1). Then*

$$\deg(f_i) = 2^{n-i} \cdot (d_1 + e) - d_1.$$

Proof. We perform an upwards induction, for n and $n - 1$ the claim is clear. Suppose the claim is true for indices greater than or equal to i , i.e. $\deg(f_i) = 2^{n-i} \cdot (d_1 + e) - d_1$. By construction, the degree of f_{i-1} is equal to the degree of $x_{i-1}^{d_1} \cdot f_i^2$. Thus:

$$\deg(f_{i-1}) = \deg(x_{i-1}^{d_1} \cdot f_i^2) = d_1 + 2 \cdot (2^{n-i} \cdot (d_1 + e) - d_1) = 2^{n-(i-1)} \cdot (d_1 + e) - d_1. \quad \square$$

2.2 Hash Function Mode

For the hash function Arion over $\mathbb{F}_p^n$, we instantiate Arion- π in Sponge mode [BDPV07, BDPV08]. We recommend employing the recently proposed SAFE framework [AKMQ23, KBM23], which discusses proper domain separation and padding for sponge functions over prime fields. In sponge mode the state size $n = b + c$ is split into the rate part b and the capacity part c . In [KBM23, Theorem 2] it has been proven that for a random permutation the sponge construction is indifferentiable from a random oracle up to $\min\{p^b, p^{c/2}\}$ queries. Therefore, to provide λ bits of security, $p^b \geq 2^\lambda$ and $p^{c/2} \geq 2^\lambda$, we require that $b \geq \frac{\lambda}{\log_2(p)}$ and $c \geq \frac{2 \cdot \lambda}{\log_2(p)}$.

2.3 Arion Specification

Keeping the intended applications in mind, the prime fields for Arion are the ≈ 256 -bit scalar fields of the elliptic curves BLS12-381 [Edg23] and BN254 [Wan22]. In Table 1 we provide the recommended number of rounds of the Arion- π permutation needed to guarantee a security level of 128 bits for the Arion hash function.

As for the remaining parameters, in both the target fields, following Definition 1, we obtain that $d_1 = 5$. Additionally, in both cases we choose $d_2 = 257$. Our choice is motivated by the following reasons:

- (1) It is compliant with Condition (i) in Definition 1 for the scalar fields of both BN254 and BLS12-381.
- (2) It provides at least 128 bits of security against Gröbner basis attacks at a very low number of rounds (see Section 3.2).
- (3) It has an efficient addition chain, making it possible to evaluate the corresponding power-map with 8 squarings and 1 multiplication, without needing any additional intermediate storage. This is beneficial both for native evaluation and for arithmetization (see Section 4.2).

The coefficients $\alpha_{i,1}, \alpha_{i,2}, \beta_i$, as well as the affine layer vectors $\mathbf{c}_i$ are chosen by following a nothing-up-my-sleeves approach, by employing rejection-sampling over the binary digits of π . The reader can refer to Appendix A for the exact details. A reference implementation is provided for open access on Zenodo [CRST26].

For the target fields, we recommend capacity $c = 1$ for sponge mode. This means that the generic security bound provided by the hash function is ≈ 127 bits for BN254 and ≈ 127.5 bits for BLS12-381. The number of rounds for Arion- π in Table 1 are chosen to provide roughly 128 bits of security against state-of-the-art polynomial system solving techniques [BBL⁺24, BBB⁺25b]. We also wish to emphasize in advance that, for the relevant parametrizations, the CICO-1 analysis (see Section 3.1.1) is the main focus for the security of Arion. For this reason, we require using a single field element as the hash output (taken from the rate part, as required by a Sponge function, namely we suggest taking the first one).

Table 1: Arion- π and Arion round numbers over primes $p \geq 2^{250}$ with $d_1 = 5$, $d_2 = 257$ such that $\gcd(d_i, p - 1) = 1$ that reach $\lambda = 128$ bits of security. Round number r has been computed as $\max(r_{\text{GB}}, r_{\text{Res}}) + 1$. The additional round is needed to protect Arion from the skip-round technique of [BBL⁺24]. See Section 3 for more details.

n	3	4	5	6	8	9	10	12
r	11	10	9	9	8	8	8	7

2.3.1 Evolution of Arion

In comparison with the earlier specification drafts of Arion [RST23], the current specification has changed in the following aspects:

- For keeping the specification simple, we only propose a unique affine layer (and avoid the circulant matrix specified in [RST23]). The current specification excludes any aggressive version.
- The choice of parameters is further supported by improved (algebraic) cryptanalysis (see Section 3) compared to [RST23]. In light of the recent cryptanalysis (in particular [YZY⁺24]), the number of rounds has been increased.

3 Security of Arion

3.1 PoSSo-Based Cryptanalysis

Gröbner basis (GB), as well as Resultant -based cryptanalysis, are essentially polynomial system solving (PoSSo) techniques that employ the elimination theoretic methods. Both techniques end by finding a univariate polynomial in some unknown input variable. The complexity of determining such a polynomial depends on the quotient ring dimension d_I of the ideal I defined by the system of equations describing the target cryptographic primitive. The quotient ring dimension is an algebraic invariant and is used to estimate the complexity of the elimination-based PoSSo algorithm. Consequently, it determines the complexity of such PoSSo-based algebraic cryptanalysis and the minimum number of rounds needed to reach a certain security level. Recent advances in cryptanalysis [BBL⁺24, YZY⁺24, BBB⁺25b, CR25, BBB⁺25a] of AO primitives have shown that determining the security of a design w.r.t the complexity of GB computation is not reliable (e.g. using Macaulay’s bound) and is not tight. While estimating the complexity of GB computation is non-trivial in general, existing works [BBL⁺24, CR25] have shown that complexity of GB computation can in fact be polynomial time (in the context of AO hash function), and thus can not be relied upon to determine the security of an AO hash function. On the other hand, procedures depending on the quotient ring dimension are well studied in the literature as well as their complexity, e.g. the complexity of computing the characteristic polynomial of a matrix of size $d_I \times d_I$ is given as $O(d_I^\omega)$ where ω is the algebra constant. Thus, in the security analysis of Arion, we do not rely on the computational difficulty of GB computation, and we focus on the complexity of the univariate polynomial finding step or the corresponding root finding step.

3.1.1 Polynomial Modeling of Arion

Let’s consider the Arion- π permutation both in sponge and feed-forward mode. Let’s consider n branches and let’s focus on the k -out-of- n instantiation, that is when the output is composed of k elements (in feed-forward constructions it translates to cropping the output to k elements).

Let c denote the number of field elements in the capacity. When the primitive is used in sponge mode $c > 0$, we must set other $n - k - c$ inputs to random values in order to obtain a determined polynomial system. When Arion- π is used in feed-forward mode (e.g. Trunc), there is no capacity ($c = 0$) and we must set $n - k$ inputs to random values in order to obtain a determined polynomial system. Instead of setting those inputs to random values, one could also add additional $n - k - c$ linear relations, either random or chosen by exploiting the structure of the linear layers (e.g. as happened in round-skipping attacks [MG26]). As a result, in both cases, we obtain $n - k$ fixed inputs and k fixed outputs. Once this is done, we can proceed with solving the polynomial system arising from the following CICO- $(n-k, k)$ problem.

Definition 1 (CICO- $(n-k, k)$ Problem [CBD⁺20]). Let $F : \mathbb{F}_p^n \rightarrow \mathbb{F}_p^n$ be a function and let $k < n$ be an integer. The CICO- $(n-k, k)$ problem consists of finding $x \in \mathbb{F}_p^k, y \in \mathbb{F}_p^{n-k}$ such that $F(\mathbf{0}^{n-k} || x) = (\mathbf{0}^k || y)$.

Polynomial Model for Arion- π . We considered the following polynomial iterated model for Arion- π :

- (i) For $0 \leq i \leq r - 1$ we denote with $\mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_n^{(i)})$ the inputs to the $(i + 1)$ -th round where each round, by specification, is composed of a linear layer $\mathcal{L}_o$ and a non-linear layer $\mathcal{F}$. Moreover, we use $\mathbf{a} \in \mathbb{F}_p^k$ to denote a valid hash value produced by Arion.
- (ii) For our polynomial model we model each round by integrating both the affine layer and the subsequent non-linear layer into a single polynomial system:

$$\tilde{\mathcal{F}}_i = \mathcal{F}_i(\mathcal{L}(\mathbf{x}^{(i-1)})) - \mathbf{x}^{(i)}.$$

For $1 \leq i \leq r$, the $\tilde{\mathcal{F}}_i$ follows the above representation, and we modify $\tilde{\sigma}_{j+1, i} = \sum_{k=j+1}^n (x_k^{(i-1)} + x_k^{(i)})$. The following polynomials define the naive model $\mathcal{F}_i$ for the i th round of Arion

$$\begin{aligned} s_j^{(i)} &:= (\mathcal{L}(\mathbf{x}^{(i-1)}))^{d_1} g_j(\tilde{\sigma}_{j+1, i}) + h_j(\tilde{\sigma}_{j+1, i}) - x_j^{(i)} \quad \text{for } 1 \leq j \leq n - 1 \\ s_n^{(i)} &:= \mathcal{L}(\mathbf{x}^{(i-1)}) - (x_n^{(i)})^{d_2}. \end{aligned}$$

- (iii) After the last round we must consider the equations generated by the last linear transformation and, in particular, those connected to the outputs replaced with the valid hash $\mathbf{a}$. In the case of Sponge mode those equations are

$$\mathcal{L}(\mathbf{x}^{(r)}) - \mathbf{a}$$

whilst in feed-forward mode are

$$\mathcal{L}(\mathbf{x}^{(r)}) + \mathbf{x}^{(0)} - \mathbf{a}.$$

Eliminating Superfluous Variables. Let us consider the previous polynomial model. We provide the detailed analysis for the first round then the same approach is extended to all rounds:

- (i) Having $n - k$ constrained inputs implies that the vector $\mathbf{x}^{(0)}$ only depends on k unknown inputs (variables), say $x_1^{(0)}, \dots, x_k^{(0)}$. Consider the polynomial representation of the first round. $s_n^{(1)}$ remains as it is and $\deg(\text{LM}(s_n^{(1)})) = d_2$. Consider the polynomials $s_j^{(1)}$ for $k+1 \leq j \leq n-1$. To proceed with the variable elimination, we choose

the weights as it has been done in [BBL⁺24], e.g. $\text{wt}(x_{n-1}^{(1)}) = d_1 \cdot \max(\{\text{wt}(\mathbf{x}^{(0)})\}) + 3$ and $\text{wt}(x_j^{(1)}) = 2 \cdot \text{wt}(x_{j+1}^{(1)}) + 1$ for $k+1 \leq j \leq n-2$. Moreover, note that the variables $x_j^{(1)}$ for $k+1 \leq j \leq n-1$ appear with degree 1 in the corresponding equations $s_j^{(1)}$. Hence, as these variables are “isolated” (something like the “isolation property” highlighted in [YZY⁺24]), we can represent those polynomials as $x_j^{(1)} = s_j^{(1)} + x_j^{(1)}$ for $k+1 \leq j \leq n-1$ (note that now, the right-hand side of the equations does not contain $x_j^{(1)}$).

For constructing the polynomials for the entire round we replace the variables $x_j^{(1)}$ in the equations $\mathbf{s}_{1:k}^{(1)}$ with $s_j^{(1)} - x_j^{(1)}$ for $k+1 \leq j \leq n-1$. Thanks to these linear substitutions, the equations $\mathbf{s}_{1:k}^{(1)}$ are modified such that $\deg(\text{LM}_m(s_j^{(1)})) = 2^{n-k}(d_1+1) - d_1$ for $1 \leq m \leq k$ and $1 \leq j \leq k$. The derived degree $2^{n-k}(d_1+1) - d_1$ can be easily obtained by considering a slight modification of Lemma 6.

Note that the replacement of the variables $x_j^{(1)}$ for $k+1 \leq j \leq n-1$ must also be performed in the subsequent round, as, by definition, the variables $x_j^{(1)}$ for $1 \leq j \leq n$ are the inputs of round 2.

- (ii) The same idea can be easily extended to subsequent rounds. Note that from round 2, we are in the same situation of the first round. Indeed, at the i -th round, the variables $\mathbf{x}_{k+1:n}^{(i-1)}$ have already been considered as leading monomials in the equations $\mathbf{s}_{k+1:n}^{(i-1)}$. Then, at each round we obtain:

- 1 equation of degree d_2
- k equations of degree $2^{n-k}(d_1+1) - d_1$.

3.1.2 Bounding the Quotient Ring Dimension

Before proceeding with the quotient ring dimension analysis, we need to recall some theorems and definitions related to Bézout bounds.

Theorem 7 (Bézout’s Theorem). *Let $I = \langle f_0, \dots, f_{m-1} \rangle \subset \mathbb{F}_p[x_0, \dots, x_{m-1}]$ be a zero-dimensional ideal and let $d_i = \deg(f_i)$ denote the total degree of f_i , for $i = 0, \dots, m-1$. If the system has a finite number of solutions (following from the finiteness theorem [CLO15] and zero-dimensional ideal), then the number of solutions (counting multiplicities) over the algebraic closure $\overline{\mathbb{F}_p}$ is at most the product of the degrees of the polynomials, i.e., $\prod_{i=0}^{m-1} d_i$.*

The number of solutions is exactly the quotient ring dimension that we want to bound and prove. It can be shown that if the number of solutions in $\overline{\mathbb{F}_p}$ is finite, that is, if I is zero-dimensional, then the (weighted) homogenization only adds a finite number of additional solutions at infinity over the projective space [MW83]. Thus, the bound is sharp if and only if the number of solutions at infinity is zero (except the trivial solution at the origin).

Assuming our polynomial system to behave like a random multivariate dense homogeneous polynomial system, we can present the following proposition, which provides a bound on the quotient ring dimension of the CICO- $(n-k, k)$ problem for Arion- π .

Proposition 1 (Quotient Ring Dimension of the CICO- $(n-k, k)$ Problem for Arion- π). *Consider an Arion- π instance (either used in Sponge or Feed-Forward mode) with r rounds and n inputs. Let I be the ideal generated by the polynomial system modelling the CICO- $(n-k, k)$ problem for Arion- π . Assuming the system behaves like a random multivariate homogeneous polynomial system the quotient ring dimension of I is bounded as $d_I \leq \left((2^{n-k}(d_1+1) - d_1)^k d_2 \right)^r$ with high probability.*

Proof. We want to show that the number of solutions at infinity is zero with high probability. Because we are dealing with weighted orders, we need to consider a weighted homogenization procedure. Let's denote with w_i the weight of variable x_i and with d_i the total degree of polynomial f_i . The weighted homogenization of f_i is defined as follows:

$$\tilde{f}_i = x_0^{d_i} f_i \left(\frac{x_1}{x_0^{w_1}}, \dots, \frac{x_n}{x_0^{w_n}} \right),$$

where x_0 is the homogenization variable. The homogenized ideal is then defined as $\tilde{I} = \langle \tilde{f}_1, \dots, \tilde{f}_m \rangle$ where m is the number of polynomials in the system. Counting the solutions at infinity is equivalent to counting the solutions of the homogenized system $\tilde{I}$ that satisfy $x_0 = 0$.

Let's consider a generic round i . Per round, after replacing x_0 with 0, we first obtain 1 homogeneous equation of degree d_2 of the form $(x_n^{(i)})^{d_2} = 0$, whose unique solution is $x_n^{(i)} = 0$.

Then, replacing $x_n^{(i)} = 0$ inside the other k equations leads to k homogeneous equations of degree $\delta = 2^{n-k}(d_1 + 1) - d_1$ of the form

$$\sum_{d_1 + \dots + d_k = \delta} \prod_{z=1}^k \left(x_z^{(i-1)} \right)^{d_z} = 0,$$

where coefficients are omitted for the sake of simplicity.

In other words, we obtain r homogeneous systems of k equations in k variables, where the degree of each equation is $\delta = 2^{n-k}(d_1 + 1) - d_1$. It is well-known in the literature [CLO15] that, with high probability, the only solution of a dense homogeneous system of k independent (otherwise it would not be a determined polynomial system) equations of degree δ in k variables is the trivial solution at the origin. Hence, each of those systems has only the trivial solution at the origin. As a result, the number of non-trivial solutions at infinity is zero and the Bézout bound is tight. $\square$

It is important to note that the previous analysis gives two different outcomes depending on whether $k > 1$ or not.

The case $k = 1$. This is the most (only) relevant case for our analysis, as Arion is intended to output a single field element. In this case, the previously described procedure directly leads to a GB where the leading monomials are pairwise co-prime and pure powers. As a consequence, the system has always no solutions at infinity (after homogenization we are left with a system of univariate power functions whose unique solution is obviously the trivial one) and the bound on the quotient ring dimension is tight: $d_I = ((2^{n-1}(d_1 + 1) - d_1) d_2)^r$. Moreover, since the ideal is zero-dimensional and has no solutions at infinity, by [CR25, Theorem 8] and [CD23, Theorem 1.2] the ideal has Shape form, thus justifying a possible application of the FGLM algorithm to solve the system after GB computation. These theoretical results are also validated experimentally. In particular, we tested over different parameters for round-reduced versions: $r = 3, 4$, $n = 3, 4, 5$, 32/64-bit prime fields, $d_1 = 3$, $d_2 = 7, 11, 23, 121, 257$. In all cases, the GB computation led to a GB with pairwise co-prime leading monomials and pure powers, confirming the theoretical analysis. This analysis confirms the bound derived in [BBL⁺24] for the CICO-1 case. However, as shown in [BBL⁺24], the case $k = 1$ allows skipping one round, leading to a quotient ring dimension of $d_I = 3d_1 ((2^{n-1}(d_1 + 1) - d_1) d_2)^{r-1}$ which is then used to compute the complexity of the resultant-based attack (see Section 3.2). Consequently, as Arion outputs one field element, the CICO-1 case and the corresponding bounds determine the number of rounds. More specifically, we decided to use the complexity provided by the resultant attack to determine the number of rounds, as it is the best known attack for the CICO-1 case (see Section 3.2).

The case $k > 1$. When $k > 1$, the previous analysis depends on the choice of the linear layer. In this case, the bound on the quotient ring dimension may not be tight, and we can only provide an upper bound: $d_I \leq \left((2^{n-k}(d_1 + 1) - d_1)^k d_2 \right)^r$. Specifically, our experimental results show that the bound is highly dependent on the choice of the linear layer: an MDS matrix always leads to a value closer to the upper bound; the actual Arion- π linear layer, instead, leads to a value that is significantly lower than the upper bound and that is the reason why we stress that *the matrix we proposed for Arion- π is meant to be used only when 1 output is produced and it must not be used otherwise* (as also noticed in [BBB⁺25a, Sec. 5.2]). As before, even this result on the quotient ring dimension has been supported by experimental evidences on round-reduced versions: $r = 2, 3$, $n = 4, 5$, $k = 2, 3$, $d_1 = 3$, $d_2 = 7, 11$, 16/20-bit prime fields and several linear layers. Which properties are necessary to guarantee that the bound is tight for the case $k > 1$ is an interesting open problem that we leave for future work.

3.2 Other Algebraic Cryptanalysis

3.2.1 Resultant Attack

We considered the recently published resultant-based cryptanalysis [YZY⁺24, BBB⁺25b] of the CICO-1 problem for deciding the round numbers in Arion- π . More specifically, we used the complexity of the resultant-based cryptanalysis provided in [BBB⁺25b, Corollary 3] to determine the minimal round number r that satisfies

$$3 \cdot d_1 \cdot \left((2^{n-1} \cdot (d_1 + 1) - d_1) \cdot d_2 \right)^{r-1} \cdot d_2^2 \cdot \left\lceil \frac{2d_2 - 1}{d_2} \right\rceil^{r+1} \geq 2^\lambda, \quad (1)$$

where λ denotes the security level (in bits).

3.2.2 Interpolation & Integral Cryptanalysis

Interpolation attacks [JK97] construct the polynomial vector representing the outputs of a hash function. For a hash function, interpolating the polynomial vector representing the function can be used to set up a collision or forgery attack, whose cost depends on the density of the polynomials.

Recall that any function $F : \mathbb{F}_p^n \rightarrow \mathbb{F}_p$ can be represented by a polynomial $f \in \mathbb{F}_p[\mathbf{X}_n] = \mathbb{F}_p[x_1, \dots, x_n] / (x_1^p - x_1, \dots, x_n^p - x_n)$, thus at most p^n monomials can be present in f . After the first round of Arion- π the terms

$$\left(\sum_{i=1}^n i \cdot x_i \right)^e + \sum_{i=1}^n i \cdot x_i \quad (2)$$

are present in every branch. After another application of the round function the terms

$$\left(\left(\sum_{i=1}^n i \cdot x_i \right)^e + \sum_{i=1}^n i \cdot x_i \right)^e \mod (x_1^p - x_1, \dots, x_n^p - x_n) \quad (3)$$

are produced in every branch. By our specification e is the inverse exponent of a relatively low degree permutation, therefore we expect that after two rounds almost all monomials from $\mathbb{F}_p[\mathbf{X}_n]$ are present in every component of Arion. By taking Meet-in-the-Middle attacks into account we conjecture that Arion resists interpolation attacks for $r \geq 5$ over the target primes. For more details we refer to Appendix D.1.

Given a polynomial $f \in \mathbb{F}_p[x_1, \dots, x_n]$ an integral distinguisher [KW02, BCD⁺20] exploits that for any affine subspace $V \subset \mathbb{F}_p^n$ with $\deg(f) < \dim(V) \cdot (p-1)$ one has that

$$\sum_{\mathbf{x} \in V} f(\mathbf{x}) = 0. \quad (4)$$

If almost all monomials are present in $\text{Arion-}\pi$, then $\deg(\text{Arion-}\pi) \approx n \cdot (p - 1)$ in every component, so only $V = \mathbb{F}_p^n$ is a suitable subspace for an integral distinguisher. Therefore, we conjecture that non-trivial integral distinguishers do not exist for $\text{Arion-}\pi$ with $r \geq 5$. For more details we refer to Appendix D.2.

The Arion permutation achieves a high density as well as high polynomial degrees, thus we do not expect that interpolation attacks can undermine the 128 bit security target. Note that the high density as well as the high polynomial degrees also protect against integral and higher-order differential distinguishers respectively attacks.

3.3 Statistical Cryptanalysis

3.3.1 Differential Cryptanalysis

In differential cryptanalysis [BS91] and its variants the propagation of input differences through the rounds of a hash function is exploited to construct a collision. For the Arion non-linear layer $\mathcal{F} : \mathbb{F}_p^n \rightarrow \mathbb{F}_p^n$ (Definition 1) the probability that an input difference $\Delta \mathbf{x} \in \mathbb{F}_p^n \setminus \{\mathbf{0}\}$ propagates to the output difference is $\Delta \mathbf{y} \in \mathbb{F}_p^n$ is bounded by (see [RS24, Corollary 18])

$$\mathbb{P}[\mathcal{F} : \Delta \mathbf{x} \rightarrow \Delta \mathbf{y}] \leq \left(\frac{d_2}{p}\right)^{\text{wt}(\Delta \mathbf{x})} \leq \frac{d_2}{p}, \quad (5)$$

where $\text{wt} : \mathbb{F}_p^n \rightarrow \mathbb{Z}$ denotes Hamming weight (the number of non-zero field elements in a vector). E.g., for $\log(p) \approx 256$ and $d_2 \leq 257$ this probability is bounded by $2^{-247 \cdot \text{wt}(\Delta \mathbf{x})} \leq 2^{-247}$.

Arion is an iterated scheme, if we assume that the differences among the rounds are independent, then we can bound the probability of a differential distinguisher covering r rounds by $\leq \left(\frac{d_2}{p}\right)^r$. We require for the security level λ of Arion against a differential characteristic covering r rounds that

$$\lambda \leq r \cdot (\log_2(p) - \log_2(d_2)). \quad (6)$$

We conjecture that $r + 2$, where r is the minimal round number satisfying Equation (6), are sufficient to protect Arion against differential cryptanalysis. Thus, for $\log(p) \approx 256$, $d_2 = 257$ and $\lambda = 128$ Arion requires at least $1 + 2 = 3$ rounds.

For more details we refer to Appendix B.1.

3.3.2 Truncated Differential & Rebound Cryptanalysis

In a truncated differential attack [Knu95] an attacker can only predict parts of the difference between pairs of plaintexts. Suppose the Arion polynomial system admits truncated differentials of Hamming weight 1 with probability 1 for the first round. If $\text{wt}(\mathbf{v}) = 1$, then we have that $\text{wt}(\text{swap}(1, \dots, n)\mathbf{v}) = n$. Therefore, such a truncated differential activates all inputs in the second round of Arion. Hence, for $\log(p) \approx 256$ and $d_2 = 257$ the differential probability for the second round is bounded by $2^{-247 \cdot n}$.

If an adversary can cover 2 rounds with a probability 1 truncated differential, then he still has to search $r - 2$ remaining rounds with a differential characteristic. Thus, we can reuse Equation (6) with $r \mapsto r - 2$ to estimate the security level against such a truncated differential characteristic. For $\log(p) \approx 256$, $d_2 = 257$ and $\lambda = 128$ we conjecture that $4 + 2 = 6$ rounds are sufficient to protect Arion against truncated differential attacks.

In a rebound attack [LMR⁺09, MRST09] an adversary connects two (truncated) differentials in the middle of the hash function. Probability 1 truncated differentials can cover at most one round of Arion, so $r - 2$ rounds can be covered with an inside-out approach. By our previous analysis we do not expect that a successful rebound attack can be mounted on 6 or more rounds on Arion within the 128 bit security target. For more details we refer to Appendix B.2.

3.3.3 Linear Cryptanalysis

Linear cryptanalysis [Mat94, BSV07] utilizes affine approximations of round functions for a sample of known plaintexts. For any non-trivial additive characters $\chi, \psi : \mathbb{F}_p \rightarrow \mathbb{C}$, the absolute value for the correlation of the Arion polynomial system is bounded by (see [RS24, Theorem 22])

$$|\text{CORR}_{\mathcal{F}_{\text{Arion}}}(\chi, \psi)| \leq \frac{d_2 - 1}{\sqrt{q}}. \quad (7)$$

E.g., for $\log(p) \approx 256$ and $d_2 = 257$ this probability is bounded by 2^{-120} .

Under the assumption that correlations are independent among the rounds of Arion- π , the correlation of any non-trivial linear trail covering r rounds is bounded by $\left(\frac{d_2-1}{p}\right)^r$. For the security level λ of Arion- π , against a linear characteristic, we require that

$$\lambda \leq r \cdot \left(\frac{\log_2(p)}{2} - \log_2(d_2) \right). \quad (8)$$

We conjecture that $r + 3$ rounds, where r is the minimal round number satisfying Equation (8), are sufficient to protect Arion against linear cryptanalysis. Thus, for $\log(p) \approx 256$, $d_2 = 257$ and $\lambda = 128$ Arion requires at least $2 + 3 = 5$ rounds. For more details we refer to Appendix B.3.

4 Performance Evaluation

To better assess the performance of Arion relative to other AO hash functions, we compare several selected instances in one of the most prominent applications of AO hash functions: the construction and verification of Merkle tree roots. In particular, the fundamental building block of a Merkle Tree hash is a fixed-length hash function mapping $\ell = ts$ $\mathbb{F}_p$ elements into s field elements. Usually, s is chosen large enough to guarantee security against generic attacks, and $t = 2$, although other small values, such as other small powers of two, are also common. As we settle for a security level of 128-bits, since $\log_2(p) \approx 256$, $s = 1$ is sufficient to guarantee resistance against generic collision attacks. Hence, in our experiments we will consider hash functions that can accommodate inputs of size $\ell \in \{2, 4, 8\}$ and output a single field element.

There are two main performance metrics against which an AO hash function ought to be evaluated: its *native* efficiency, i.e. the execution time of its direct implementation (say on a CPU), and its *ZK-circuit* performance, i.e. its complexity with respect to some arithmetization technique. We have measured the arithmetization complexity of Arion, and compared it against other popular AO hash functions, in terms of Plonk (Section 4.3), R1CS (Section 4.4), and AIR (Section 4.5) arithmetization. Additionally, we have experimentally compared the native performance of Arion and the other designs on an x86-64 CPU (Section 4.6).

4.1 Comparable Designs and Their Parametrization

To ensure a fair and reasonably extensive evaluation of Arion, we selected, to the best of our knowledge, some of the most popular AO designs that have been proposed for similar use-cases. Note that, we will exclude constructions tailored for small prime fields, as well as lookup-oriented constructions intended for different types of proof systems.

For these reasons, our comparisons include the following designs: Anemoi [BBC⁺23], GRIFFIN [GHR⁺23], POSEIDON [GKR⁺21] and its recent variation, POSEIDON2 [GKS23], and Rescue-Prime [SAD20]. For each of them, we consider the corresponding Sponge-based hash function under different input rates. Throughout this comparison, we use ℓ

to denote the input size (equal to the Sponge rate), while the r , consistently with the notation introduced above, denotes the number of rounds.

For some constructions, subsequent cryptanalytic results in the literature have shown that parameter sets originally believed to achieve a given security level in fact provide lower security than initially estimated. Since the parameterization of **Arion**, and in particular the number of rounds of its underlying permutation, takes these more recent results into account, we briefly summarize below how the parameters of the target designs, specifically, their number of rounds, were selected for our comparison.

Remark 8 (Small degree power map). A common feature of all target designs is their use of a small degree power map. While it is always defined in the same way, the name attached to it can differ, which may lead to confusion. In particular, **Anemoui**, **Rescue-Prime** and **POSEIDON** name the power map’s exponent α , while **GRIFFIN** and **POSEIDON2** name it d , and we name it d_1 . In the specific case of the target fields (BN254, BLS12-381), the exponent is equal to 5.

- **Griffin.** In [BBL⁺24], various parametrizations of the GRIFFIN hash function were shown to be insecure with respect to the ‘FreeLunch’ cryptanalysis. Furthermore, a recent cryptanalysis based on resultants [BBB⁺25b] can also be applied to GRIFFIN. Therefore, we account for the complexity estimates derived from [BBL⁺24, §4.1] and [BBB⁺25b, Corollary 3], on top of the bounds proposed by GRIFFIN’s authors. Finally, we remark that the permutation underlying the GRIFFIN hash function is only well-defined over $\mathbb{F}_p^n$ when $n = 3$ or $n \equiv 0 \pmod{4}$, hence a larger state size than what would be strictly necessary is required to hash inputs at a rate of 4 or 8.
- **Anemoui.** Similarly to GRIFFIN, **Anemoui** is also susceptible, albeit to a minor extent, to the resultant-based cryptanalysis of [BBB⁺25b], and, depending on the permutation size, to the recent GB cryptanalysis of [CR25]. Note that the permutation underlying the **Anemoui** hash function is defined over $\mathbb{F}_p^n$ only if $n \equiv 0 \pmod{2}$.
- **Poseidon.** POSEIDON employs r_P partial and r_F full rounds. In [GKR⁺21, Table 1], round numbers are proposed for our target parametrizations. However, substituting the relevant parameter values appears to yield results that are inconsistent with [GKR⁺21, Proposition 5.1]; moreover, other works do not seem to use them. We therefore instead rely on the script provided in [GKR⁺21, Appendix A], which has been recently updated to address a newer cryptanalysis [ABM24]. This is also consistent with those used in existing comparisons in the literature. Our POSEIDON parameters are derived using the same script.
- **Poseidon2.** The only difference between POSEIDON and POSEIDON2 are the matrices used within the full and partial rounds. The round numbers are derived in the same way as for POSEIDON. However, unlike its predecessor, its underlying permutation is not well-defined unless $n = 3$ or $n \equiv 0 \pmod{4}$.
- **Rescue-Prime.** The **Rescue** strategy was initially proposed in [AAB⁺20], and the authors proposed it with a large security margin. The **Rescue-Prime** specification paper [SAD20] reduced this margin as it was deemed too large, hence we follow this latter recommendation.

We summarize the resulting round numbers for **Arion** as well as for the other hash functions in Table 2.

Table 2: Number of rounds r required by the target Sponge hash functions to reach a security level of 128 bits over the target scalar fields, to hash ℓ input elements into 1 output element, with rate ℓ and capacity c equal to the smallest integer such that $c \geq 1$ and a permutation of size $n = \ell + c$ is well-defined.

ℓ	Arion		GRIFFIN		Anemoui		Rescue-Prime		POSEIDON			POSEIDON2		
	n	r	n	r	n	r	n	r	n	r_F	r_P	n	r_F	r_P
2	3	11	3	19	4	18	3	14	3	8	56	3	8	56
4	5	9	8	20	6	13	5	9	5	8	56	8	8	57
8	9	8	12	20	10	11	9	8	9	8	57	12	8	57

4.2 Optimizing Arion Arithmetization Complexity

An important step towards constructing an efficient arithmetization, is to exploit equivalent representations of the underlying arithmetic circuit to reduce the number of constraints. This observation has been used in the past to improve the arithmetization complexity of **Rescue**, **GRIFFIN** and **Anemoui**. By definition of the **Arion** non-linear layer (Definition 1), at some point one has to verify that

$$y = x^e = x^{1/d_2} \quad (9)$$

for some $x, y \in \mathbb{F}_p$. Since x^{d_2} is a low-degree map, x^e will necessarily be a high degree map, and a naive circuit representation of Equation (9) would have a high multiplicative depth, which is problematic regardless of the exact arithmetization technique being used. On the other hand, from a verifier’s perspective, checking that Equation (9) holds is equivalent to verifying that

$$y^{d_2} = x \quad (10)$$

for any values of x, y . Thus, to reduce the number of multiplicative constraints, we are well-advised to employ the equivalent representation instead of the straightforward one.

In [BBC⁺23, §4], this simple equivalence was formalized through the notion of CCZ-equivalence [CCZ98]. For completeness, we provide a characterization of the **Arion** non-linear layer $\mathcal{F}$ in terms of CCZ-equivalence in Appendix E. Interestingly, the CCZ-equivalent representation of $\mathcal{F}$ is not low-degree, but due to its inductive structure, it still allows us to greatly reduce the multiplicative complexity of a verification circuit.

4.3 Plonk Performance

Plonk [GWC19] is a well-established ZK-SNARK system. In Plonk, a 2-fan-in constraint is of the form (see [GWC19, §6]):

$$(a \cdot b) \cdot q_M + a \cdot q_L + b \cdot q_R + c \cdot q_O + q_C = 0, \quad (11)$$

a and b denote the left and right input variable, c denotes the output variable and q_M, q_L, q_R, q_O and q_C denote the “selector coefficient” of the multiplication, the variables and the constant term. The 3-fan-in Plonk constraint has 3 addition gates:

$$(a \cdot b) \cdot q_M + a \cdot q_L + b \cdot q_R + c \cdot q_O + d \cdot q_F + q_C = 0, \quad (12)$$

where d is the “fourth” variable and q_F its selector coefficient.

Remark 9. Many Plonk frameworks (e.g. [Dus22, Tea22]) support user-defined *custom gates*, which can reduce the total number of constraints by replacing several simpler constraints with more costly gates. Whether this translates into an actual performance improvement depends strongly on the specific context and implementation. Consequently, inferring practical performance solely from the number of constraints can be difficult.

Next we count the number of Plonk constraints for Arion.

Lemma 10. *Let $\mathbb{F}_p$ be a finite field, let $r, n \geq 2$ be integers, and let Arion with r rounds be defined over $\mathbb{F}_p^n$. For $i = 1, 2$ denote with $d_{i,inc}$ the minimal number of multiplications to compute the univariate power permutation x^{d_i} .*

- (1) *A prover circuit needs $N_{\mathcal{F},2} = ((n-1) \cdot (d_{1,inc} + 6) + d_{2,inc} - 1)$ 2-wire and $N_{\mathcal{F},3} = ((n-1) \cdot (d_{1,inc} + 4) + d_{2,inc})$ 3-wire Plonk constraints for the non-linear layer $\mathcal{F}$.*
- (2) *A prover circuit needs $N_{\mathcal{L},2} = 3 \cdot (n-1)$ 2-wire and*

$$N_{\mathcal{L},3} = \begin{cases} n, & n = 2, 3, \\ n + \left\lceil \frac{n-3}{2} \right\rceil, & n \geq 4, \end{cases}$$

3-wire Plonk constraints for the affine layer $\mathcal{L}_c$.

Proof. For (1), we can use the CCZ-equivalence trick (see Proposition 16) to build an efficient circuit for the non-linear layer $\mathcal{F}$. For x^{d_i} , one needs $d_{i,inc}$, so we need $(n-1) \cdot d_{1,inc} + d_{2,inc}$ constraints for the univariate permutation polynomials. For $\tau_{n,n}$, one needs one constraint, and for $\tau_{i+1,n}$, where $i < n-1$, one needs two 2-wire constraints, so in total one needs $1 + 2 \cdot (n-2)$ 2-wire constraints to compute the $\tau_{i+1,n}$'s. On the other hand, for 3-wire constraints one can compute all $\tau_{i+1,n}$'s with one constraint, so $n-1$ in total. To compute:

$$\begin{aligned} g_i &= \tau^2 + \alpha_{i,1} \cdot \tau_{i+1,n} + \alpha_{i,2}, \\ h_i &= \tau^2 + \beta_i \cdot \tau_{i+1,n} \end{aligned}$$

one needs two constraints since one can build any quadratic polynomial with the 2-wire Plonk constraint, see Equation (11). To compute $x_i^{d_1} \cdot g_i + h_i$ one needs two 2-wire constraints and one 3-wire constraint. We have to do this $(n-1)$ times, hence in total we need $(n-1) \cdot d_{1,inc} + d_{2,inc} + 1 + 2 \cdot (n-2) + 4 \cdot (n-1)$ 2-wire and $(n-1) \cdot d_{1,inc} + d_{2,inc} + (n-1) + (n-1) \cdot (2+1)$ 3-wire constraints.

For (2), we build the circuit with Algorithm 1. For the first component we need to compute the sum of n elements. The remaining components can be computed from the first via two additions. So one needs $(n-1) + 2 \cdot (n-1)$ 2-wire constraints and $1 + \left\lceil \frac{n-3}{2} \right\rceil + (n-1)$ 3-wire constraints. $\square$

The number of Plonk constraints for Arion is then given by:

$$N_{\text{Arion}} = r \cdot N_{\mathcal{F}} + (r+1) \cdot N_{\mathcal{L}}. \quad (13)$$

Note that for $n \geq 4$ Algorithm 1 yields a more efficient 2-wire and 3-wire circuit than standard matrix multiplication (which would need $n \cdot (n-1)$ 2-wire and $n \cdot (1 + \left\lceil \frac{n-3}{2} \right\rceil)$ 3-wire constraints). For the other target hash functions, we compute the number of Plonk constraints as follows:

- For **Anemoui**, we follow the arithmetization described in [BBC⁺23, § 7.2] for the 3-fan-in case, while adapting the process for the 2-fan-in case.
- For **GRIFFIN**, we use the formulae provided in [GHR⁺23, § 7.3].
- For **Rescue-Prime**, we used [GHR⁺23, Table 5].
- For **POSEIDON** and **POSEIDON2**, there are several existing optimizations based on using custom degree- d polynomial gates, e.g. in [GKS23, § 8.3]. However, as we limit our analysis to standard Plonk gates, we do not consider such extensions.

For clarity, we provide the exact formulae for all primitives in Appendix G

In Table 3 we compare the theoretical number of Plonk constraints for the target hash functions. For an input rate $\ell = 2$ field elements, GRIFFIN requires the lowest number of gates, with Arion landing in second place. For higher rate, however, Arion takes the lead, often by a significant margin. In particular, for larger rates its advantage increases over all targets, with POSEIDON2 requiring up to $2\text{--}3\times$ more constraints.

Table 3: Number of 2-wire (resp. 3-wire) Plonk constraints needed for a single iteration of the target hash functions, when parametrized as per Table 2. The best result is highlighted in **bold**, and the runner-up in underline.

gate fan-in	2			3		
ℓ	2	4	8	2	4	8
Arion	<u>358</u>	516	856	<u>289</u>	393	628
Anemoui	504	<u>624</u>	<u>1100</u>	432	520	<u>990</u>
GRIFFIN	271	1084	1824	231	900	1434
POSEIDON	624	1568	5067	432	928	2727
POSEIDON2	565	1434	2166	435	1227	1791
Rescue-Prime	420	630	1584	336	<u>450</u>	1008

4.4 R1CS Performance

Estimating the number of multiplicative constraints in a R1CS circuit is luckily much more straight-forward. For Arion, we give the following result:

Lemma 11. *Let $r, n (\geq 2)$ be integers, and for $i = 1, 2$ denote with $d_{i,inc}$ the minimal number of multiplications to compute the univariate power permutation x^{d_i} over $\mathbb{F}_p$. The number of multiplicative constraints in the prover R1CS circuit for Arion over $\mathbb{F}_p^n$ with r rounds is*

$$N_{\text{Arion}} = r \cdot ((n-1) \cdot (d_{1,inc} + 2) + d_{2,inc}).$$

Proof. By Corollary 17 one needs $d_{2,inc}$ constraints in the n^{th} branch. In each of the remaining $n-1$ branches one needs $d_{1,inc}$ constraints for the power permutation, 1 constraint for the computation of g_i and h_i and 1 multiplication for the product of the power permutation and g_i . $\square$

Analogously, the number of R1CS constraints for Anemoui, GRIFFIN, POSEIDON, POSEIDON2 and Rescue-Prime (cf. [BBC⁺23, §7], [GHR⁺23, §7.2], [GKR⁺21] and [SAD20, §7.2] respectively) are given by:

$$N_{\text{Anemoui}} = \frac{r \cdot n}{2} \cdot (d_{inc} + 2), \quad (14)$$

$$N_{\text{GRIFFIN}} = 2 \cdot r \cdot (d_{inc} + n - 2), \quad (15)$$

$$N_{\text{POSEIDON2}} = N_{\text{POSEIDON}} = d_{inc} \cdot (n \cdot r_f + r_p), \quad (16)$$

$$N_{\text{Rescue-Prime}} = 2 \cdot d_{inc} \cdot n \cdot r. \quad (17)$$

In Table 4 we compare the number of R1CS constraints required by the target hash functions. For an input rate of 2 elements, GRIFFIN takes the lead, but its advantage soon vanishes for larger rates, where Anemoui offers the lowest number of constraints. Arion remains competitive, but the margin with POSEIDON shrinks as affine operations, which POSEIDON requires many more of, are not relevant in this setting⁴.

⁴However, while not affecting its size, linear combinations still contribute to increasing the density of the constraint system.

Table 4: Number of R1CS constraints needed for a single iteration of the target hash functions, when parametrized as per Table 2. The best result is highlighted in **bold**, and the runner-up in underline.

ℓ	2	4	8
Arion	209	<u>261</u>	392
Anemoi	<u>180</u>	195	275
GRIFFIN	152	360	520
POSEIDON	240	288	<u>387</u>
POSEIDON2	240	363	459
Rescue-Prime	252	270	432

4.5 AIR Performance

ZK-STARKs [BBHR18] are post-quantum zero-knowledge proof systems for computational integrity that do not require a trusted setup phase (transparency). Current STARK frameworks are based on the *Algebraic Intermediate Representation* (AIR) arithmetization technique over the *Algebraic Execution Trace* (AET) of an arithmetic circuit. The AET is a $w \times T$ matrix where each column represents the evolution over time of a register (variable) over $\mathbb{F}_p$. In the AIR, the contents of the registers $x_{1 \leq i \leq w}$ in some row l of the AET and $y_{1 \leq i \leq w}$ in some other row l' of the AET are constrained via polynomials over $\mathbb{F}_p[X_1, \dots, X_w, Y_1, \dots, Y_w]$, having maximum total degree D . The cost metric of the AIR arithmetization can then be expressed as a function which grows with $\mathcal{O}(w \cdot T \cdot D)$ [AAB⁺19]. There are also more framework-specific ways to express the AIR cost in a STARK system [BGL20], however the “area-degree product” described above is the de-facto standard metric [AAB⁺19, GKR⁺21, GHR⁺23, BBC⁺23].

A common approach to devise a compact AIR for Arithmetization-Oriented iterated permutations is to express the polynomial constraints for a full round as a relation between two consecutive rows in the AET, and then repeat for as many rounds as necessary (i.e. the i^{th} row will contain the inputs to the i^{th} round which are also the outputs of the previous round): it is however possible to introduce either intermediate variables or additional rows if this can help reduce the overall area-degree cost.

Consider the specific case of *Arion* with $n = 3$ elements, exponent $d_1 \leq 5$, exponent $d_2 = 257$, and $R = 6$ rounds: following the *naive arithmetization* described above, we will have $w = n = 3$ and $T = R = 6$; then, for $i \in \{1, \dots, 6\}$, we set the following multivariate polynomial constraints between the i^{th} row and the successive one, in order to express the $\mathcal{F}$ transformation:

$$\begin{cases} y_1 = (x_1^5 \cdot g(x_2 + y_2 + x_3 + y_3) + h(x_2 + y_2 + x_3 + y_3)), \\ y_2 = (x_2^5 \cdot g(x_3 + y_3) + h(x_3 + y_3)), \\ y_3^{257} = x_3, \end{cases}$$

where $g(x)$ is an irreducible quadratic polynomial and $h(x)$ is a quadratic polynomial (see Definition 2.1). The area-degree cost of this naive arithmetization would be $3 \cdot 6 \cdot 257 = 4626$. To reduce the cost, we introduce 3 additional registers z_1, z_2, z_3 , and replace the last equation in the previous constraint system with the following equations:

$$y_3^4 = z_1, \quad z_1^4 = z_2, \quad z_2^4 = z_3, \quad y \cdot z_3^4 = x_3. \quad (18)$$

In this way, we increased the width of the AET from 3 to 6 registers, while also reducing the maximum degree from 257 to 5, giving us a much lower area-degree product of $6 \cdot 6 \cdot 5 = 180$. For the cases of interest then, denoting with N_{Arion} the area-degree cost

of the AIR Arion constraint system, we have:

$$N_{\text{Arion}} = (n + 3) \cdot R \cdot 5.$$

In case $d_1 > 5$ or $d_2 \neq 257$, we can use the same idea to come up with a lower area-degree cost.

On the other hand, for the other target designs we have (cf. [BBC⁺23, §7.3]):

$$N_{\text{Anemoi}} = n \cdot R \cdot d_{\text{inc}}, \quad (19)$$

$$N_{\text{GRIFFIN}} = n \cdot R \cdot d_{\text{inc}}, \quad (20)$$

$$N_{\text{POSEIDON2}} = N_{\text{POSEIDON}} = n \cdot d_{\text{inc}} \cdot \left(R_f + \left\lceil \frac{R_p}{n} \right\rceil \right), \quad (21)$$

$$N_{\text{Rescue-Prime}} = n \cdot R \cdot d_{\text{inc}}. \quad (22)$$

Table 5: Area-degree cost for the AIR arithmetization needed for a single iteration of the target hash functions, when parametrized as per Table 2. The best result is highlighted in **bold**, and the runner-up in underline.

ℓ	2	4	8
Arion	330	<u>360</u>	<u>480</u>
Anemoi	360	390	550
GRIFFIN	<u>285</u>	800	1200
POSEIDON	405	500	675
POSEIDON2	405	640	780
Rescue-Prime	210	225	360

In Table 5 we compare the optimized AIR system of Arion with the other target hash functions. We can see how the overhead introduced by using additional registers, depending only on the value of d_2 , is amortized when employing larger state sizes. **Rescue-Prime**, which was specifically designed for AIR arithmetization, performs the best, with Arion usually being the closest design, solidifying its position as a versatile hash function which performs well over different arithmetization techniques.

4.6 Native Performance Evaluation

In order to provide a fair comparison of the native performance of the target hash functions, we implemented them over the same finite field arithmetic backend. All benchmarks were executed on a machine equipped with an Intel Core i9-13900KF CPU, 64 GB of DDR5-4800 RAM, and running a CachyOS 24.12 operating system with its stock kernel. The code was compiled with the Intel DPC++ compiler with optimizations and native instruction dispatch enabled. During benchmarking, OS load was kept to a minimum, the governor was set to ‘performance’, hyper-threading and clock boosting technologies were disabled, and the benchmark process was attached to the P0 core.

Our results, reported in Table 6, show that POSEIDON and POSEIDON2 are the fastest hash function to evaluate, with both benefiting from only using low degree maps, and the latter having a linear performance scaling due to its lightweight affine layer. On the other hand, Arion’s performance is ahead by a huge margin among the designs using high-degree maps: even when only considering two-to-one compression, it is between $\approx 2\text{--}7\times$ faster than Anemoi, GRIFFIN and Rescue-Prime.

Interestingly, the lower number of rounds required to reach the target security level at larger input sizes, allows Arion to partially compensate for the overhead of absorbing more

field elements. Its excellent scaling allows Arion to surpass POSEIDON at larger state sizes, and to become more competitive with POSEIDON2.

Table 6: Native performance of a single iteration of the target hash functions, when parametrized as per Table 2. The target field was set to the scalar field of the BN254 elliptic curve. Results were obtained by taking the normalized average of 100 batches consisting of 100 executions each. Standard deviation was negligible and omitted for clarity. The best result is highlighted in **bold**, and the runner-up in underline.

ℓ	2	4	8
Arion	94.73 μ s	83.66 μ s	<u>86.62 μs</u>
Anemoi	395.14 μ s	428.13 μ s	618.50 μ s
GRIFFIN	209.07 μ s	270.33 μ s	329.28 μ s
POSEIDON	<u>30.28 μs</u>	<u>63.67 μs</u>	177.65 μ s
POSEIDON2	11.42 μs	31.35 μs	45.53 μs
Rescue-Prime	350.61 μ s	380.70 μ s	626.12 μ s

Overall Efficiency Considerations. Taken together, our comparisons highlight Arion as a versatile design. Across all the considered arithmetization techniques, it is consistently among the most competitive constructions. On the native side, while it does not reach the throughput of low-degree round designs like POSEIDON, it is still considerably faster than other ‘low-degree equivalent’ designs. This can be a worthwhile trade-off in real workloads: while native efficiency is important, it is rarely the dominant factor in applications that motivate AO hashing, especially when considering that proof generation runtime grows as $\mathcal{O}(N \log N)$ in the number of constraints [ABRT24]. The small native overhead is thus quickly amortized, whereas Arion’s constraint-count advantages translate directly into faster proofs as the circuit grows, making it an attractive all-around choice for ZK-heavy deployments.

Acknowledgments

The authors thank all reviewers for their suggestions towards improving the quality of the paper. Stefano Trevisani has been supported by the Austrian Science Fund (FWF) under project number 10.55776/F85.

References

- [AAB⁺19] Abdelrahman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. Cryptology ePrint Archive, Report 2019/426, 2019. URL: <https://eprint.iacr.org/2019/426>.
- [AAB⁺20] Abdelrahman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symm. Cryptol.*, 2020(3):1–45, 2020. doi:[10.13154/tosc.v2020.i3.1-45](https://doi.org/10.13154/tosc.v2020.i3.1-45).
- [ABM24] Tomer Ashur, Thomas Buschman, and Mohammad Mahzoun. Algebraic cryptanalysis of the HADES design strategy: Application to Poseidon and Poseidon2. In Tianqing Zhu and Yannan Li, editors, *ACISP 24, Part II*, volume 14896 of *LNCS*, pages 225–244. Springer, Singapore, July 2024. doi:[10.1007/978-981-97-5028-3_12](https://doi.org/10.1007/978-981-97-5028-3_12).

- [ABRT24] Elena Andreeva, Rishiraj Bhattacharyya, Arnab Roy, and Stefano Trevisani. On efficient and secure compression functions for arithmetization-oriented hashing. In *CSF 2024 Computer Security Foundations Symposium*, pages 1–16. IEEE Computer Society Press, July 2024. doi:[10.1109/CSF61375.2024.00045](https://doi.org/10.1109/CSF61375.2024.00045).
- [ACG⁺19] Martin R. Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 371–397. Springer, Cham, December 2019. doi:[10.1007/978-3-030-34618-8_13](https://doi.org/10.1007/978-3-030-34618-8_13).
- [AGP⁺19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, Cham, September 2019. doi:[10.1007/978-3-030-29962-0_8](https://doi.org/10.1007/978-3-030-29962-0_8).
- [AGR⁺16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219. Springer, Berlin, Heidelberg, December 2016. doi:[10.1007/978-3-662-53887-6_7](https://doi.org/10.1007/978-3-662-53887-6_7).
- [AKM⁺22] Tomer Ashur, Al Kindi, Willi Meier, Alan Szepeieniec, and Bobbin Threadbare. Rescue-Prime Optimized. Cryptology ePrint Archive, Report 2022/1577, 2022. URL: <https://eprint.iacr.org/2022/1577>.
- [AKMQ23] JP Aumasson, Dmitry Khovratovich, Bart Mennink, and Porçu Quine. SAFE: Sponge API for field elements. Cryptology ePrint Archive, Report 2023/522, 2023. URL: <https://eprint.iacr.org/2023/522>.
- [BBB⁺25a] Antoine Bak, Augustin Bariant, Aurélien Boeuf, Pierre Briaud, Morten Øygarden, and Atharva Phanse. The algebraic CheapLunch: Extending FreeLunch attacks on arithmetization-oriented primitives beyond CICO-1. *IACR Cryptol. ePrint Arch.*, 2025. URL: <https://eprint.iacr.org/2025/2040>.
- [BBB⁺25b] Augustin Bariant, Aurélien Boeuf, Pierre Briaud, Maël Hostettler, Morten Øygarden, and Håvard Raddum. Improved resultant attack against arithmetization-oriented primitives. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part V*, volume 16004 of *LNCS*, pages 335–367. Springer, Cham, August 2025. doi:[10.1007/978-3-032-01901-1_11](https://doi.org/10.1007/978-3-032-01901-1_11).
- [BBC⁺23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New design techniques for efficient arithmetization-oriented hash functions: Anemoi permutations and Jive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 507–539. Springer, Cham, August 2023. doi:[10.1007/978-3-031-38548-3_17](https://doi.org/10.1007/978-3-031-38548-3_17).
- [BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. URL: <https://eprint.iacr.org/2018/046>.

- [BBL⁺24] Augustin Bariant, Aurélien Boeuf, Axel Lemoine, Irati Manterola Ayala, Morten Øygarden, Léo Perrin, and Håvard Raddum. The algebraic FreeLunch: Efficient Gröbner basis attacks against arithmetization-oriented primitives. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IV*, volume 14923 of *LNCS*, pages 139–173. Springer, Cham, August 2024. doi:[10.1007/978-3-031-68385-5_5](https://doi.org/10.1007/978-3-031-68385-5_5).
- [BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *EUROCRYPT’99*, volume 1592 of *LNCS*, pages 12–23. Springer, Berlin, Heidelberg, May 1999. doi:[10.1007/3-540-48910-X_2](https://doi.org/10.1007/3-540-48910-X_2).
- [BCD⁺20] Tim Beyne, Anne Canteaut, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, Yu Sasaki, Yosuke Todo, and Friedrich Wiemer. Out of oddity - new cryptanalytic techniques against symmetric primitives optimized for integrity proof systems. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 299–328. Springer, Cham, August 2020. doi:[10.1007/978-3-030-56877-1_11](https://doi.org/10.1007/978-3-030-56877-1_11).
- [BCR⁺19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Cham, May 2019. doi:[10.1007/978-3-030-17653-2_4](https://doi.org/10.1007/978-3-030-17653-2_4).
- [BDFG21] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo Infinite: Proof-carrying data from additive polynomial commitments. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 649–680, Virtual Event, August 2021. Springer, Cham. doi:[10.1007/978-3-030-84242-0_23](https://doi.org/10.1007/978-3-030-84242-0_23).
- [BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. Ecrypt Hash Workshop, 2007. URL: <https://keccak.team/files/SpongeFunctions.pdf>.
- [BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the Sponge construction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, Berlin, Heidelberg, April 2008. doi:[10.1007/978-3-540-78967-3_11](https://doi.org/10.1007/978-3-540-78967-3_11).
- [BGK⁺25a] Clémence Bouvier, Lorenzo Grassi, Dmitry Khovratovich, Katharina Koschatko, Christian Rechberger, Fabian Schmid, and Markus Schofneger. Skyscraper: Fast hashing on big primes. *IACR TCHES*, 2025(2):743–780, 2025. doi:[10.46586/tches.v2025.i2.743-780](https://doi.org/10.46586/tches.v2025.i2.743-780).
- [BGK⁺25b] Clémence Bouvier, Lorenzo Grassi, Dmitry Khovratovich, Katharina Koschatko, Christian Rechberger, Fabian Schmid, and Markus Schofneger. Skyscraper-v2: Fast hashing on big primes. Cryptology ePrint Archive, Report 2025/058, 2025. URL: <https://eprint.iacr.org/2025/058>.
- [BGL20] Eli Ben-Sasson, Lior Goldberg, and David Levit. STARK friendly hash – survey and recommendation. Cryptology ePrint Archive, Report 2020/948, 2020. URL: <https://eprint.iacr.org/2020/948>.

- [BR14] Andrey Bogdanov and Vincent Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *DCC*, 70(3):369–383, 2014. doi:[10.1007/s10623-012-9697-z](https://doi.org/10.1007/s10623-012-9697-z).
- [BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, January 1991. doi:[10.1007/BF00630563](https://doi.org/10.1007/BF00630563).
- [BSV07] Thomas Baignères, Jacques Stern, and Serge Vaudenay. Linear cryptanalysis of non binary ciphers. In Carlisle M. Adams, Ali Miri, and Michael J. Wiener, editors, *SAC 2007*, volume 4876 of *LNCS*, pages 184–211. Springer, Berlin, Heidelberg, August 2007. doi:[10.1007/978-3-540-77360-3_13](https://doi.org/10.1007/978-3-540-77360-3_13).
- [BW12] Andrey Bogdanov and Meiqin Wang. Zero correlation linear cryptanalysis with reduced data complexity. In Anne Canteaut, editor, *FSE 2012*, volume 7549 of *LNCS*, pages 29–48. Springer, Berlin, Heidelberg, March 2012. doi:[10.1007/978-3-642-34047-5_3](https://doi.org/10.1007/978-3-642-34047-5_3).
- [CBD⁺20] Anne Canteaut, Tim Beyne, Itai Dinur, Maria Eichlseder, Gregor Leander, Gaëtan Leurent, María Naya Plasencia, Léo Perrin, Yu Sasaki, Yosuke Todo, and Friedrich Wiemer. Report on the security of STARK-friendly hash functions (version 2.0). Technical Report, 2020. URL: https://eips.etherbase.org/assets/eip-5988/papers/report_security_stark_friendly_hash.pdf.
- [CCZ98] Claude Carlet, Pascale Charpin, and Victor A. Zinoviev. Codes, bent functions and permutations suitable for DES-like cryptosystems. *DCC*, 15(2):125–156, 1998. doi:[10.1023/A:1008344232130](https://doi.org/10.1023/A:1008344232130).
- [CD23] David Cox and Carlos D’Andrea. Subresultants and the shape lemma. *Math. Comput.*, 92(343):2355–2379, 2023. doi:[10.1090/MCOM/3840](https://doi.org/10.1090/MCOM/3840).
- [CLO15] David A. Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer International Publishing, 4 edition, 2015. doi:[10.1007/978-3-319-16721-3](https://doi.org/10.1007/978-3-319-16721-3).
- [COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 769–793. Springer, Cham, May 2020. doi:[10.1007/978-3-030-45721-1_27](https://doi.org/10.1007/978-3-030-45721-1_27).
- [CR25] Luca Campa and Arnab Roy. Gröbner basis cryptanalysis of Anemoi. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part I*, volume 15601 of *LNCS*, pages 303–332. Springer, Cham, May 2025. doi:[10.1007/978-3-031-91107-1_11](https://doi.org/10.1007/978-3-031-91107-1_11).
- [CRST26] Luca Campa, Arnab Roy, Matthias Johann Steiner, and Stefano Trevisani. Arion: Arithmetization-oriented hashing for zero- knowledge proof systems (tosc 2026 reference implementation), August 2026. doi:[10.5281/zenodo.22095243](https://doi.org/10.5281/zenodo.22095243).
- [Dus22] Dusk Network. PLONK, 2022. Version: 0.13.1. URL: <https://github.com/dusk-network/plonk>.
- [Edg23] Ben Edgington. BLS12-381 for the rest of us, Jun 2023. URL: <https://hackmd.io/@benjaminion/bls12-381#Resources-and-further-reading>.

- [GHR⁺23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst meets fluid-SPN: Griffin for zero-knowledge applications. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 573–606. Springer, Cham, August 2023. doi:10.1007/978-3-031-38548-3_19.
- [GKL⁺22] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Reinforced Concrete: A fast hash function for verifiable computation. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 1323–1335. ACM Press, November 2022. doi:10.1145/3548606.3560686.
- [GKL⁺24] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Monolith: Circuit-friendly hash functions with new nonlinear layers for fast and constant-time implementations. *IACR Trans. Symm. Cryptol.*, 2024(3):44–83, 2024. doi:10.46586/tosc.v2024.i3.44-83.
- [GKR⁺21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 519–535. USENIX Association, August 2021. URL: <https://www.usenix.org/conference/usenixsecurity21/presentation/grassi>.
- [GKS23] Lorenzo Grassi, Dmitry Khovratovich, and Markus Schofnegger. Poseidon2: A faster version of the Poseidon hash function. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *AFRICACRYPT 23*, volume 14064 of *LNCS*, pages 177–203. Springer, Cham, July 2023. doi:10.1007/978-3-031-37679-5_8.
- [GLR⁺20] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 674–704. Springer, Cham, May 2020. doi:10.1007/978-3-030-45724-2_23.
- [Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.
- [GW20] Ariel Gabizon and Zachary J. Williamson. plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Report 2020/315, 2020. URL: <https://eprint.iacr.org/2020/315>.
- [GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. URL: <https://eprint.iacr.org/2019/953>.
- [HBHW22] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. Zcash protocol specification. Zcash Improvement Proposals Website, Sep 2022. URL: <https://zips.z.cash>.
- [HHL⁺25] Jincheol Ha, Seongha Hwang, Jooyoung Lee, Seungmin Park, and Mincheol Son. Polocolo: A ZK-friendly hash function based on S-boxes using power

- residues. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 303–332. Springer, Cham, May 2025. doi:[10.1007/978-3-031-91134-7_11](https://doi.org/10.1007/978-3-031-91134-7_11).
- [JK97] Thomas Jakobsen and Lars R. Knudsen. The interpolation attack on block ciphers. In Eli Biham, editor, *FSE'97*, volume 1267 of *LNCS*, pages 28–40. Springer, Berlin, Heidelberg, January 1997. doi:[10.1007/BFb0052332](https://doi.org/10.1007/BFb0052332).
- [JP07] Antoine Joux and Thomas Peyrin. Hash functions and the (amplified) boomerang attack. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 244–263. Springer, Berlin, Heidelberg, August 2007. doi:[10.1007/978-3-540-74143-5_14](https://doi.org/10.1007/978-3-540-74143-5_14).
- [KBM23] Dmitry Khovratovich, Mario Marhuenda Beltrán, and Bart Mennink. Generic security of the SAFE API and its applications. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 301–327. Springer, Singapore, December 2023. doi:[10.1007/978-981-99-8742-9_10](https://doi.org/10.1007/978-981-99-8742-9_10).
- [KLR24] Katharina Koschatko, Reinhard Lüftenegger, and Christian Rechberger. Exploring the six worlds of Gröbner basis cryptanalysis: Application to Anemoi. *IACR Trans. Symm. Cryptol.*, 2024(4):138–190, 2024. doi:[10.46586/tosc.v2024.i4.138-190](https://doi.org/10.46586/tosc.v2024.i4.138-190).
- [Knu95] Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *FSE'94*, volume 1008 of *LNCS*, pages 196–211. Springer, Berlin, Heidelberg, December 1995. doi:[10.1007/3-540-60590-8_16](https://doi.org/10.1007/3-540-60590-8_16).
- [KW02] Lars R. Knudsen and David Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *FSE 2002*, volume 2365 of *LNCS*, pages 112–127. Springer, Berlin, Heidelberg, February 2002. doi:[10.1007/3-540-45661-9_9](https://doi.org/10.1007/3-540-45661-9_9).
- [Lai94] Xuejia Lai. Higher order derivatives and differential cryptanalysis. In Richard E. Blahut, Daniel J. Costello, Ueli Maurer, and Thomas Mittelholzer, editors, *Communications and Cryptography: Two Sides of One Tapestry*, pages 227–233, Boston, MA, 1994. Springer US. doi:[10.1007/978-1-4615-2694-0_23](https://doi.org/10.1007/978-1-4615-2694-0_23).
- [LMR⁺09] Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer. Rebound distinguishers: Results on the full Whirlpool compression function. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 126–143. Springer, Berlin, Heidelberg, December 2009. doi:[10.1007/978-3-642-10366-7_8](https://doi.org/10.1007/978-3-642-10366-7_8).
- [Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseeth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 386–397. Springer, Berlin, Heidelberg, May 1994. doi:[10.1007/3-540-48285-7_33](https://doi.org/10.1007/3-540-48285-7_33).
- [MG26] Simon-Philipp Merz and Àlex Rodríguez García. Skipping class: Algebraic attacks exploiting weak matrices and operation modes of Poseidon2(b). *IACR Cryptol. ePrint Arch.*, 2026. URL: <https://eprint.iacr.org/2026/306>.
- [MRST09] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 260–276. Springer, Berlin, Heidelberg, February 2009. doi:[10.1007/978-3-642-03317-9_16](https://doi.org/10.1007/978-3-642-03317-9_16).

- [MW83] David W. Masser and Gisbert Wüstholz. Fields of large transcendence degree generated by values of elliptic functions. *Inventiones mathematicae*, 72(3):407–464, 1983. doi:10.1007/BF01398396.
- [NIS15a] NIST. Secure Hash Standard (SHS). National Institute of Standards and Technology, NIST FIPS PUB 180-4, U.S. Department of Commerce, August 2015. doi:10.6028/NIST.FIPS.180-4.
- [NIS15b] NIST. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. National Institute of Standards and Technology, NIST FIPS PUB 202, U.S. Department of Commerce, August 2015. doi:10.6028/NIST.FIPS.202.
- [Nyb94] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 55–64. Springer, Berlin, Heidelberg, May 1994. doi:10.1007/3-540-48285-7_6.
- [PD20] Yiannis Psaras and David Dias. The InterPlanetary File System and the Filecoin Network. In *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pages 80–80, 2020. doi:10.1109/DSN-S50200.2020.00043.
- [RAS20] Arnab Roy, Elena Andreeva, and Jan Ferdinand Sauer. Interpolation cryptanalysis of unbalanced Feistel networks with low degree round functions. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O’Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 273–300. Springer, Cham, October 2020. doi:10.1007/978-3-030-81652-0_11.
- [RS24] Arnab Roy and Matthias Johann Steiner. Generalized triangular dynamical system: An algebraic system for constructing cryptographic permutations over finite fields. In Maria Eichlseder and Sébastien Gambs, editors, *SAC 2024, Part II*, volume 15517 of *LNCS*, pages 139–165. Springer, Cham, August 2024. doi:10.1007/978-3-031-82841-6_6.
- [RST23] Arnab Roy, Matthias Johann Steiner, and Stefano Trevisani. Arion: Arithmetization-oriented permutation and hashing from generalized triangular dynamical systems. arXiv: 2303.04639, 2023. Version: 3. arXiv:2303.04639, doi:10.48550/ARXIV.2303.04639.
- [SAD20] Alan Szepieniec, Tomer Ashur, and Siemen Dhooghe. Rescue-Prime: a standard specification (SoK). Cryptology ePrint Archive, Report 2020/1143, 2020. URL: <https://eprint.iacr.org/2020/1143>.
- [Tea22] Polygon Zero Team. Plonky2: Fast recursive arguments with plonk and fri, September 2022. URL: <https://github.com/0xPolygonZero/plonky2/blob/main/plonky2/plonky2.pdf>.
- [The22] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*, 2022. <https://www.sagemath.org>.
- [Wag99] David Wagner. The boomerang attack. In Lars R. Knudsen, editor, *FSE’99*, volume 1636 of *LNCS*, pages 156–170. Springer, Berlin, Heidelberg, March 1999. doi:10.1007/3-540-48519-8_12.
- [Wan22] Jonathan Wang. BN254 for the rest of us, Aug 2022. URL: <https://hackmd.io/@jpw/bn254>.

- [YZY⁺24] Hong-Sen Yang, Qun-Xiong Zheng, Jing Yang, Quan feng Liu, and Deng Tang. A new security evaluation method based on resultant for arithmetic-oriented algorithms. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part VII*, volume 15490 of *LNCS*, pages 457–489. Springer, Singapore, December 2024. doi:[10.1007/978-981-96-0941-3_15](https://doi.org/10.1007/978-981-96-0941-3_15).
- [YZY25] Hong-Sen Yang, Qun-Xiong Zheng, and Jing Yang. Algebraic cryptanalysis of AO primitives based on polynomial decomposition - applications to rain and full AIM-I/III/V. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part I*, volume 16245 of *LNCS*, pages 383–410. Springer, Singapore, December 2025. doi:[10.1007/978-981-95-5018-0_13](https://doi.org/10.1007/978-981-95-5018-0_13).

A Generation of Arion- π Constants

In order to generate the constants for Arion- π 's affine layer, as well as the coefficients of the GTDS h_i and g_i quadratics, we employ rejection-sampling on the binary expansion of the fractional part of $\pi = 3.1415\dots$ (i.e. we ignore the leading 3). Let $b = \lceil \log_2(p) \rceil$, where p is the prime order of the field over which Arion- π is being defined. We proceed as follows:

- (1) Sample a $2b$ -bit sequence $\langle \alpha_1 \rangle \parallel \langle \alpha_2 \rangle$: if its (big-endian, unsigned) integer decoding $\alpha_1, \alpha_2 < p$, and the Legendre symbol $\left(\frac{\alpha_1^2 - 4\alpha_2}{p}\right) = -1$, then keep the pair, otherwise reject and repeat. At the end, set all $\alpha_{i,1} = \alpha_1$, and all $\alpha_{i,2} = \alpha_2$.
- (2) Sample a b -bit sequence $\langle \beta \rangle$ until $\beta < p$. Then, set all $\beta_i = \beta$.
- (3) For $1 \leq i \leq n$, sample a b -bit sequence $\langle c_i \rangle$ until $c_i < p$. At the end, set $\mathbf{c} = (c_1, \dots, c_n)$.

For our target fields, where we set $d_2 = 257$, this gives a full specification of Arion- π for any input size and any number of rounds.

In total, a software implementation will need to keep track of a total of just $n + 3$ constants, regardless of the number of rounds. Given the efficient algorithm to evaluate the swap matrix-vector product, this makes Arion- π particularly memory efficient.

B Statistical Attacks on Arion

B.1 Differential Cryptanalysis

Differential cryptanalysis [BS91] and its variants are the most widely applied attack vectors against symmetric-key ciphers and hash functions. It is based on the propagation of input differences through the rounds of a hash function. In its base form an attacker requests the hashes for large numbers of chosen messages. Then he assumes that for $r - 1$ rounds the input difference is $\Delta \mathbf{x} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$ and the output difference is $\Delta \mathbf{y} \in \mathbb{F}_q^m$. Under the assumption that the differences in the last round are fixed the attacker can then deduce how the hash function behaves. The key quantity to estimate the effectiveness of differential cryptanalysis is the so-called differential uniformity.

Definition 12 ([Nyb94]). *Let $\mathbb{F}_q$ be a finite field, and let $f : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ be a function.*

- (1) *The differential distribution table of f at $\mathbf{a} \in \mathbb{F}_q^n$ and $\mathbf{b} \in \mathbb{F}_q^m$ is defined as*

$$\delta_f(\mathbf{a}, \mathbf{b}) = |\{\mathbf{x} \in \mathbb{F}_q^n \mid f(\mathbf{x} + \mathbf{a}) - f(\mathbf{x}) = \mathbf{b}\}|.$$

- (2) *The differential uniformity of f is defined as*

$$\delta(f) = \max_{\substack{\mathbf{a} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}, \\ \mathbf{b} \in \mathbb{F}_q^m}} \delta_f(\mathbf{a}, \mathbf{b}).$$

Given the differential uniformity of a function one can upper bound the probability of differential cryptanalysis with input differences $\Delta \mathbf{x} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}$ and output differences $\Delta \mathbf{y} \in \mathbb{F}_q^m$ by

$$\mathbb{P}[f : \Delta \mathbf{x} \rightarrow \Delta \mathbf{y}] \leq \frac{\delta_f(\Delta \mathbf{x}, \Delta \mathbf{y})}{q^n} \leq \frac{\delta(f)}{q^n}. \quad (23)$$

Naturally, the lower the differential uniformity the stronger is the resistance of a block cipher against differential cryptanalysis.

By definition of the Arion system, see Section 2, we have that $d_2 \geq d_1$, so with [RS24, Corollary 18] the probability of any differential for Arion' non-linear layer $\mathcal{F} : \mathbb{F}_p^n \rightarrow \mathbb{F}_p^n$ (Definition 1) is bounded by

$$\mathbb{P}[\mathcal{F} : \Delta \mathbf{x} \rightarrow \Delta \mathbf{y}] \leq \left(\frac{d_2}{p}\right)^{\text{wt}(\Delta \mathbf{x})} \leq \frac{d_2}{p}, \quad (24)$$

where $\text{wt} : \mathbb{F}_q^n \rightarrow \mathbb{Z}$ denotes the Hamming weight, i.e. the number of non-zero entries of a vector. Recall that the DDT is preserved under affine equivalence of functions, therefore Equation (24) also applies to the Arion round function. If we assume that the differentials are independent among the rounds of Arion, then the probability of an r round distinguisher is bounded by

$$\mathbb{P}[\mathcal{R}^{(r)} \circ \dots \circ \mathcal{R}^{(1)} : \Delta \mathbf{x}_1 \rightarrow \dots \rightarrow \Delta \mathbf{x}_{r+1}] \leq \left(\frac{d_2}{p}\right)^{\sum_{i=1}^r \text{wt}(\Delta \mathbf{x}_i)} \quad (25)$$

where $\mathcal{R} := (\mathcal{L}_c \circ \mathcal{F})$.

We estimate the security level λ of Arion with r against any differential trail via the maximal probability of the differential distinguisher

$$\left(\frac{d_2}{p}\right)^r \leq 2^{-\lambda} \implies \lambda \leq r \cdot (\log_2(p) - \log_2(d_2)). \quad (26)$$

So, for $\log_2(p) \approx 256$ we conjecture that 3 rounds are sufficient to protect Arion against differential attacks.

B.2 Truncated Differential and Rebound Attacks

In a truncated differential attack [Knu95] an attacker can only predict parts of the difference between pairs of text. The Arion non-linear layer (Definition 1) admits truncated differentials with probability 1. For example $(\alpha, \dots, 0, 0)^\top \xrightarrow{\mathcal{F}} (\beta, \dots, 0, 0)^\top$ where $\alpha, \beta \in \mathbb{F}_p^\times$. Note that for any vector $\mathbf{v} \in \mathbb{F}_p^n$ with $\text{wt}(\mathbf{v}) = 1$ one has that

$$\text{wt}(\text{swap}(1, \dots, n) \mathbf{v}) = n. \quad (27)$$

Therefore, for any truncated differential of the Arion non-linear layer where only one element is active in the input and output one has that all components are active after application of the affine layer of Arion. For truncated differentials we then estimate the security level λ of Arion via

$$\left(\frac{d_2}{p}\right)^n \leq 2^{-\lambda} \implies \lambda \leq n \cdot (\log_2(p) - \log_2(d_2)). \quad (28)$$

Suppose an adversary can cover two rounds with a truncated differential of probability 1, his best bet is to search the remaining rounds for input/output differentials of weight 1. For such an adversary we estimate the security level as

$$\left(\frac{d_2}{p}\right)^{r-2} \leq 2^{-\lambda} \implies \lambda \leq (r-2) \cdot (\log_2(p) - \log_2(d_2)). \quad (29)$$

Let n and r minimal parameters satisfying Equations (28) and (29), we conjecture that $r + 2 + 2$ rounds of Arion are sufficient to protect Arion against truncated differential attacks. E.g., for $\log_2(p) \approx 256$, $n \geq 3$, and $d_2 = 257$ Arion requires at least 6 rounds to achieve security level $\lambda = 128$ against truncated differential cryptanalysis.

In a rebound attack [LMR⁺09, MRST09] one has to find two input/output pairs such that the inputs satisfy a certain (truncated) input difference and the outputs satisfy a certain (truncated) output difference. Such an attack can be split into two phases: an *inbound* and an *outbound* phase. Let $P_{\text{Arion}} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ be the target permutation, then we split it into three sub-parts $P_{\text{Arion}} = P_{fw} \circ P_{in} \circ P_{out}$. The inbound phase is placed in the middle preceded and followed by the two outbound phases. Then, in the outbound phase two high-probability (truncated) differential trails are constructed which are connected with the inbound phase. Since a truncated differential with probability 1 can only cover a single round an attacker can cover only $r - 2$ rounds with an inside-out approach. For our target primes BLS12 and BN254 one inbound round is sufficient to achieve the 128 bit security target. We conjecture that over the target primes more than 6 rounds are sufficient to nullify this attack vector.

B.3 Linear Cryptanalysis

In linear cryptanalysis [Mat94, BSV07] one tries to discover affine approximations of round functions for a sample of known input messages. The key quantity to estimate the effectiveness of linear cryptanalysis is the so-called correlation.

Definition 13. Let $\mathbb{F}_q$ be a finite field, let $n \geq 1$, let $\chi, \psi : \mathbb{F}_q^n \rightarrow \mathbb{C}$ be non-trivial additive characters, and let $F : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ be a function. The correlation for the characters (χ, ψ) of F is defined as

$$\text{CORR}_F(\chi, \psi) = \frac{1}{q^n} \cdot \sum_{\mathbf{x} \in \mathbb{F}_q^n} \overline{\chi(F(\mathbf{x}))} \cdot \psi(\mathbf{x}).$$

For Arion- π we approximate every round by affine functions, and we call the additive character tuple $\Omega = (\chi_0, \dots, \chi_r)$ a linear trail for Arion- π . It is well-known that the absolute value of the correlation is preserved under affine equivalence, therefore to bound the correlation of a single Arion- π round it suffices to bound the correlation of the Arion non-linear layer (Definition 1). By [RS24, Theorem 22] we have that

$$|\text{CORR}_{\mathcal{F}}(\chi, \psi)| \leq \begin{cases} 1, & \chi, \psi = 1, \\ \frac{d_2 - 1}{\sqrt{q}}, & \text{else.} \end{cases} \quad (30)$$

If we assume that the approximations among $r + 1$ rounds of Arion- π are independent (and non-trivial), then

$$|\text{CORR}_{\text{Arion}}(\Omega)| \leq \left(\frac{d_2 - 1}{\sqrt{p}} \right)^r. \quad (31)$$

We require for the security level λ of Arion against a linear distinguisher covering r rounds that

$$\left(\frac{d_2 - 1}{\sqrt{p}} \right)^r \leq 2^{-\lambda} \implies \lambda \leq r \left(\frac{\log_2(p)}{2} - \log_2(d_2 - 1) \right). \quad (32)$$

We conjecture that $r + 3$ rounds, where r is the minimal round number satisfying Equation (32), are sufficient to protect Arion against linear cryptanalysis. So, for $\log_2(p) \approx 256$ we require 5 rounds against linear cryptanalysis.

B.4 Other Statistical Attacks

By the definition of $\text{swap}(1, \dots, n)$ a difference in one component can affect the whole state by a single round function call. Therefore, impossible differentials [BBS99] and zero-correlation [BW12, BR14] can hardly be mounted on 4 or more rounds.

Boomerang attacks [Wag99, JP07] search for quartets that satisfy two differential paths simultaneously. For our target primes BLS12 and BN254 no differentials with high probability exist for more than 2 rounds, see Equation (25). Therefore, a boomerang attack against 4 rounds of Arion has very low probability.

C Computing The Number of Rounds for Anemoi and Griffon

Let α be the degree of the small degree power map in the non-linear layer of **Anemoi** and **GRIFFIN**. The number of rounds for **Anemoi** to reach a security level of $\lambda = 128$ bits, w.r.t. [BBB⁺25b, Section 5], is given as

$$\min r: \left\lceil \frac{\alpha + 2}{\alpha} \right\rceil^r (2\alpha - 1)^{(r-1)\frac{n}{2}+3} \geq 2^\lambda. \quad (33)$$

Regarding **GRIFFIN**, let us denote as b the number of bypassed rounds, and let d_b be the resulting variable degree after the bypass. For $t = 3, 4$, we have $b = 1$ and $d_1 = 1$; for $t = 8$, we have $b = 2$ and $d_2 = 3$; for $t \geq 12$, we have $b = 3$ and $d_3 = 6\alpha + 3$. The number of rounds for **GRIFFIN** to reach a security level of $\lambda = 128$ bits, w.r.t [BBB⁺25b, Section 5], is given as

$$\min r: d_b(2\alpha + 1)^{r-b}\alpha^{r-b+1}\alpha^2 \left\lceil \frac{2\alpha - 1}{\alpha} \right\rceil^{r-b+3} \geq 2^\lambda. \quad (34)$$

D Algebraic Attacks on Arion

D.1 Higher-Order Differential & Interpolation Attacks

Interpolation attacks [JK97] construct the polynomial vector representing the outputs of a hash function. For a hash function the interpolated polynomial vector can be exploited to set up collision or forgery attacks. The cost of interpolating a polynomial depends on the number of monomials present in the polynomial vector representing the cryptographic function. Recall that any function $F: \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ can be represented by a unique polynomial $f \in \mathbb{F}_q[\mathbf{X}_n] = \mathbb{F}_q[x_1, \dots, x_n] / (x_1^q - x_1, \dots, x_n^q - x_n)$, thus at most q^n monomials can be present in f . Clearly, if f is dense, then an interpolation attack cannot be done faster than exhaustive search.

Let $\mathcal{R}^{(i)} \subset \mathbb{F}_p[\mathbf{X}_n]$ denote the polynomial vector of the **Arion- π** round function, we say that $\mathcal{R}^{(i+1)} \circ \mathcal{R}^{(i)}$ has a degree overflow if we have to reduce with at least one of the field equations to compute the unique representation in $\mathbb{F}_p[\mathbf{X}_n]$. For the scalar fields of the BLS12-381 and BN254 curves, with $e \equiv 1/257 \pmod{p-1}$, we quickly achieve a degree overflow: for BN254, $\lceil p/e \rceil = 2$, while for BLS12-381 $\lceil p/e \rceil = 3$. Moreover, after the first round the terms

$$\left(\sum_{i=1}^n i \cdot x_i \right)^e + \sum_{i=1}^n i \cdot x_i \quad (35)$$

are present in every branch. After another application of the round function we produce the terms

$$\left(\left(\sum_{i=1}^n i \cdot x_i \right)^e + \sum_{i=1}^n i \cdot x_i \right)^e \pmod{(x_1^p - x_1, \dots, x_n^p - x_n)} \quad (36)$$

in every branch. By our specification e is the inverse exponent of a relatively low degree permutation, therefore we expect that the degrees of some of the aforementioned terms are close to $p-2$. In particular, this is the case if $e^2 \geq p$. In addition, we also expect that a big fraction of the monomials in $\mathbb{F}_p[\mathbf{X}_n]$ is present in the components of the polynomial vector

of Arion- π after at least two iterations. Although an adversary can nullify the matrix that is applied to the input vector in some attack scenarios, he cannot do so for later rounds. Therefore, we expect that at least after three rounds terms similar to Equation (36) are present in the polynomial vector of Arion- π . Further, to frustrate Meet-in-the-Middle (MitM) attacks we require that the number of rounds $r \geq 5$.

We implemented Arion in SageMath [The22] to compute the density of Arion- π for small primes. Our findings in Table 7 suggest that after two rounds Arion- π has already almost full density over $\mathbb{F}_p$.

Table 7: Observed minimum density of Arion- π for small primes, $n = 3, 4, 5$ and $d_1, d_2 = 3, 5$.

p	r	Minimum density after 2 rounds	Degree after 2 rounds	Univariate degree after 2 rounds
11	6	$\geq 82\%$	$n \cdot (p-1) - 1$	$p - 1$
13	6	$\geq 91\%$	$n \cdot (p-1) - 1$	$p - 1$
17	6	$\geq 91\%$	$n \cdot (p-1) - 1$	$p - 1$
19	6	$\geq 92\%$	$n \cdot (p-1) - 1$	$p - 1$
23	6	$\geq 90\%$	$n \cdot (p-1) - 1$	$p - 1$

If after two rounds the density of Arion- π is $\geq 0.8 \cdot p^n$, then for $p = 2^{60}$ and $n = 3$ already more than 2^{128} terms are present in Arion- π . Therefore, we expect that Arion resists interpolation attacks.

At worst only one input of Arion is unknown to the adversary, he then still has to factor a polynomial of degree close to p . Therefore, for our target primes BLS12 and BN254 this complexity always exceeds the 128 bit security target after two rounds.

Higher-order differential attacks [Knu95, Lai94, BCD⁺20] exploit that higher differentials will vanish at some point. Since the density of Arion- π exceeds $\geq 0.8 \cdot p^n$ after two rounds and its univariate degrees are close to p , we do not expect that higher-order differentials and distinguishers on Arion can undermine the 128 bit security target for our target primes BLS12 and BN254 after two rounds.

D.2 Integral Attacks

The notion of integral cryptanalysis was introduced by Knudsen & Wagner [KW02] and generalized to arbitrary finite fields by Beyne et al. [BCD⁺20]. It is based on the following properties of polynomial valued functions.

Proposition 14 ([BCD⁺20, Proposition 1, 2]). *Let $\mathbb{F}_q$ be a finite field, and let $F : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$ be a function.*

- (1) *Let $V \subset \mathbb{F}_q^n$ be an affine subspace of dimension k . If $\deg(F) < k \cdot (q-1)$, then*

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

- (2) *Let $G_1, \dots, G_n \subset \mathbb{F}_q^\times$ be multiplicative subgroups, and let $G = \times_{i=1}^n G_i$ their product group. If $\deg_{x_i}(F) < |G_i|$ for all $1 \leq i \leq n$, then*

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

Proof. (1) was proven in [BCD⁺20, Corollary 1], for (2) we index the terms of F by j and then rearrange the sum

$$\begin{aligned} \sum_{\mathbf{x} \in G} F(\mathbf{x}) &= \sum_{x_1 \in G_1, \dots, x_n \in G_n} \sum_{j \geq 0} a_j \cdot x_1^{k_{1,j}} \dots x_n^{k_{n,j}} \\ &= \sum_{j \geq 0} \sum_{x_1 \in G_1, \dots, x_n \in G_n} a_j \cdot x_1^{k_{1,j}} \dots x_n^{k_{n,j}}. \end{aligned}$$

For the zero term this sums to $|G| \cdot F(\mathbf{0})$, now assume that at least one $k_{i,j} > 0$, say $k_{1,j}$, then we further rearrange

$$\begin{aligned} &\sum_{x_1 \in G_1, \dots, x_n \in G_n} a_j \cdot x_1^{k_{1,j}} \dots x_n^{k_{n,j}} \\ &= \sum_{x_2 \in G_1, \dots, x_n \in G_n} \left(\sum_{x_1 \in G_1} a_j \cdot x_1^{k_{1,j}} \dots x_n^{k_{n,j}} \right). \end{aligned}$$

For the inner sum on the right-hand side we can consider $x_2, \dots, x_n$ as non-zero field elements, by assumption $k_{1,j} < |G_1|$ so by [BCD⁺20, Proposition 2] the sum vanishes. This proves the claim. $\square$

Let us investigate the capabilities of integral distinguishers on Arion- π . Without loss of generality we can ignore the application $\text{swap}(1, \dots, n)$ before the first Arion system. To minimize the degree after the first Arion system we fix all inputs except x_1 , then only the first component is non-constant in x_1 , but after application of the first affine layer all components contain the monomial $x_1^{d_1}$. With our findings from interpolation attacks we expect that two additional rounds of Arion- π are sufficient to increase the degree in x_1 of all components to at least $p - 2$.

Therefore, for our target fields BLS12 and BN254 we do not expect that integral attacks can invalidate our 128 bit security target after five or more rounds of Arion- π .

E Arion CCZ-equivalent Polynomial System

Definition 15. Let $\mathbb{F}_p$ be a finite field, and let $F, G : \mathbb{F}_p^n \rightarrow \mathbb{F}_p^m$ be functions.

- (1) The graph of F is defined as $\Gamma_F = \{ (\mathbf{x}, F(\mathbf{x})) \mid \mathbf{x} \in \mathbb{F}_p^n \}$.
- (2) F and G are said to be CCZ-equivalent if there exists an affine permutation $\mathcal{A}$ of $\mathbb{F}_p^n \times \mathbb{F}_p^m$ such that $\Gamma_F = \mathcal{A}(\Gamma_G)$.

Now we describe a non-linear layer that is CCZ-equivalent to the Arion one.

Proposition 16. Let $\mathbb{F}_p$ be a prime field, and let $n, d_1, d_2, e \in \mathbb{Z}_{>1}$ be integers satisfying the three conditions as in Definition 1. Let $\mathcal{F} = (f_1, \dots, f_n)$ be the Arion non-linear layer (Definition 1), let $g_i, h_i \in \mathbb{F}_p^\times$ be the polynomials that define $\mathcal{F}$, and let $\mathcal{G} = (\hat{f}_1, \dots, \hat{f}_n)$ be defined as

$$\begin{aligned} \hat{f}_i(x_1, \dots, x_n) &= x_i^{d_1} \cdot g_i(\tau_{i+1,n}) + h_i(\tau_{i+1,n}), \\ \hat{f}_n(x_1, \dots, x_n) &= x_n^{d_2}, \end{aligned}$$

where $1 \leq i \leq n - 1$ and

$$\tau_{i+1} = \sum_{j=i+1}^n x_j + \hat{f}_j(x_1, \dots, x_n).$$

Then $\mathcal{F}$ is CCZ-equivalent to $\mathcal{G}$.

Proof. We consider the affine permutation $\mathcal{A} : \mathbb{F}_p^{2n} \rightarrow \mathbb{F}_p^{2n}$ that swaps the n^{th} element with the $(2n)^{\text{th}}$ element, moreover we consider the substitution $\mathbf{x} = (\hat{x}_1, \dots, \hat{x}_{n-1}, \hat{x}_n^{d_2})$. Now we apply the affine permutation to $\Gamma_{\mathcal{F}}$ which yields:

$$\mathcal{A}(\mathbf{x}, \mathcal{F}(\mathbf{x})) = \begin{pmatrix} (\hat{x}_i)_{1 \leq i \leq n-1} \\ \hat{x}_n^{e \cdot d_2} \\ (f_i(\hat{x}_1, \dots, \hat{x}_n^{d_2}))_{1 \leq i \leq n-1} \\ \hat{x}_n^{d_2} \end{pmatrix}.$$

By construction of d_2 and e , we have that $x^{e \cdot d_2} = x$ for every $x \in \mathbb{F}_p$. Let's now investigate what happens to the f_i 's. Recall the definition of σ_i from Definition 1; starting with f_{n-1} , we have that

$$\sigma_n = \hat{x}_n^{d_2} + \hat{x}_n^{e \cdot d_2} = \hat{x}_n + \hat{x}_n^{d_2} = \tau_n,$$

for all $\hat{x}_n \in \mathbb{F}_p$, and therefore

$$f_{n-1}(\hat{x}_{n-1}, \hat{x}_n^{d_2}) = \hat{f}_{n-1}(\hat{x}_{n-1}, \hat{x}_n).$$

Inductively, we go through all the branches and conclude that $f_i(\hat{x}_1, \dots, \hat{x}_n^{d_2}) = \hat{f}_i(\hat{x}_1, \dots, \hat{x}_n)$ for all $1 \leq i \leq n-1$ which proves that $\mathcal{A}(\mathbf{x}, \mathcal{F}(\mathbf{x})) = (\hat{\mathbf{x}}, \mathcal{G}(\hat{\mathbf{x}}))$, and henceforth CCZ-equivalence. $\square$

Corollary 17. *Verifying that $(y_1, \dots, y_n) = \mathcal{F}(x_1, \dots, x_n)$ is equivalent to verifying that $(y_1, \dots, y_{n-1}, x_n) = \mathcal{G}(x_1, \dots, x_{n-1}, y_n)$.*

Note that it follows from [RS24, Theorem 17, 22] that the $\mathcal{F}$ and the CCZ-equivalent function $\mathcal{G}$ from Proposition 16 are in the same security class with respect to differential and linear cryptanalysis. Unlike in the case of **Anemoi**, the CCZ-equivalent function $\mathcal{G}$ is not a low degree function (as a whole), though when implementing it as prover circuit we never have to use multiplications to compute τ_{i+1} , as we can rely on its recursive nature.

F Diagrammatic Depiction

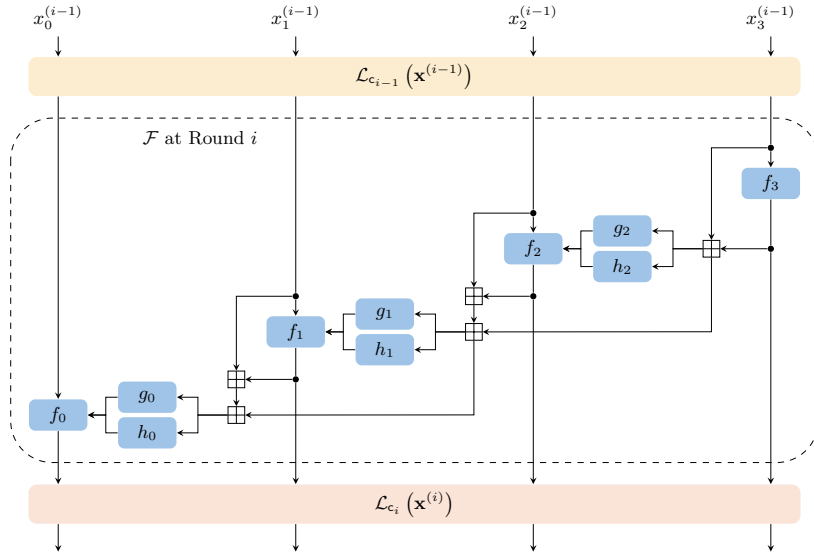


Figure 1: A diagrammatic depiction of the round function in **Arion-π** for $n = 4$

G Formulae For Plonk Constraints

Table 8: Formulae used to derive the Plonk constraint numbers for **Anemoi** [BBC⁺23, §7.2], **GRIFFIN** [GHR⁺23], **POSEIDON** [GKR⁺21], **POSEIDON2** [GKS23] and **Rescue-Prime** [SAD20].

Hash	2-wire constraints
Anemoi	$\frac{r \cdot n}{2} \cdot (d_{inc} + 7) + r \cdot \frac{n^2}{2}$
GRIFFIN	$r \cdot (2 \cdot d_{inc} + 4 \cdot n - 9) + \begin{cases} (r+1) \cdot 5, & n = 3, \\ 7 \cdot r + 8, & n = 4, \\ 23 \cdot r + 24, & n = 8, \\ (r+1) \cdot \left(\frac{8 \cdot n}{4} + 2 \cdot n - 4\right) - r, & n \geq 12, \end{cases}$
POSEIDON	$d_{inc} \cdot (n \cdot r_f + r_p) + (r_f + r_p) \cdot n \cdot (n - 1)$
POSEIDON2	$d_{inc} \cdot (n \cdot r_f + r_p) + r_p \cdot (2 \cdot n - 1) + (r_f + 1) \cdot \begin{cases} 5, & n = 3, \\ 8, & n = 4, \\ 24, & n = 8 \\ \frac{8 \cdot n}{4} + 2 \cdot n - 4, & n \geq 12 \end{cases}$
Rescue-Prime	$2 \cdot n \cdot r \cdot (d_{inc} + n - 1)$
Hash	3-wire constraints
Anemoi	$\frac{r \cdot n}{2} \cdot (d_{inc} + 5) + r \cdot \begin{cases} 2, & n = 2, \\ 8, & n = 4, \\ 16, & n = 6, \\ 24, & n = 8 \\ \frac{n^2}{2}, & n \geq 10 \end{cases}$
GRIFFIN	$r \cdot (2 \cdot d_{inc} + 3 \cdot n - 6) + \begin{cases} (r+1) \cdot 3, & n = 3, \\ 5 \cdot r + 6, & n = 4, \\ 19 \cdot r + 20, & n = 8, \\ (r+1) \cdot \left(\frac{6 \cdot n}{4} + 4 \cdot \left\lfloor \frac{\frac{n}{4}-1}{2} \right\rfloor + n\right) - r, & n \geq 12. \end{cases}$
POSEIDON	$d_{inc} \cdot (n \cdot r_f + r_p) + (r_f + r_p) \cdot n \cdot \begin{cases} 1, & n = 2, 3, \\ 1 + \lceil \frac{n-3}{2} \rceil, & n \geq 4 \end{cases}$
POSEIDON2	$d_{inc} \cdot (n \cdot r_f + r_p) + r_p \cdot \begin{cases} 3 & n = 3, \\ (n + 1 + \lceil \frac{n-3}{2} \rceil), & n \geq 4 \end{cases} \\ + (r_f + 1) \cdot \begin{cases} 3, & n = 3, \\ 6, & n = 4, \\ 20, & n = 8, \\ \frac{6 \cdot n}{4} + 4 \cdot \left\lfloor \frac{\frac{n}{4}-1}{2} \right\rfloor + n, & n \geq 12 \end{cases}$
Rescue-Prime	$2 \cdot n \cdot r \cdot d_{inc} + 2 \cdot n \cdot \begin{cases} 1, & n = 2, 3, \\ 1 + \lceil \frac{n-3}{2} \rceil, & n \geq 4 \end{cases}$