RPO-M31 and XHash-M31: Efficient Hash Functions for Circle STARKs

Tomer Ashur[1] and Sundas Tariq[1,2]

[1] 3MI Labs, Leuven, Belgium, tomer.ashur@3milabs.tech
[2] COSIC KU Leuven, Leuven, Belgium, sundas.tariq@kuleuven.be

Abstract. We present two new arithmetization oriented hash functions based on RPO [Ashur, kindi, Meier, Szepieniec, Threadbare; ePrint 2022/1577] and XHash-12 [Ashur, Bhati, Kindi, Mahzoun, Perrin; ePrint 2023/1045] adapted for $p = 2^{31} - 1$ and ready to use in Circle STARKs [Habock, Levit, Papini; ePrint 2024/278].

Keywords: Mersenne prime, RPO-M31, XHash-31

1 Introduction

A Zero Knowledge Proof (ZKP) system is a common method in cryptography for a party (*i.e.,* the prover) to convince another party (*i.e.,* the verifier) in the correctness of a certain statement without revealing any secret information. One common application seen in recent years is blockchain technology, in which proof systems are used to improve a blockchain's scalability. One type of ZKP commonly used in this context are ZK-STARKs, first introduced in [BBHR18]. As a hardness assumption, ZK-STARKs employ cryptographic hash function.

The efficiency of a ZK-protocol is directly related to the algebraic complexity of the circuit it is implementing. Therefore, traditional hash functions such as SHA-2 and SHA-3 are not suitable candidates to be implemented in ZKP due to their algebraic complexity. Consequently, a new approach in symmetric-key cryptography is focused on building hash functions that are efficient in this context; such hash functions are said to be *arithmetization oriented* (AO). For this, the authors in [AAB+20] described the Marvellous design strategy for building secure and efficient hash function for ZKP applications. Later, different hash functions were built specifically for large prime fields; a few of them are: Rescue-Prime [SAD20], RPO [AKM+22], Griffin [GHR+23], Anemoi [BBC+23], Poseidon [GKR+21], and XHash [ABK+23].

Traditional STARKs [BSBHR18] require a cyclic group of smooth order in the field. This allows efficient interpolation of points using the Fast Fourier Transform (FFT) algorithm and writing constraints that involve neighboring rows. Elliptic Curve Fast Fourier Transform (ECFFT) [BCKL21, BCKL22] introduced a way to make efficient STARKs for any large finite field (*e.g.,* Goldilocks prime $p = 2^{64} - 2^{32} + 1$), by using a cyclic group of an EC.

However, small prime fields such as Mersenne prime M31 ($p = 2^{31} - 1$) is preferable in terms of efficiency both in arithmetic operations and field implementation. In [HLP24], the authors provided a new STARK construction for the complex extension of the M31 over the circle curve $x^2 + y^2 = 1$, which is as efficient as traditional STARKs and ECFFT. This calls for designing AO hash functions optimized for this field in a similar manner to what was done for Goldilocks. We pick up this challenge and adapt RPO and XHash12 to this setting, resulting in the hash functions RPO-M31 and XHash-M31, respectively.

The rest of the paper is organized as follows. Section 2 is based on preliminary information necessary for this work. The specifications and designs of both new hash functions are explained in Section 3, and finally, Section 4 is about the security rationale.

2 Background and Preliminaries

This section is based on some basic definitions and specifications of both RPO and XHash12, which play a crucial role for the sake of better understanding.

2.1 Notation

In this paper, we construct two hash functions optimized for circle STARKs. Circle STARKs have been introduced in [HLP24] with respect to a Mersenne field of size 31, *i.e.,* $\mathbb{F}_p$, where $p = 2^{31} - 1$. Throughout the paper, we sometimes use field extensions. In particular, for $p = 2^{31} - 1$, we define the cubic extension field $\mathbb{F}_{p^3}$ similar to [ABK+23].

2.2 Finite field algebra

Let $\mathbb{F}_p$ be a prime finite field with p elements. An extension field is a quotient ring *i.e.,* $\mathbb{F}_{p^n} = \mathbb{Z}_p[x]/\langle f(x)\rangle$ where $f(x) = \sum_{i=0}^{n} a_i x^i \in F[x]$ with $a_i \in \mathbb{F}_p, n \geq 1$ is an n-degree irreducible polynomial.

We say that $\alpha \in \mathbb{F}_p$ is a primitive element (or a generator) of $\mathbb{F}_p$ when all elements in the field can be written in the form $(0, \alpha, \alpha^2, \ldots, \alpha^{p-1})$. The order of any nonzero element $\beta \in \mathbb{F}_p$ is the smallest positive integer k such that $\beta^k = 1$. We denote the order of an element by $|\beta| = k$.

2.3 Roots of Unity

We say that a complex number $\gamma \in \mathbb{C}$ is an n^{th} root of unity if $\gamma^n = 1$. We denote by U_n the set of all n^{th} roots of unity and use Euler's formula to find it; *i.e.,*

$$U_n = \{e^{\frac{2k\pi\iota}{n}} | k \in \{1, \ldots, n\}\}.$$

In the sequel, we will use a generator $\alpha_{U_n} \in \mathbb{C}$ of U_n to construct an $m \times m$ MDS matrix in $\mathbb{F}_p$. To do so, we need to find the quadratic residue of different elements in $\mathbb{F}_p$ corresponding to α_{U_n}.

Let $x, y \in \mathbb{F}_p$, we say that y is a quadratic residue of x if and only if $x^2 = y$. For the convenience of readers, we give an example for the calculation of 16^{th} root of unity and quadratic residues in Subsection A.1.

2.4 The Marvellous Design Strategy

The Marvellous design strategy provides a framework to design secure and efficient hash functions for advanced cryptographic protocols. Vision and Rescue are two families introduced in [AAB+20]. Vision is specifically designed for binary fields, whereas Rescue is specifically designed for large prime fields. Both families are sponge-based; take an m-element input and iterate over N rounds to obtain the digest. Figure 1 depicts a single round of Rescue. The interested reader is referred to [AAB+20] for a full description.

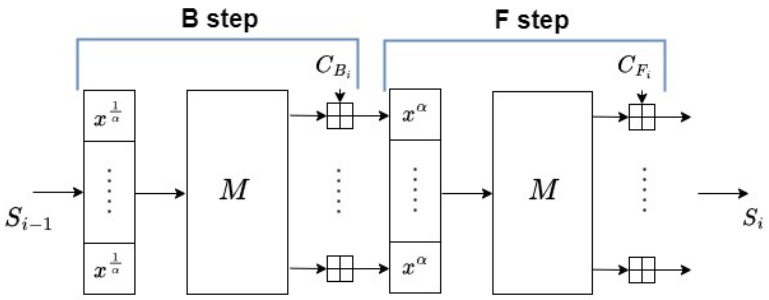


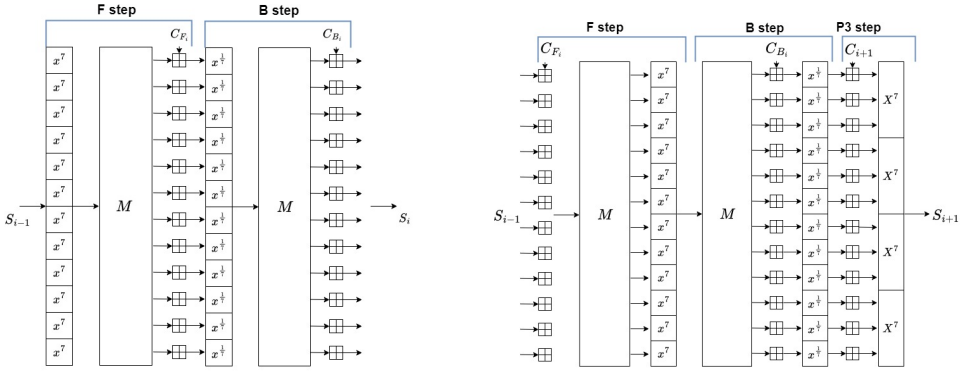
Figure 1: One round of Rescue consists of two steps (B and F).

2.4.1 Rescue Prime Optimized (RPO)

RPO [AKM+22] is an instance of Rescue, optimized for the Goldilocks prime $p = 2^{64} - 2^{32} + 1$. It is a sponge function employing the RPO permutation. For a security level of 128-bit, the permutation is defined over a state of 12 field elements and consists of seven rounds. Each round of RPO is further split into two steps, applying the x^7 power map in the even steps and the $x^{\frac{1}{7}}$ power map in the odd steps, together with a specially crafted MDS matrix and round constants. Figure 2a depicts a single round of RPO. A complete description of RPO can be found in [AKM+22].

2.4.2 XHash12

XHash12 [ABK+23] is a variant of Rescue where the RPO rounds are interleaved with extension field operations. Similar to RPO, it is a sponge function, employing a permutation of 12 field elements in the Goldilocks field. The permutation consists of six rounds, where three are standard RPO rounds with minor modifications in the order of the internal operations, and the other three rounds are power maps over a cubic extension field. After six rounds, an additional step applies once more the MDS matrix and adds round constants. Figure 2b depicts two consecutive rounds of XHash12, and the full description can be found in [ABK+23].



(a) One round of RPO

(b) Two consecutive rounds of XHash 12

Figure 2: Round function of RPO and XHash 12

3 RPO-M31 and XHash-M31

In this work, we extend RPO and XHash12 to Circle STARKs [HLP24]. Circle STARKs are built over the 31$^{\text{st}}$ Mersenne prime, *i.e.,* $2^{31} - 1$. We refer to the new functions as RPO-M31 and XHash-M31, respectively.

State (m). Whereas RPO and XHash both operate on a 12-element state, owing to the smaller field, their M31 counterparts operate on a 24-element state. We utilize the sponge construction and designate the top 16 elements as *rate (r)* and the remaining 8 elements as *capacity (c)*.

MDS. To find a corresponding MDS, we employ the methods suggested in [Hab24]. We start by constructing a circulant MDS matrix with 32 elements. Then, we unroll this matrix to its full 32×32 form and truncate the last eight columns and the last eight rows. The complete procedure can be found in Subsection A.3, where we also provide a first row of the circulant matrix.

S-box. For the S-boxes, we note that $GCD(2^{31} - 2, 5) = 1$, allowing to define the two S-boxes:

$$\begin{aligned} &\Omega_\beta : \mathbb{F}_p \to \mathbb{F}_p \\ &\Omega_\beta(x) = x^5 \,, \end{aligned} \tag{1}$$

and

$$\begin{aligned} &\Omega_{\frac{1}{\beta}} : \mathbb{F}_p \to \mathbb{F}_p \\ &\Omega_{\frac{1}{\beta}}(x) = x^{\frac{1}{5}} \,. \end{aligned} \tag{2}$$

For XHash-M31, we also recognize that $GCD((2^{31} - 1)^3 - 1, 5) = 1$ allowing to define an S-box over a cubic extension field:

$$\begin{aligned} &\Omega^3_\beta : \mathbb{F}_{p^3} \to \mathbb{F}_{p^3} \\ &\Omega^3_\beta(X) = X^5. \end{aligned} \tag{3}$$

Security level (s). In Section 4 we show that the two permutations are indistinguishable from uniformly sampled ones up to at least 2^{128} primitive calls. Then, setting the state to 24 elements with $c = 8$, the generic security of the sponge becomes the bottleneck and the hash function provides $8 \cdot \log_2(2^{31} - 1) = 124$ bits of security.

Round Constants. Based on the methodology for RPO [ABK+23] constants, we employ a deterministic way to produce round constants for both RPO-M31 and XHash-M31. First, a string corresponding to the four parameters (p, m, c, N) in ASCII decimal format is created. After formatting the string, it is passed into the SHAKE256 hash function, which generates pseudo-random bytes worth $(5 \cdot 3 \cdot N \cdot m)$. The resultant byte stream is split into 5-byte chunks, where the least significant byte appears first and each chunk is read as a base-256 integer. Next, we reduce these integers modulo p. Finally, the reduced integers make up the round constants. This method ensures that constants are generated transparently and reliably, thereby avoiding arbitrary or hidden selections. The code for generating the constants can be found in [ST24] and the round constant are given in Subsection A.2.

Number of Rounds (N). In [AAB+20, page 18], the authors explained how a safe number of rounds for a Marvellous design should be selected. According to this, the number of rounds should follow $2 \cdot \lceil \max(l_0, l_1, 5) \rceil$, where l_0 is the maximal number of rounds that can be attacked by any statistical and algebraic attacks except the Gröbner basis (GB) attack, l_1 is the maximal number of rounds that can be attacked by a GB attack, and 5 is a sanity factor. The safety margin was later relaxed in [SAD20] to be $1.5 \cdot \lceil \max(l_0, l_1, 5) \rceil$, and, seeing that no attacks were suggested over a couple of years, the number for RPO [AKM+22] was set to 7 *i.e.,* $1.5 \cdot \lfloor \max(l_0, l_1, 5) \rfloor$).

Table 1 in [AAB+20, page 20] shows different formulas to calculate the secure number of rounds against Differential Cryptanalysis (DC) and the interpolation attack. In addition, for the GB attack, a formula is given on [AAB+20, page 30]. Consequently and in line with [AKM+22, ABK+23] we decided on 7 rounds for RPO-M31 and 3 rounds for XHash-M31.

Padding. Similar to [ABK+23] we use the padding scheme suggested in [AB24]; that is, if the length of the last block is smaller than $r = 16$ field elements, a sufficient amount of [0] elements are appended to complete it. In addition, the input space is partitioned into 16 input domains: all messages whose last block is of length 16 are designated to the 0-domain; all messages whose last block is of length 15 are designated to the 1-domain, *etc.* To enforce this domain separation, the 17^{th} state element (*i.e.,* the first element of the inner part) is initialized to the domain identifier.

3.1 Specifications

We provide specifications for RPO-M31 and Xhash-M31 in Subsubsection 3.1.1 and Subsubsection 3.1.2, respectively. The polynomial representation of Ω^3_β is given in Subsection A.4.

3.1.1 RPO-M31

A round of RPO-M31 is defined in terms of two steps: a forward step (F_M) and a backward step (B_M). Each step starts by multiplying the state by the MDS matrix, followed by the addition of step constants, ending with applying Ω_β in the even steps and $\Omega_{\frac{1}{\beta}}$ in the odd steps. The round is depicted in Figure 3, and the procedure outlined in Section 3 will be used to determine the round constants. After applying seven rounds (14 steps), *i.e.,*

$$(F_M B_M)(F_M B_M)(F_M B_M)(F_M B_M)(F_M B_M)(F_M B_M)(F_M B_M)$$

the state passes through the last step in which it is multiplied once more with the MDS matrix, and an additional set of round constants is injected. The state is then truncated to its topmost 16 elements, which are returned as the hash function's digest.

Verification. In addition to the procedural description of RPO-M31 given above, we provide a polynomial description. This is useful, e.g., for ZK verification. The state after step $i : 1 \leq i \leq r$ is denoted by S_i, and its verification polynomial is:

$$(M \cdot S_{i-1} + C_{F_{M_i}})^5 - M^{-1}(S_i^5 - C_{B_{M_i}}) = 0\,. \tag{4}$$

Let S_r be the output after all rounds and S_L be the output of the linear layer. The final linear layer is verified by:

$$M \cdot S_r + C_{L_s} - S_L = 0\,. \tag{5}$$

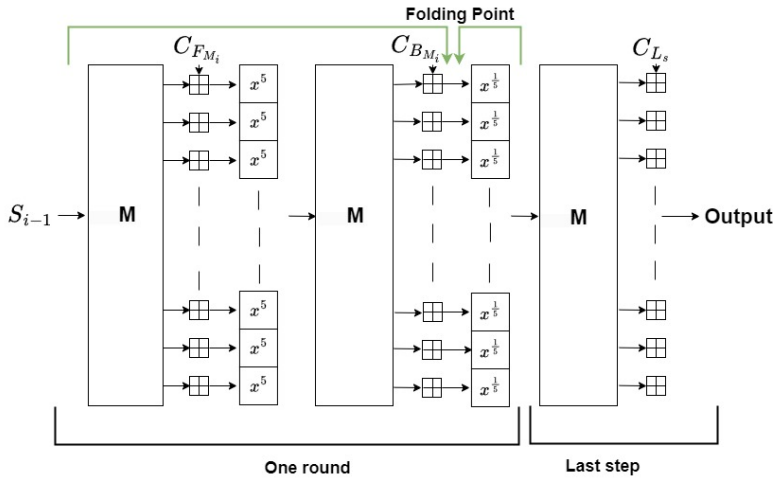


Figure 3: One round of RPO-M31 with a last step.

3.1.2 XHash-M31

A round in XHash-M31 is defined in terms of three steps: a forward step (F_M), a backward step (B_M), and an extension field $(P3_M)$ step. The (F_M) and the (B_M) steps start with multiplying the state by the MDS matrix, followed by the addition of step constants, ending with applying Ω_β in the (F_M) step and $\Omega_{\frac{1}{\beta}}$ in the (B_M) step. The $(P3_M)$ step starts by adding the set of constants and then applying Ω^3_β. The round function is depicted in Figure 4 and the procedure outlined in Section 3 will be used to determine the round constants. After applying three rounds (9 steps), i.e.,

$$(F_M B_M P3_M)(F_M B_M P3_M)(F_M B_M P3_M)$$

the state is multiplied once with an MDS matrix and a set of round constants is injected. The state is then truncated to its topmost 16 elements, which are returned as the hash function's digest.

Verification. A single round of XHash-M31 requires two types of polynomials. One type is for the $(F_M B_M)$ steps and another for the $(P3_M)$ step.

Let S_{i-1} be the round input and S_{B_i} the state after the (B_M) step, then for $1 \leq i \leq r$ the following polynomial verifies the $(F_M B_M)$ steps:

$$(M \cdot S_{i-1} + C_{F_{M_i}})^5 - M^{-1}(S^5_{B_i} - C_{B_{M_i}}) = 0\,. \tag{6}$$

The three polynomials given in Equation 8 verify the $(P3_M)$ step, depending on the position of the element in the state vector. The Modeling of the $(P3_M)$ step is described in Subsection A.4. Let S_i be the state at the end of round $1 \leq i \leq r$ and let $S_{B_i} + C_{P3_{M_i}} = \nu_i$. For $j \in \{0, 3, 6, 9, 12, 15, 18, 22\}$, the Ω^3_β S-box is described in the extension field as

$$\nu^5_{i_{[j]},[j+1],[j+2]} = (\nu_{i_{[j]}} + \nu_{i_{[j+1]}} X + \nu_{i_{[j+2]}} X^2)^5\,. \tag{7}$$

Viewed in the base field, for $0 \leq j \leq 23$, each coordinate in the extension field is described as

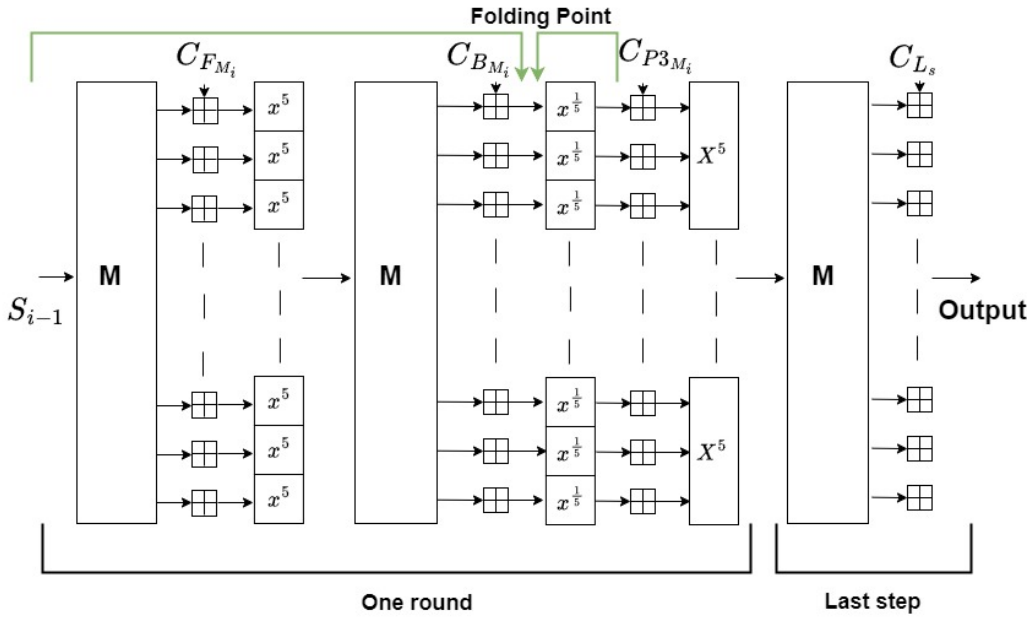


Figure 4: One round of XHash-M31 with a last step.

$$\begin{cases} \nu_{i_{[j]}}^5 - 60 \cdot \nu_{i_{[j]}}^3 \cdot \nu_{i_{[j+1]}} \cdot \nu_{i_{[j+2]}} - 20 \cdot \nu_{i_{[j]}}^2 \cdot \nu_{i_{[j+1]}}^3 \\ +40 \cdot \nu_{i_{[j]}}^2 \cdot \nu_{i_{[j+2]}}^3 + 120 \cdot \nu_{i_{[j]}} \cdot \nu_{i_{[j+1]}}^2 \cdot \nu_{i_{[j+2]}}^2 \\ +20 \cdot \nu_{i_{[j+1]}}^4 \cdot \nu_{i_{[j+2]}} - 40 \cdot \nu_{i_{[j+1]}} \cdot \nu_{i_{[j+2]}}^4 - S_{i_{[j]}} = 0 & \text{if } j \equiv 0 \pmod 3 \\ \\ 5 \cdot \nu_{i_{[j-1]}}^4 \cdot \nu_{i_{[j]}} - 20 \cdot \nu_{i_{[j-1]}}^3 \cdot \nu_{i_{[j+1]}}^2 - 60 \cdot \nu_{i_{[j-1]}}^2 \cdot \nu_{i_{[j]}}^2 \cdot \nu_{i_{[j+1]}} \\ -10 \cdot \nu_{i_{[j-1]}} \cdot \nu_{i_{[j]}}^4 + 40 \cdot \nu_{i_{[j-1]}} \cdot \nu_{i_{[j]}} \cdot \nu_{i_{[j+1]}}^3 \\ +40 \cdot \nu_{i_{[j]}}^3 \cdot \nu_{i_{[j+1]}}^2 - 8 \cdot \nu_{i_{[j+1]}}^5 - S_{i_{[j]}} = 0 & \text{if } j \equiv 1 \pmod 3 \\ \\ 5 \cdot \nu_{i_{[j-2]}}^4 \cdot \nu_{i_{[j]}} + 10 \cdot \nu_{i_{[j-2]}}^3 \cdot \nu_{i_{[j-1]}}^2 - 60 \cdot \nu_{i_{[j-2]}}^2 \cdot \nu_{i_{[j-1]}} \cdot \nu_{i_{[j]}}^2 \\ -60 \cdot \nu_{i_{[j-2]}} \cdot \nu_{i_{[j-1]}}^3 \cdot \nu_{i_{[j]}}^3 + 20 \cdot \nu_{i_{[j-2]}} \cdot \nu_{i_{[j]}}^4 \\ -2 \cdot \nu_{i_{[j-1]}}^5 + 40 \cdot \nu_{i_{[j-1]}}^2 \cdot \nu_{i_{[j]}}^3 - S_{i_{[j]}} = 0 & \text{if } j \equiv 2 \pmod 3 \end{cases} \tag{8}$$

Let S_r be the output after all rounds and S_L be the output of the linear layer. The final linear layer is verified by:

$$M \cdot S_r + C_{L_s} - S_L = 0\,. \tag{9}$$

4 Security Rationale

We provide a security argument showing resistance against known attacks. We start in Subsection 4.1 with resistance against algebraic attacks, followed by arguments for the resistance against statistical attacks, in Subsection 4.2.

4.1 Resistance to Algebraic Attacks

In this section, we argue the resistance of our new designs against algebraic attacks. We investigate two properties of the polynomial modelling: (1) polynomial degree; and (2) density. These two properties are known to be linked to the efficiency of algebraic attacks. In Subsubsection 4.1.1 - Subsubsection 4.1.2 we show that RPO-M31 and XHash-M31, have sufficiently high degrees and densities to thwart all known algebraic attacks against symmetric-key primitives.

In Subsubsection 4.1.3 we give special treatment to resistance against GB attacks. We show that the complexity of finding a GB is prohibitively high. This is done in two forms. Following [AKM+22] we experiment with toy variants to show that the complexity of finding a GB in DEGREVLEX order is too high. In addition, following [BBL+24], we show that finding a GB in a special weighted order is also infeasible (*i.e.,* there are no free lunches).

4.1.1 Polynomial Degree

We investigate two consecutive $(F_M)(B_M)$ steps. In the first step, each element of the state is raised to the 5^{th} power. Then, in the second step, an MDS matrix is applied to the state. As a result, each input to the $\Omega_{\frac{1}{\beta}}$ S-box is a linear combination of degree-5 monomials. Each such element is raised to the power $x^{\frac{1}{5}} = x^{\frac{2p-1}{5}} = x^{1717986917}$, and since it consists of a linear combination as shown in Equation 12, we expect that in general, already after one round, the output would be a polynomial of maximal degree or close to that.

$$(M \cdot S_0 + C_F)^5 = \mu, \tag{10}$$

$$(M \cdot \mu + C_B)^{\frac{1}{5}} = S_1, \tag{11}$$

$$(M \cdot (M \cdot S_0 + C_F)^5 + C_B)^{1717986917} = S_1. \tag{12}$$

Observing that both RPO-M31 and XHash-M31 start with such consecutive $(F_M)(B_M)$ steps, we conclude that their polynomial representation is of maximal degree.

4.1.2 Density

The density for RPO-M31 and XHash-M31 is calculated in two ways: (1) procedural-; and (2) non-procedural modeling.

A procedural execution is one that executes operations sequentially, *i.e.,* the output state is obtained by passing the input state sequentially through all the components that are involved in a round. Conversely, a non-procedural execution consists of a set of constraints that all must be satisfied simultaneously.

The procedural case for $(F_M)(B_M)$ steps. It follows from Equation 10 that the (F_M) step results in 98280 monomials in 24 variables due to the Multinomial Theorem. Similarly, Equation 11 describes the output after the (B_M) step, and here, we have an extremely large number of monomials again in 24 variables; *i.e.,* applying the Multinomial Theorem to the large exponent to find the number of multinomial coefficients results in $\binom{n+m-1}{m-1} = \binom{1717986917+24-1}{24-1} = \binom{1717986940}{23}$ monomials after each S-box.

Bringing these two observations together, we see in Equation 12 that following a pair of $(F_M)(B_M)$ steps there are 24 polynomials, all in the same 24 variables, each consisting of a finitely large number of monomials. This indicates that RPO-M31's polynomials are highly dense under procedural modeling. Similarly, the same holds for XHash-M31 except that in addition to this innate density, the $(P3_M)$ step further mixes triplets of polynomials in a non-linear way.

The non-procedural case after $(F_M)(B_M)$ steps. In considering a non-procedural model for RPO-M31, we use the folding outlined in Figure 3. A resulting polynomial has the form depicted in Equation 13. It follows from Equation 10 that the $M \cdot S_0$ term induces 98280 monomials in 24 variables, whereas the S_1 term adds another variable to each polynomial, or 24 variables in total. At this point, we have 24 polynomials, each consisting of $98280 + 1 = 98281$ monomials in 25 variables.

$$\begin{aligned} &M \cdot (M \cdot S_0 + C_F)^5 + C_B = (S_1)^5 \implies \\ &M \cdot (M \cdot S_0 + C_F)^5 + C_B - (S_1)^5 = 0. \end{aligned} \tag{13}$$

Concluding, we see that in the non-procedural case, RPO-M31's polynomials are dense.

Again, we observe that in addition to the $(F_M)(B_M)$ steps that are common to both algorithms, in XHash-M31, the $(P3_M)$ step adds another layer of non-linear mixing. Let S_1 be the entire state following the $(F_M)(B_M)$ steps. The $(P3_M)$ step splits the state into triplet. Without loss of generality, let $(\nu_{0,1,2})$ be one of these triplets residing in $\mathbb{F}_{p^3}$. Then, the output of Ω_β^3 when applied to $(\nu_{0,1,2})$ is a polynomial with 21 monomials in three variables. When viewed in the base field, each application of the S-box adds three polynomials, each consisting of seven monomials in three variables. The $(P3_M)$ step consists of 8 S-box applications, and therefore, it adds an overall of $8 \times 3 \times 7 = 168$ monomials in 25 variables to the polynomial system. We provide a full derivation of these polynomials in Subsection A.4.

4.1.3 Resistance to Gröbner Basis Attacks

The GB attack is an algebraic attack. The attack garners more attention when it is applied to AO primitives. Following [SS21], we view the attack as consisting of four steps: polynomial modeling, GB computation, term order change, and solution readout.

Polynomial Modeling. In the polynomial modeling step, the cryptosystem is modeled as a multivariate low-degree polynomial system. Mostly, intermediate variables are introduced to describe nonlinear operations. Different cryptosystems admit different modelings, and in fact, the same cryptosystem can be modeled in more than one way. In turn, the resulting model affects the complexity of subsequent steps. Consequently, polynomial modeling is somewhat of an art, and no general method for finding the "right" model has been developed yet.

GB Computation. The second step is concerned with finding a GB for the polynomial system devised in the previous step. Many algorithms to do this have been suggested in the literature. Early research focused on the complexity of this step to argue the security of new primitives. Motivation for this approach was provided in [AAB+20, section 6].

Term Order Change. Usually, the output of the second step is not an *elimination order*, which is needed for the fourth step. So, after the GB computation step, a term reordering algorithm is applied to convert the GB into an elimination order. Recent works showed low-complexity methods for finding non-elimination GB and suggested that arguing the security of new primitives should be done through the complexity of the term order change step. For an elaborate discussion on this, we refer the reader to [BBL+24].

Solution Readout. Once an elimination order GB is found, the last step eliminates variables iteratively to determine the variety of the polynomial system. This is done using general methods, and in general the complexity of this step is negligible compared to other steps.

Resistance Against GB Computation As we mentioned above, some previous work argued the security of new primitives via the complexity of the second step, whereas other works used the complexity of the third step. Considering that both approaches have merit, in this work we argue the security of our new designs from both perspectives such that

even if future work improves the complexity of either step, the designs would remain safe to use.

As RPO-M31 is an instance of Rescue, we can use the formula provided in [AAB+20] to calculate a secure number of rounds.

$$l_1 = \lceil \frac{s+3}{5.5 \cdot m} \rceil.$$

Setting $s = 128$ and $m = 24$ we get that $\ell_1 = \lceil \frac{128+3}{5.5 \cdot 24} \rceil = 1$; *i.e.,* the GB attack is thwarted after a single round.

Observing that RPO-M31's first round is a subset of XHash-M31's we conclude that the latter is safe after the first two steps. In support of this claim, we refer to the experiments performed in [ABK+23] on toy versions of the cipher. We conclude that for both RPO-M31 and XHASH-M31, the complexity of the GB computation step (*i.e.,* the second step of the attack) is high.

Resistance Against Term Order Change In [BBL+24] the authors argue that resistance to GB should be based on the complexity of the Term Order Change (*i.e.,* the third step in the attack). They motivate this approach by showing that in some cases, a *FreeLunch* system can be devised, *i.e.,* that the complexity of the second step is negligible.

To argue the security of both RPO-M31 and XHash-M31, we first observe that neither RPO nor XHash-12 was susceptible to FreeLunch modeling. This alone should be enough to revert to the second step as the bottleneck.

Nevertheless, following the same argumentation of [BBL+24], we see that the complexity of the third step is $(5^{24})^2 \cdot \log(5^2) \approx 2^{112}$ already after the first pair of $(F_M)(B_M)$ steps. We conclude that the complexity of term order change for RPO-M31 and XHash-M31 is too high to mount an attack following the prescribed number of rounds.

4.2 Statistical Attacks

Statistical attacks (*e.g.,* DC) have been a source of trouble in the design of traditional symmetric algorithms (*i.e.,* non-AO algorithms). Early work in the area of AO primitives observed that in this setting even a small number of rounds is enough to thwart them. Nevertheless, for completeness, we provide arguments based on the wide-trail strategy.

4.2.1 Differential Cryptanalysis

We recall the Two-Round Propagation Theorem given in [DR02]. In essence, this theorem derives an upper bound for the probability of the best differential characteristic. This is done by finding an upper bound on the differential transition probability and a lower bound for the number of active S-boxes. Then, the former is raised to the power of the latter to complete the argument.

Theorem 1 ([ABK+23]). *Let $\mathbb{F}_q$ be a finite field of order $q = p^n$ and characteristic p. Let $F(x) = x^\lambda$ be a power map defined over $\mathbb{F}_q$, then F is differentially $(\lambda - 1)$-uniform.*

In [ABK+23, Thm. 6.1], the authors proved this theorem. According to this, all power maps use in this work are differentially 4-uniform.

Analysis of the $(F_M)(B_M)$ steps against DC. In the $(F_M)(B_M)$ steps, we have two S-boxes Ω_β and $\Omega_{\frac{1}{\beta}}$ that are defined on the same domain and range and a 24×24 MDS. Further, considering the structure of the sponge construction, the adversary can access *rate* (*i.e.,* 16) elements. Moreover, the adversary can change the domain separation identifier,

which is the 17^{th} element. Therefore, according to the Two-Round Propagation Theorem, 25 S-boxes should be activated after both steps. [1]

In [AAB+20, equation 5], the authors provided a formula to calculate the upper bound probability for differential characteristics. In this case, the upper bound probability is 2^{-29}. Therefore, for the 128-bit security level, the probability of the optimal differential transition is upper bound by $2^{25\cdot(-29)} = 2^{-725}$ after just two steps. This shows that these two steps are enough already to resist the differential attack.

Analysis of the $(F_M)(B_M)(P3_M)$ steps against DC. In the $(F_M)(B_M)(P3_M)$ steps, S-boxes are defined on base field and extension field. Therefore, the Two-Round Propagation Theorem is not directly applicable. In a similar way, we did our analysis to find the number of active S-boxes after each layer in a round shown in Table 1.

Table 1: Active S-boxes at each stage of a round in both cases RPO-M31 and XHash-M31

Input	M_{F_M}	Ω_β	M_{B_M}	$\Omega_{\frac{1}{\beta}}$	Ω^3_β
1	24	24	1	1	1
2	23	23	2	2	≥ 1
3	22	22	3	3	≥ 1
4	21	21	4	4	≥ 1
5	20	20	5	5	≥ 1
6	19	19	6	6	≥ 1
7	18	18	7	7	≥ 1
8	17	17	8	8	≥ 1
9	16	16	9	9	≥ 2
10	15	15	10	10	≥ 2
11	14	14	11	11	≥ 2
12	13	13	12	12	≥ 2
13	12	12	13	13	≥ 2
14	11	11	14	14	≥ 2
15	10	10	15	15	≥ 2
16	9	9	16	16	≥ 2
17	8	8	17	17	≥ 3

We see that already after two steps, *i.e.,* the $(F_M)(B_M)$, at least 25 S-boxes are activated, and now at least one extension-field S-box (*i.e.,* Ω^3_β) is activated after the $(P3_M)$ step. In this case, the upper bound probability for differential characteristics is 2^{-29} for the $(F_M)(B_M)$ and 2^{-91} for the $(P3_M)$.

Therefore, for the 128-bit security level, the probability of the optimal differential transition across a triplet $(F_M)(B_M)(P3_M)$ is upper bound by $2^{25\cdot(-29)} \cdot 2^{1\cdot(-91)} = 2^{-725-91} = 2^{-816}$.

References

[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symmetric Cryptol.*, 2020(3):1–45, 2020.

[AB24] Tomer Ashur and Amit Singh Bhati. Generalized indifferentiable sponge and its application to polygon miden VM. Cryptology ePrint Archive, Paper 2024/911, 2024.

[1] Our analysis is given in Table 1.

[ABK+23] Tomer Ashur, Amit Singh Bhati, Al Kindi, Mohammad Mahzoun, and Léo Perrin. XHash: Efficient STARK-friendly hash function. Cryptology ePrint Archive, Paper 2023/1045, 2023.

[AKM+22] Tomer Ashur, Al Kindi, Willi Meier, Alan Szepieniec, and Bobbin Threadbare. Rescue-prime optimized. Cryptology ePrint Archive, Paper 2022/1577, 2022.

[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: ttanemoi permutations and ttjive compression mode. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539. Springer, 2023.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptol. ePrint Arch.*, page 46, 2018.

[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, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV*, volume 14923 of *Lecture Notes in Computer Science*, pages 139–173. Springer, 2024.

[BCKL21] Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, and David Levit. Elliptic curve fast fourier transform (ECFFT) part I: fast polynomial algorithms over all finite fields. *Electron. Colloquium Comput. Complex.*, TR21-103, 2021.

[BCKL22] Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, and David Levit. Scalable and transparent proofs over all large fields, via elliptic curves (ECFFT part II). *IACR Cryptol. ePrint Arch.*, page 1542, 2022.

[BSBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Paper 2018/046, 2018.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[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, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 573–606. Springer, 2023.

[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 D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 519–535. USENIX Association, 2021.

[Hab24] Ulrich Haböck. A note on circulant MDS matrices. Private Correspondence, 2024.

[HLP24] Ulrich Haböck, David Levit, and Shahar Papini. Circle STARKs. Cryptology ePrint Archive, Paper 2024/278, 2024.

[SAD20] Alan Szepieniec, Tomer Ashur, and Siemen Dhooghe. Rescue-prime: a standard specification (SoK). Cryptology ePrint Archive, Paper 2020/1143, 2020.

[SS21] Jan Ferdinand Sauer and Alan Szepieniec. SoK: Gröbner basis algorithms for arithmetization oriented ciphers. Cryptology ePrint Archive, Paper 2021/870, 2021.

[ST24] 3MI-Labs Sundas Tariq. https://github.com/3MI-Labs/rpo-xhash-m31, 2024.

A Appendix

A.1 Example – Deriving The 16^{th} Primitive Root Of Unity In M31

As an instructive example, we calculate the 16^{th} primitive root of unity in the M31 field.

$$\begin{aligned}
e^{\frac{2\pi}{16}\iota} &= cos(\frac{2\pi}{16}) + \iota sin(\frac{2\pi}{16}) \\
&= cos(\frac{\pi}{8}) + \iota sin(\frac{\pi}{8}) \\
&= \sqrt{\frac{1+cos(\frac{\pi}{4})}{2}} + \iota\sqrt{\frac{1-cos(\frac{\pi}{4})}{2}} \\
&= \sqrt{\frac{1+\frac{1}{\sqrt{2}}}{2}} + \iota\sqrt{\frac{1-\frac{1}{\sqrt{2}}}{2}} \\
&= \sqrt{\frac{1+\frac{\sqrt{2}}{2}}{2}} + \iota\sqrt{\frac{1-\frac{\sqrt{2}}{2}}{2}} \\
&= \frac{\sqrt{2+\sqrt{2}}}{2} + \iota\frac{\sqrt{2-\sqrt{2}}}{2} \\
&= \frac{1181536708}{2} + \iota\frac{190298901}{2}
\end{aligned}$$

Where the penultimate transition uses the fact that the quadratic residue of 2 in M31 is 65536 (*i.e.,* $(65536)^2 \bmod (2^{31}-1) = 2$). In the last transition, the quadratic residues of 65538 and -65534 are 1181536708 and 190298901, respectively. Therefore, after simplification, the 16^{th} primitive root of unity in M31 is:

$$e^{\frac{2\pi}{16}\iota} = 590768354 + \iota(1168891274).$$

A.2 Round Constants

[175084324, 307267372, 926126032, 968091831, 685157891, 2105385954, 1172337223, 442111374, 619202169, 1608687569, 401276325, 388976039, 747174524, 900395791,
455481706, 590173634, 460285180, 1470272960, 1563942345, 1398899312, 1846418244, 36836460, 1811927922, 167064228, 1142000631, 1434982414, 252489916, 1557799233, 44644995, 571347645, 2114100107, 1615612652, 1413066402, 1869125653, 860912024,

1777495020, 508781706, 1428126799, 416376872, 1627243238, 497670979, 623439811,
142579728, 1866460309, 1828857899, 943424486, 1308979384, 766782025, 715146265,
1305338204, 51159015, 1587489391, 1181321653, 450830656, 1557015574, 1309301546,
1364577641, 1865151097, 1321152955, 1856328988, 1256788027, 2070028866, 142189526,
997525660, 562303485, 1508364161, 546634038, 1855151308, 414030721, 204461777,
2076592910, 2081781835, 1924095286, 696694645, 1016992245, 329457036, 1746153094,
731129352, 1492879802, 109518645, 1506440475, 100640273, 348659820, 1975620907,
434979487, 1607949717, 477749360, 2108279807, 1872602192, 202188144, 1436324583,
352322420, 1267287222, 1759870960, 1698893287, 1409042830, 745465753, 830790775,
1303358776, 199137519, 379646508, 1448243080, 1839615190, 2001025364, 653435503,
1716903235, 1900980930, 127210052, 390073071, 1658571691, 918791597, 215732434,
1641205084, 64509353, 717938853, 1752985684, 549192987, 1277187607, 67103433,
1017165561, 1676100448, 873408598, 846744900, 590232860, 1007269234, 818506088,
397365975, 1627556280, 581304394, 1271111567, 1283713593, 1322890807, 1217915758,
52786324, 1589034864, 868215969, 1806826068, 1096088086, 1008187268, 991073257,
363192403, 156626590, 1864304269, 1797233804, 1592167597, 1870691635, 2116736597,
849359517, 146393037, 1368837761, 1242565219, 703324792, 1936640162, 314274059,
1938208790, 1311570627, 408019024, 488941726, 901158511, 975722452, 245478744,
1081502805, 346944102, 2071265418, 1734880865, 2040503850, 1001865044, 1112077629,
2127315813, 333970284, 576756426, 1421324759, 1034309564, 744510717, 961397525,
754415630, 1300959581, 581645912, 2106918608, 2110129877, 621267186, 1695312995,
609661509, 1913309112, 524300388, 1422632796, 1563605034, 1108753047, 25572393,
1300775045, 523497505, 835791762, 1768040971, 544480237, 388157441, 402171746,
794892581, 1182705154, 49930297, 129243944, 898441973, 429535588, 832127270, 546908747,
756292413, 686066810, 1544391099, 507384091, 625744584, 1233142395, 319177062,
1456409083, 1253953699, 818812123, 1094995825, 1400859865, 777162622, 1879784377,
714561023, 241470607, 1150358400, 868674491, 175364440, 1327652105, 580586163,
612569861, 956760481, 1756883627, 1101943429, 1895088739, 1933031694, 96732478,
224374088, 1381921877, 1325469080, 1889000316, 240001382, 1593956508, 1823462840,
141557806, 530616713, 1907165223, 62079523, 1848686076, 1295897728, 2114545896,
1693023412, 1742672882, 850732884, 196076552, 1821815249, 1719176962, 1889769026,
1061219948, 2025865748, 1939508721, 310849717, 457781360, 517901710, 65968871,
2142772976, 44714611, 101341948, 1253176252, 1930140759, 1799155961, 556217344,
2014050480, 1100303935, 465058938, 511760837, 919812381, 534265201, 908812575,
1963467299, 1628330602, 792487704, 1521690338, 417067392, 1577632259, 1903162100,
1543112201, 1593989960, 2130901602, 2065951024, 2100295115, 1076992712, 1775119720,
221638, 376032096, 1163943790, 1873812351, 375537131, 1957308375, 1069952773,
614213758, 1605486744, 826369864, 1502009496, 659760721, 548669432, 1335550272,
356452350, 675054034, 1993131373, 531587486, 789573063, 1293440420, 1846683787,
1973045453, 768581688, 2002516043, 176434293, 35489330, 821062873, 2030650085,
415910390, 1579605726, 908199138, 1233034527, 1269179452, 80243926, 840306158,
745618431, 209711912, 1828092214, 1665839141, 2134136721, 322680959, 423515746,
47676516, 584020357, 696915461, 1661795164, 2078707993, 584661952, 819846299,
178937197, 1077415206, 1242466361, 1672876753, 1473607717, 1726777726, 644234821,
1262513151, 899912102, 101933161, 1586394069, 727328391, 374984230, 1329273882,
1147628599, 408096360, 1131592455, 1676005029, 1755004107, 1185832027, 1350545055,
234313183, 355535981, 238746508, 813087134, 1623705969, 191472601, 1778703463,
1975849135, 868764694, 366997281, 292403184, 128442573, 1236216701, 666390358,
171867585, 699865044, 1892219207, 349907584, 178565821, 1631281908, 1869158961,
1167585085, 1751361316, 1810451152, 1722258621, 607370421, 1385259181, 852241363,
241792223, 1915594678, 747912505, 1061992187, 678259751, 883281053, 2077909673,
1171481040, 1259386471, 330158658, 899226172, 1120707192, 1291422605, 449005100,

407642177, 1598934822, 472095767, 238981193, 754266611, 724775445, 561137592, 592730614, 510534042, 1435373010, 287991094, 2130283411, 340860454, 1585669652, 179260010, 954133253, 652560445, 1406248378, 539102991, 534552243, 1189488938, 521087116, 611770989, 1646782070, 1539056906, 7599192, 1699865995, 649906382, 1682111335, 1156311902, 225372763, 1906892131, 1665305577, 761480569, 149073325, 1236756677, 293122269, 1540522128, 913784661, 683283264, 1349827183, 302210927, 499701005, 752368547, 1117252871, 1642552701, 1240405350, 1711703709, 1345225300, 36959573, 753313677, 2140985386, 933254780, 587810324, 1096977153, 695098010, 1119833811, 1167568020, 203344362, 1933501599, 525914421, 423784548, 1493400934, 1250606839, 1640607559, 878417286, 2100328023, 1685128921, 495027371, 114619005, 137126815, 1022247997, 369347858, 1069621656, 561463310, 233344007, 2079249531, 233542204, 2020769996, 373955554, 1407923718, 310196918, 308842651, 1757125438, 811364578, 1504528972, 1244302447, 1512031330, 1902963598, 1508403531, 356407202, 171711516, 1246960371, 1119845703, 867812005, 2024394375, 1233055993, 1048805681, 305973465, 575344339, 1306988127, 579259204, 1448192336, 291836854]

A.3 24 x 24 MDS matrix

We recall Theorem 2 from [Hab24]:

Theorem 2 ([Hab24]). *Let $\mathbb{F}_p$ be the Mersenne prime field with modulus $p = 2^{31} - 1$, choose message size $N = 2^n$ with $1 \leq n \leq 30$, and any non-zero λ from $\mathbb{F}_p$. Consider the systematic code over the alphabet $\mathbb{F}_p$ with word size $2 \times N$ and circulant systematic encoding matrix $A = (a_{i,j}); 0 \leq i, j \leq (N-1)$ defined by*

$$a_{i,j} = \frac{1}{N} \cdot \left[\lambda - \left[\frac{Im(\tau^{1+2(j-i)}}{1 - Re(\tau^{1+2(j-i)})} \right] \right] \tag{14}$$

where τ is the $(2 \times N)$-th primitive root of unity in the complex extension $\mathbb{C}(\mathbb{F}_p)$. Then, independent of the choice of λ, the distance of the code is at least N. In addition, if λ is such that $(\frac{1+\iota\cdot\lambda}{1-\iota\cdot\lambda})^{2\cdot N} \neq 1$, then the code has distance $(N+1)$ and thus is maximum distance separable.

Generation of MDS matrix. According to Theorem 2, the first possible circulant matrix for a 24-element state is of size 32×32. To find this matrix, we use the 64^{th} primitive root of unity in $\mathbb{C}(\mathbb{F}_p)$, which is given in Equation 15. It can be calculated as in example A.1.

$$\tau = \frac{\sqrt{2 + \sqrt{2 + \sqrt{2 + \sqrt{2}}}}}{2} + \iota \frac{\sqrt{2 - \sqrt{2 + \sqrt{2 + \sqrt{2}}}}}{2} \tag{15}$$

To represent Equation 15 in $\mathbb{C}(\mathbb{F}_p)$, first we calculate the quadratic residue of each term. After simplification, τ in $\mathbb{C}(\mathbb{F}_p)$ is $456695729 + \iota(1567857810)$. The first row of the 32×32 circulant matrix is given in Equation 16 which is generated by calculating Equation 14 using τ. However, to get 24×24 MDS matrix, we take a sub-matrix of the circulant matrix having the first 24 rows and 24 columns.

$$\begin{bmatrix} 185870542 & 2144994796 & 1696461115 & 215190769 \\ 930115258 & 766567118 & 2003379079 & 1770558586 \\ 1779722644 & 434368282 & 289154277 & 1979813463 \\ 1436360233 & 1342944808 & 63026005 & 903393155 \\ 1512525948 & 105409451 & 1072974295 & 979558870 \\ 436105640 & 2126764826 & 1981550821 & 636196459 \\ 645360517 & 412540024 & 1649351985 & 1485803845 \\ 53244687 & 719457988 & 270924307 & 82564914 \end{bmatrix} \tag{16}$$

A.4 Polynomial Representation of Ω^3_β

There are many irreducible polynomials in $\mathbb{F}_{p^3}$ allowing for different polynomial reductions. Without loss of generality, we use $X^3 + 2$. Let $\nu_{[j]} \in \mathbb{F}_p$. The Ω^3_β S-box in extension field is described in Equation 7. The Multinomial Theorem

$$\sum_{a+b+c=5} \left(\frac{5!}{a!b!c!}\right)\left(\nu_{[j]}^a \cdot (\nu_{[j+1]}X)^b \cdot (\nu_{[j+2]}X^2)^c\right)$$

is used to expand Equation 7:

$$
\begin{aligned}
\nu^5_{[j],[j+1],[j+2]} &= (\nu_{[j]} + \nu_{[j+1]}X + \nu_{[j+2]}X^2)^5 \\
&= \nu^5_{[j]} + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+1]})X + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+2]})X^2 + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+1]})X^2 \\
&\quad + (30\cdot\nu^3_{[j]}\cdot\nu_{[j+1]}\cdot\nu_{[j+2]})X^3 + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+2]})X^4 + (10\cdot\nu^2_{j}\cdot\nu^3_{[j+1]})X^3 \\
&\quad + (30\cdot\nu^2_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu_{[j+2]})X^4 + (30\cdot\nu^2_{[j]}\cdot\nu_{[j+1]}\cdot\nu^2_{[j+2]})X^5 \\
&\quad + (10\cdot\nu^2_{[j]}\cdot\nu^3_{[j+2]})X^6 + (5\cdot\nu_{[j]}\cdot\nu^4_{[j+1]})X^4 \\
&\quad + (30\cdot\nu_{[j]}\cdot\nu^3_{[j+1]}\cdot\nu_{[j+2]})X^5 + (30\cdot\nu_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu^2_{[j+2]})X^6 \\
&\quad + (10\cdot\nu_{[j]}\cdot\nu_{[j+1]}\cdot\nu^3_{[j+2]})X^7 + (5\cdot\nu_{[j]}\cdot\nu^4_{[j+2]})X^8 + (\nu^5_{[j+1]})X^5 \\
&\quad + (5\cdot\nu^4_{[j+1]}\cdot\nu_{[j+2]})X^6 + (10\cdot\nu^3_{[j+1]}\cdot\nu^2_{[j+2]})X^7 \\
&\quad + (10\cdot\nu^2_{[j+1]}\cdot\nu^3_{[j+2]})X^8 + (5\cdot\nu_{[j+1]}\cdot\nu^4_{[j+2]})X^9 + (\nu^5_{[j+2]})X^{10} \\
&= \nu^5_{[j]} + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+1]})X + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+2]})X^2 + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+1]})X^2 \\
&\quad + (30\cdot\nu^3_{[j]}\cdot\nu_{[j+1]}\cdot\nu_{[j+2]})(-2) + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+2]})(-2X) \\
&\quad + (10\cdot\nu^2_{[j]}\cdot\nu^3_{[j+1]})(-2) + (30\cdot\nu^2_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu_{[j+2]})(-2X) \\
&\quad + (30\cdot\nu^2_{[j]}\cdot\nu_{[j+1]}\cdot\nu^2_{[j+2]})(-2X^2) + (10\cdot\nu^2_{[j]}\cdot\nu^3_{[j+2]})(4) \\
&\quad + (5\cdot\nu_{[j]}\cdot\nu^4_{[j+1]})(-2X) + (30\cdot\nu_{[j]}\cdot\nu^3_{[j+1]}\cdot\nu_{[j+2]})(-2X^2) \\
&\quad + (30\cdot\nu_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu^2_{[j+2]})(4) + (10\cdot\nu_{[j]}\cdot\nu_{[j+1]}\cdot\nu^3_{[j+2]})(4X) \\
&\quad + (5\cdot\nu_{[j]}\cdot\nu^4_{[j+2]})(4X^2) + (\nu^5_{[j+1]})(-2X^2) + (5\cdot\nu^4_{[j+1]}\cdot\nu_{[j+2]})(4) \\
&\quad + (10\cdot\nu^3_{[j+1]}\cdot\nu^2_{[j+2]})(4X) + (10\cdot\nu^2_{[j+1]}\cdot\nu^3_{[j+2]})(4X^2) \\
&\quad + (5\cdot\nu_{[j+1]}\cdot\nu^4_{[j+2]})(-8) + (\nu^5_{[j+2]})(-8X) \\
&= \nu^5_{[j]} + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+1]})X + (5\cdot\nu^4_{[j]}\cdot\nu_{[j+2]})X^2 + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+1]})X^2 \\
&\quad - (60\cdot\nu^3_{[j]}\cdot\nu_{[j+1]}\cdot\nu_{[j+2]}) - (20\cdot\nu^3_{[j]}\cdot\nu^2_{[j+2]})X - (20\cdot\nu^2_{[j]}\cdot\nu_{j+1}[j+1]^3) \\
&\quad - (60\cdot\nu^2_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu_{[j+2]})X - (60\cdot\nu^2_{[j]}\cdot\nu_{[j+1]}\cdot\nu^2_{[j+2]})X^2 \\
&\quad + (40\cdot\nu^2_{[j]}\cdot\nu^3_{[j+2]}) - (10\cdot\nu_{[j]}\cdot\nu^4_{[j+1]})X - (60\cdot\nu_{[j]}\cdot\nu^3_{[j+1]}\cdot\nu_{[j+2]})X^2 \\
&\quad + (120\cdot\nu_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu^2_{[j+2]}) + (40\cdot\nu_{[j]}\cdot\nu_{[j+1]}\cdot\nu^3_{[j+2]})X \\
&\quad + (20\cdot\nu_{[j]}\cdot\nu^4_{[j+2]})X^2 - (2\cdot\nu^5_{[j+1]})X^2 + (20\cdot\nu^4_{[j+1]}\cdot\nu_{[j+2]}) \\
&\quad + (40\cdot\nu^3_{[j+1]}\cdot\nu^2_{[j+2]})X + (40\cdot\nu^2_{[j+1]}\cdot\nu^3_{[j+2]})X^2 \\
&\quad - (40\cdot\nu_{[j+1]}\cdot\nu^4_{[j+2]}) - (8\cdot\nu^5_{[j+2]})X \\
\nu^5_{[j],[j+1],[j+2]} &= [\nu^5_{[j]} - 60\cdot\nu^3_{[j]}\cdot\nu_{[j+1]}\cdot\nu_{[j+2]} - 20\cdot\nu^2_{[j]}\cdot\nu^3_{[j+1]} + 40\cdot\nu^2_{[j]}\cdot\nu^3_{[j+2]} \\
&\quad + 120\cdot\nu_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu^2_{[j+2]} + 20\cdot\nu^4_{[j+1]}\cdot\nu_{[j+2]} - 40\cdot\nu_{[j+1]}\cdot\nu^4_{[j+2]}] \\
&\quad + [(5\cdot\nu^4_{[j]}\cdot\nu_{[j+1]})X - (20\cdot\nu^3_{[j]}\cdot\nu^2_{[j+2]})X - (60\cdot\nu^2_{[j]}\cdot\nu^2_{[j+1]}\cdot\nu_{[j+2]})X \\
&\quad - (10\cdot\nu_{[j]}\cdot\nu^4_{[j+1]})X + (40\cdot\nu_{[j]}\cdot\nu_{[j+1]}\cdot\nu^3_{[j+2]})X \\
&\quad + (40\cdot\nu^3_{[j+1]}\cdot\nu^2_{[j+2]})X - (8\cdot\nu^5_{[j+2]})X] + [(5\cdot\nu^4_{[j]}\cdot\nu_{[j+2]})X^2 \\
&\quad + (10\cdot\nu^3_{[j]}\cdot\nu^2_{[j+1]})X^2 - (60\cdot\nu^2_{[j]}\cdot\nu_{[j+1]}\cdot\nu^2_{[j+2]})X^2 \\
&\quad - (60\cdot\nu_{[j]}\cdot\nu^3_{[j+1]}\cdot\nu_{[j+2]})X^2 + (20\cdot\nu_{[j]}\cdot\nu^4_{[j+2]})X^2 - (2\cdot\nu^5_{[j+1]})X^2 \\
&\quad + (40\cdot\nu^2_{[j+1]}\cdot\nu^3_{[j+2]})X^2]
\end{aligned}
$$

After simplification, a concise description of the first, second, and third elements of each tuple is given in Equation (17)–(19), respectively.

$$S_{[j]} = \nu_{[j]}^5 - 60 \cdot \nu_{[j]}^3 \cdot \nu_{[j+1]} \cdot \nu_{[j+2]} - 20 \cdot \nu_{[j]}^2 \cdot \nu_{[j+1]}^3 + 40 \cdot \nu_{[j]}^2 \cdot \nu_{[j+2]}^3 + 120 \cdot \nu_{[j]} \cdot \nu_{[j+1]}^2 \cdot \nu_{[j+2]}^2 + 20 \cdot \nu_{[j+1]}^4 \cdot \nu_{[j+2]} - 40 \cdot \nu_{[j+1]} \cdot \nu_{[j+2]}^4 \quad (17)$$

$$S_{[j]} = 5 \cdot \nu_{[j-1]}^4 \cdot \nu_{[j]} - 20 \cdot \nu_{[j-1]}^3 \cdot \nu_{[j+1]}^2 - 60 \cdot \nu_{[j-1]}^2 \cdot \nu_{[j]}^2 \cdot \nu_{[j+1]} - 10 \cdot \nu_{[j-1]} \cdot \nu_{[j]}^4 + 40 \cdot \nu_{[j-1]} \cdot \nu_{[j]} \cdot \nu_{[j+1]}^3 + 40 \cdot \nu_{[j]}^3 \cdot \nu_{[j+1]}^2 - 8 \cdot \nu_{[j+1]}^5 \quad (18)$$

$$S_{[j]} = 5 \cdot \nu_{[j-2]}^4 \cdot \nu_{[j]} + 10 \cdot \nu_{[j-2]}^3 \cdot \nu_{[j-1]}^2 - 60 \cdot \nu_{[j-2]}^2 \cdot \nu_{[j-1]} \cdot \nu_{[j]}^2 - 60 \cdot \nu_{[j-2]} \cdot \nu_{[j-1]}^3 \cdot \nu_{[j]}^3 + 20 \cdot \nu_{[j-2]} \cdot \nu_{[j]}^4 - 2 \cdot \nu_{[j-1]}^5 + 40 \cdot \nu_{[j-1]}^2 \cdot \nu_{[j]}^3 \quad (19)$$