How to Attack Poseidon via Smart Subspace Restriction

Amit Singh Bhati[1], Sundas Tariq[1,2], and Tomer Ashur[1]

[1] 3MI Labs, Belgium
[2] COSIC, KU Leuven, Belgium
{amitsingh.bhati, sundas.tariq, tomer}@3milabs.tech

Abstract. Poseidon [Grassi, Khovratovich, Rechberger, Roy, and Schofnegger; USENIX'21] is an arithmetization-oriented (AO) hash function designed to be highly efficient in real-world zero-knowledge (ZK) applications. As a sponge-based construction, its algebraic security inherently relies on its underlying permutation, Poseidon-π, and the dense mixing of its layers: the initial full rounds (R_{f_0}), middle partial rounds (R_p), and final full rounds (R_{f_1}). One of the common approaches to argue security is to bound the effort of solving the *k-Constrained-Input k-Constrained-Output* (CICO-k) problem over this permutation of state size $t \geq 2k$. Recent cryptanalysis has utilized subspace restriction to mitigate algebraic degree growth in Poseidon and used it to argue improvements in CICO-k solving, though prior methods have faced strict limits on the number of rounds they can successfully bypass (more specifically, < 2) and have targeted $k \leq 2$.

In this paper, we introduce a unified mathematical framework for advanced subspace restriction in Poseidon cryptanalysis. We first present a baseline gadget; Generalized S-Box Skipping via Subspace Restriction (GSR); that deterministically absorbs $2t - 2k$ total S-boxes across: one initial full round (t S-boxes) and $t - 2k$ partial rounds (one S-box per round). This significantly reduces the polynomial degree of the Poseidon multivariate polynomial system. By restricting the subspace of the total constraints satisfying solutions, independent of the round constants and MDS matrix selection, the distinguisher expends input degrees of freedom (DoFs) to linearize the internal state transitions where the dense algebraic mixing usually occurs.

We then explain and break the first theoretical barrier that confines such subspace restriction techniques to configurations with exactly $R_{f_0} = 1$ initial full round. We formalize $(k+1)$-*wise collapsing*, a non-trivial DoF consumption framework. By parameterizing the system from an intermediate state and formulating constraints backwards, Round 1 is naturally skipped at zero cost. Building on this baseline, for $R_{f_0} \geq 2$ and under the CICO-k setting, $(k+1)$-wise collapsing non-trivially expends a portion of the available DoFs to project state variables into the nullspace of targeted inverse MDS submatrices. This forces a perfect algebraic cancellation of non-linear inverse S-boxes, effectively absorbing the second full round entirely.

We show how the residual DoFs, after applying $(k+1)$-wise collapsing, can be either used to combine GSR for partial round skipping or al-

ternatively deployed for bidirectional collapsing, *i.e.,* to skip almost one full round from the back (defeating prepended/appended linear layer defenses) or can be used to construct exactly determined zero-dimensional ideals of strict degree-$3^{1+R_{f_1}}$ for $R_{f_0} = 3$ instances.
Our framework drastically improves upon the state-of-the-art algebraic limits of Poseidon, most notably by identifying a critical gap in derived theoretical cryptanalysis bounds. We demonstrate that monolithic Macaulay matrix approximations heavily over-estimate true solving complexity. By utilizing residual DoFs to construct exactly determined zero-dimensional ideals of strict degree-3 for $R_{f_0} = 3$ instances, we empirically validate our framework by tracking the step-wise intermediate peak of Faugère's F_4 algorithm in `msolve`. Through this, we successfully extract the Gröbner basis for a mathematically intractable $(3, 8, 0)$ instance of Poseidon with KoalaBear ($t = 24$) in under 5 minutes on a standard 96-thread server.
Further, in terms of monolithic complexities, where baseline estimates shows a mathematically intractable F_4 and FGLM time complexities of $\approx 2^{560}$ and $\approx 2^{404}$ bit operations, respectively, for a $(3, 10, 3)$ CICO-1 configuration, our framework successfully isolates the system into a degree-3^4 ideal, reducing the theoretical time complexities to $\approx 2^{121}$ and $\approx 2^{71}$ bit operations, respectively. Similarly, for a $(3, 5, 3)$ CICO-2 instance, we achieve a reduction in F_4 and FGLM time complexities from $\approx 2^{540}$ and $\approx 2^{390}$, respectively, to $\approx 2^{100}$ and $\approx 2^{69}$ bit operations, respectively, thereby bringing heavily constrained Gröbner basis attacks significantly closer to practical feasibility.
We then demonstrate how deploying $(k + 1)$-wise collapsing combined with GSR significantly improves the reach of interpolation-based and resultant-based CICO attacks on configurations targeting $R_{f_0} = 2$. For the industry standard KoalaBear instance where $(R_{f_0}, R_p, R_{f_1}) = (4, 23, 4)$, our results achieve penetrations up to (2,14,4), *i.e.,* 20 rounds for CICO-1, and (2,9,4), *i.e.,* 15 rounds for CICO-2.
Finally, when applying our baseline GSR technique only to $R_{f_0} = 1$ configurations, our results further extend the attack reach to larger round-reduced versions. We successfully attack up to (1,23,4), *i.e.,* 28 out of 31 rounds for CICO-1, and (1,20,4), *i.e.,* 25 out of 31 rounds for CICO-2, overall establishing new practical limits for subspace restriction against Poseidon.

Keywords: Poseidon Hash Function · Algebraic Cryptanalysis · S-Box Skipping · $(k+1)$-wise Collapsing · Subspace Restriction · Bidirectional Skipping · CICO Problem · Interpolation · Sylvester Resultant · GSR · Gröbner Basis

1 Introduction

The rapid adoption of Zero-Knowledge Proofs (ZKPs), such as SNARKs and STARKs, has driven the design of Arithmetization-Oriented Hash Functions (AOHFs). These cryptographic primitives are specifically tailored to minimize multiplicative complexity over large prime fields, optimizing them for algebraic

circuit constraints. Poseidon [13] is arguably the most prominent AOHF, heavily deployed in protocols like Filecoin, Dusk Network, and StarkNet. Its underlying permutation operates over a state of t elements and utilizes a partial SPN (Substitution-Permutation Network) structure. It sandwiches a middle layer of R_p partial non-linear rounds (where S-box is applied only to a single element) between R_{f_0} initial and R_{f_1} final full non-linear rounds (where S-boxes are applied to all t elements).

The security of Poseidon inherently relies on the algebraic density provided by its Maximum Distance Separable (MDS) matrix and the exponential degree growth induced by its S-boxes ($\mathsf{S}(x) = x^\alpha$). Consequently, algebraic attacks, specifically those utilizing Gröbner bases to solve the Constrained-Input Constrained-Output (CICO) problem, have become the primary vector for analyzing Poseidon's security margins.

To mount efficient algebraic attacks, cryptanalysts employ *subspace restriction* to delay degree growth by constraining internal states to linear subspaces. The application of subspace restriction in Poseidon cryptanalysis was first formalized by Bariant, Bouvier, Leurent, and Perrin [6], who demonstrated that input degrees of freedom (DoFs) can be used to reparameterize the initial full rounds. However, their formulation was mathematically bounded to CICO-1 ($k = 1$), could not successfully skip the dense MDS matrix transitions of the second round, and remained structurally confined to < 2 initial full round layers. Once their parameterization entered the partial layer, the MDS matrix densely mixed the remaining unconstrained variables, causing the polynomial degree to grow exponentially over all remaining unskipped rounds.

Reevaluating the initial security analysis given by the designers of Poseidon, Ashur, Buschman, and Mahzoun [3] refined the Gröbner bases complexity bounds for both Poseidon and Poseidon2 [14]. Subsequently, Grassi, Koschatko, and Rechberger [15] refined Gröbner basis attacks using finite subspace trails to linearize some partial rounds of fixed Poseidon, Poseidon2, and Neptune instances. Bak, Bariant, Boeuf, Briaud, Øygarden, and Phanse [4] presented the CheapLunch framework for multiple-output CICO-k systems, proving that the ideal degree for Poseidon is upper bounded by $\alpha^{k(R_{f_0}+R_{f_1})+R_p}$ without round skipping.

Recent efforts have further improved algebraic attacks on round-reduced instances. Reduced-round Poseidon and Poseidon2 bounty challenges were solved using resultants and univariate root finding [5]. Additionally, Zhao, Sanso, Vitto, and Ding [21] enhanced Graeffe-based root-finding techniques over NTT-friendly fields, while Sanso and Vitto [19] proposed the degree annihilation attack, which linearizes the first two partial rounds for the round-reduced KoalaBear instance with state size $t = 16$, $\alpha = 3$, and $(R_{f_0}, R_p, R_{f_1}) = (3, 8, 3)$ in the CICO-2 setting. Most recently, Vitto [20] constructed a constrained input family alongside a round-constant-dependent MDS matrix[3] to establish a finite trail that holds the active S-box coordinate constant across partial rounds, bypassing 18 out of 28 rounds for CICO-2 on a KoalaBear instance ($t = 16$).

[3] Note that this attack does not utilize the standard fixed MDS matrix of Poseidon.

In this paper, we present a unified algebraic framework that expands the frontier of subspace restriction against Poseidon-like architectures. We first introduce GSR[4], which generalizes S-box skipping to handle arbitrary CICO-k configurations ($k \geq 1$) by explicitly targeting the S-boxes *inside* the partial rounds, neutralizing the dense algebraic mixing in the middle partial layer of the permutation. Building upon this, we formalize $(k+1)$-*wise collapsing*, a generalized technique capable of completely absorbing multiple dense full rounds. Finally, we highlight the significant gap between theoretical algebraic cryptanalysis bounds and the empirical reality of modern Gröbner basis solvers.

Our Contributions. We advance the algebraic cryptanalysis of Poseidon through the following core contributions:

- **Generalized Partial-Layer S-Box Skipping:** We formulate a generalized S-box skipping gadget GSR that deterministically absorbs $2t - 2k$ total S-boxes across; one initial full round (t S-boxes) and $t-2k$ partial rounds (one S-box per round). By shifting the geometric reference point and expending DoFs to linearize intermediate partial round S-boxes, we strictly bound the polynomial degree and variable count of the resulting algebraic ideal for generic CICO-k configurations.
- **General Framework for $(k+1)$-wise Collapsing:** We present a formal mechanism to absorb multiple initial full rounds. By parameterizing backward from an intermediate state, Round 1 is naturally skipped at zero cost. For $R_{f_0} \geq 2$, we introduce $(k+1)$-wise collapsing: a technique that consumes specific DoFs to project state variables into the nullspace of targeted submatrices of the inverse MDS matrix. This perfectly cancels non-linear inverse S-boxes (compositional inverse power maps), effectively deleting Round 2 from the algebraic system.
- **Optimized Solving Strategies for $R_{f_0} = 1$:** For this setting, we apply only GSR. For CICO-1 against Poseidon's KoalaBear instance ($t = 24$) with $(R_{f_0}, R_p, R_{f_1}) = (1, 23, 4)$ (*i.e.,* 28 out of 31 rounds), we show a simple state recovery using the Cantor-Zassenhaus algorithm. For CICO-2 on $(1, 20, 4)$ (*i.e.,* 25 out of 31 rounds), we utilize Sylvester matrix-based resultants to project the bivariate system into a single dimension to recover the state, bypassing the exponential memory bottlenecks of traditional multivariate Gröbner bases. We extend this approach to $k > 2$ via a trivial pairwise resultant sequence and provide all attacked instances with bounds in Table 1.
- **Optimized Solving Strategies for $R_{f_0} = 2$:** For this setting, we combine $(k+1)$-wise collapsing with GSR, and extend the reach of algebraic attacks on Poseidon's KoalaBear instance to mathematically stronger configurations. For CICO-1 with $(R_{f_0}, R_p, R_{f_1}) = (2, 14, 4)$ (*i.e.,* 20 out of 31 rounds), we achieve equally simple state recovery as with GSR-only using the Cantor-Zassenhaus algorithm while successfully absorbing two initial full rounds. For CICO-2 on $(2, 9, 4)$ (*i.e.,* 15 out of 31 rounds), we similarly manage to

[4] Short for **G**eneralized S-box Skipping via **S**ubspace **R**estriction

absorb the first two full rounds and utilize Sylvester matrix-based resultants to recover the state. This marks a significant advancement over standard subspace restriction limits historically bounded to only 2 full and 2 partial rounds in [19], and we detail the exact complexity bounds for these extended attacks in Table 2.
- **Bidirectional Round Skipping:** We demonstrate how after applying $(k+1)$-wise collapsing, instead of jointly applying GSR, the residual DoFs can alternatively be utilized to re-apply collapsing techniques at the output boundary. This bidirectional approach absorbs almost an entire full round at the end of the cipher while already skipping two in the front. It also fundamentally defeats the recently considered defensive countermeasure that prepend a linear MDS matrix layer to the whole permutation to thwart the initial two round skip as proposed in [14]. With bidirectional skipping, we can still skip almost two full rounds, one from front and almost one from back, while this countermeasure is in place.
- **Degree-$3^{1+R_{f_1}}$ Ideal Reductions for $R_{f_0} = 3$:** We show how the same residual DoFs can alternatively be leveraged to isolate the high-degree cascade of an unskipped initial third full round. By forcing the system into an exactly determined zero-dimensional ideal of all-but-one degree-3 equations, we bypass the dense polynomial explosion typical of full-state evaluations, allowing us to mathematically target configurations of the form $(3, x, y)$. We provide improved degrevlex Gröbner basis computation cost bounds (evaluating both the F_4 [10] and FGLM [11] stages) using $(k+1)$-wise collapsing and GSR up to $(3, 10, 3)$ for $k = 1$ and up to $(3, 5, 3)$ for $k = 2$.
- **Refuting Monolithic Gröbner Estimates:** We show the inherent inaccuracy of using classical monolithic Macaulay matrix dimensions to bound F_4 Gröbner basis complexity. We mathematically and empirically demonstrate that F_4's step-wise matrix reductions peak at an intermediate degree of regularity, resulting in matrices exponentially smaller than monolithic estimates. We provide an irrefutable empirical benchmark: solving a mathematically intractable $(3, 8, 0)$ CICO-1 configuration for Gröbner basis over Poseidon-KoalaBear in just 5 minutes on a standard 96-thread server and < 10GB of RAM.

2 Preliminaries

2.1 Formal Description of Poseidon

Let $\mathbb{F}_p$ be a prime field of size p. The Poseidon permutation [13] operates on a state vector $S \in \mathbb{F}_p^t$, where the state size $t = c + r$ is the sum of the *capacity* c and the *rate* r. The non-linear layer utilizes the S-box $\mathsf{S}(x) = x^\alpha$, where $\gcd(\alpha, p-1) = 1$. This condition guarantees that $\mathsf{S}(x)$ is a bijection with a well-defined compositional inverse $\mathsf{S}^{-1}(x) = x^{1/\alpha}$. The linear layer applies a Maximum Distance Separable (MDS) matrix $M \in \mathbb{F}_p^{t \times t}$ generated according to the secure conditions detailed in [1] and [16].

A single round function $\mathcal{R}^{(i)}$ consists of three sequential operations: the addition of round-specific constants $C^{(i)} \in \mathbb{F}_p^t$, the application of the S-box layer, and the matrix multiplication by M. The permutation is structured in three consecutive phases:

1. **Initial Full Rounds (R_{f_0}):** The S-box is applied to all t elements of the state.
2. **Partial Rounds (R_p):** The S-box is applied solely to the first state element at the index 0; all other elements pass through the identity function.
3. **Final Full Rounds (R_{f_1}):**[5] The S-box is again applied to all t elements of the state.

2.2 Degrees of Freedom and Subspace Restriction

CICO-k problem. A standard CICO-k problem requires finding an initial state $S_{in} \in \mathbb{F}_p^t$ such that some predetermined k input words and some k predetermined output words of the permutation match predefined constants (typically zero). Because k words of S_{in} are strictly constrained, the system possesses $t - k$ free variables at the input, referred to as *input degrees of freedom* (DoF).

Subspace Restriction. A generic polynomial system representing the full cipher possesses a high polynomial degree and therefore is expected to exhibit a large solution space over the algebraic closure. By actively consuming degrees of freedom, *i.e.,* fixing certain variables to strategically chosen constants (let's say, Δs), the global solution space can be artificially restricted, trading off subspace size to easily find at least one valid solution.

More specifically, fixing the input of non-linear components (*i.e.,* S-boxes) allows us to evaluate them as lower-degree (*e.g.,* linear) transformations. The objective then is to expend available DoFs to linearize partial round S-boxes (and therefore the partial rounds themselves), ensuring the restricted system is uniquely equivalent to a smaller, low-degree polynomial ideal. Hence, it becomes relatively easy to solve and find at least one valid solution. Note that iterating over multiple Δ tuples allows the cryptanalyst to efficiently generate distinct CICO-k solutions each at the same order of cost, facilitating multi-target or collision attacks.

3 The GSR Gadget: Targeting S-Boxes in the Partial Layer

We generalize round skipping into S-box skipping on two fronts: 1) we allow it to target the intermediate partial round S-boxes rather than exclusively the initial full rounds, and 2) in order to skip an S-box, we set the variable before it to a constant rather than introducing a new linear constraint among the variables of

[5] Poseidon instances in [13] are often defined with $R_{f_0} = R_{f_1}$, using a single notation R_F to denote their sum.

that round. This approach avoids dependencies on the MDS matrix and round constants, which historically limited the choices of k to very small values. Consequently, we make the skipping free of these parameters and therefore applicable to arbitrary k values. For context, existing skipping results [6, 19] have been strictly limited to $k \in \{1, 2\}$.

While previous methods [6] looked to maximize the absolute number of skipped S-boxes ($2t-k$) by skipping almost two full rounds, the baseline GSR gadget reallocates the $t-2k$ consumable DoFs to skip the first full round and $t-2k$ partial rounds, resulting in a polynomial system of much lower degree. Overall, the baseline GSR skips $2t-2k$ total S-boxes; exactly k fewer S-boxes than the theoretical bound of [6]. However, by projecting the constraints into the partial rounds, the dense algebraic mixing in the middle of the cipher is neutralized. This is where the degree growth per variable is otherwise harder to contain, thereby strictly bounding the polynomial degree of the final system.

3.1 Zero-Cost Initial Round Skipping and Parameterization Setup

Rather than parameterizing the initial state S_{in} directly, we shift our geometric reference point. Let $Y_1 \in \mathbb{F}_p^t$ be the state *immediately after* the Round 1 S-box layer.

Tracing backward, the state at the beginning of the cipher is given by the inverse S-box and constant subtraction:

$$S_{in} = \mathsf{S}^{-1}(Y_1) - C^{(1)} .$$

where $\mathsf{S}^{-1}(x) = x^{1/\alpha}$ represents the compositional inverse power map of the S-box, and $C^{(1)}$ is the first round constant vector.

To satisfy the k predefined input constraints, let say, zeros, of the CICO-k problem ($S_{in}[j] = 0$ for $j \in I_{in}$; the input constraints index set), we isolate the elements:

$$Y_1[j]^{1/\alpha} - C^{(1)}[j] = 0 \implies Y_1[j] = (C^{(1)}[j])^{\alpha} .$$

By evaluating the fully known field constant $(C^{(1)}[j])^{\alpha}$, the non-linear fractional power map corresponding to the Round 1 S-box layer is bypassed entirely. This natural reverse parameterization yields exactly k strictly linear constraints (fixing k variables of Y_1 to constants) and skips Round 1 without consuming any of the available $t-k$ input degrees of freedom.

3.2 Forward Linearization of the Partial Round S-Boxes

For instances with $R_{f_0} = 1$, the partial layer begins immediately at Round 2. We expend $t-2k$ of the remaining $t-k$ DoFs in Y_1 to trace forward and linearize the partial layer.

Let $S^{(r)}$ denote the rth partial-round's state before the addition of the round constant $C^{(r)}$ generated using a random seed in the SHAKE-256 hash function [18]. The state entering the first partial round (Round 2) is simply $S^{(1)} = M \cdot Y_1$.

For each of the first $t-2k$ partial rounds ($r \geq 2$), let $\delta^{(r)} \in \mathbb{F}_p$ be an arbitrary constant chosen by the attacker. We constrain the input to the single active S-box at index 0 to equal $\delta^{(r)}$:

$$(S^{(r)} + C^{(r)})[0] = \delta^{(r)} .$$

Consequently, the S-box output is deterministically the field element $(\delta^{(r)})^\alpha$. The transition to round $r+1$ is given by the affine mapping:

$$S^{(r+1)} = M \cdot \left(S^{(r)} + C^{(r)} - \delta^{(r)} \cdot e_0 + (\delta^{(r)})^\alpha \cdot e_0 \right)$$

where e_0 is the standard basis vector. Because the mapping relies solely on constant additions and matrix multiplications, every state $S^{(r)}$ with $2 \leq r \leq t-2k+1$ remains an affine combination of the variables in Y_1.

Enforcing this condition for $t-2k$ consecutive partial rounds generates exactly $t-2k$ **strictly linear** constraints on Y_1. In total, we consume $t-2k$ DoFs in Y_1 to linearize the forward transition, while k variables are fixed by the backward transition. This leaves exactly k variables free. Since there are exactly k output constraints to satisfy at the end of the cipher, the system becomes determined (*i.e.,* a zero-dimensional ideal). Varying the target tuple $\Delta = (\delta^{(2)}, \ldots, \delta^{(t-2k+1)})$ generates distinct solution subspaces.

4 Breaking the $R_{f_0} = 1$ Barrier: $(k+1)$-wise Collapsing

While the baseline GSR gadget effectively neutralizes partial layers, its forward parameterization inherently limits the attack to configurations where the initial full rounds are fixed to $R_{f_0} = 1$. To skip an additional full round from the front ($R_{f_0} \geq 2$), we must eliminate the non-linear degree contribution of Round 2.

4.1 Backward Linearization of Round 2

We shift our parameterization one round deeper. Let $Z \in \mathbb{F}_p^t$ be the intermediate state located immediately after the round constant addition layer of Round 3. Tracing backward, the state $Y \in \mathbb{F}_p^t$ immediately after the Round 2 S-box layer is given by:

$$Y = M^{-1}(Z - C^{(3)}) \tag{1}$$

The state transformations back to S_{in} across Round 2 and Round 1 are then:

$$S_{in} = \mathsf{S}^{-1}\left(M^{-1}\left(\mathsf{S}^{-1}(Y) - C^{(2)} \right) \right) - C^{(1)} \tag{2}$$

To satisfy k input constraints $S_{in,j} = 0$ for indices $j \in I_{in} = \{i_1, \ldots, i_k\}$, we require:

$$\left(M^{-1}\left(\mathsf{S}^{-1}(Y) - C^{(2)} \right) \right)_j = \mathsf{S}\left(C_j^{(1)} \right) = \left(C_j^{(1)} \right)^\alpha \tag{3}$$

As established in Section 3.1, this reverse formulation naturally skips Round 1. However, we must now additionally eliminate the non-linear terms of the form $Y^{1/\alpha}$ arising from the Round 2 S-box layer.

4.2 Pairwise Collapsing for CICO-1 ($k = 1$)

To skip an additional full round from the front ($R_{f_0} \geq 2$), we must eliminate the non-linear degree contribution of $\mathsf{S}^{-1}(Y)$, which contains terms of the form $Y_v^{1/\alpha}$ arising from the Round 2 S-box layer.

For $k = 1$, let the single constrained input index be $m = i_1$. Expanding the backward constraint equation yields the following.

$$\sum_{v=0}^{t-1} M_{m,v}^{-1} \cdot \left(Y_v^{1/\alpha} - C_v^{(2)} \right) = \left(C_m^{(1)} \right)^{\alpha}$$

$$\implies \sum_{v=0}^{t-1} M_{m,v}^{-1} \cdot Y_v^{1/\alpha} = \left(C_m^{(1)} \right)^{\alpha} + \sum_{v=0}^{t-1} M_{m,v}^{-1} \cdot C_v^{(2)} = \mathcal{C}_{target} \tag{4}$$

where $\mathcal{C}_{target} \in \mathbb{F}_p$ is a fully known field constant.

Equation 4 contains t non-linear terms $Y_v^{1/\alpha}$. To force these non-linear terms to cancel out seamlessly, we introduce **pairwise collapsing**. We partition $t-2$ variables of Y into $N_{pairs} = (t-2)/2$ disjoint pairs of indices $(2i, 2i+1)$ for $0 \leq i \leq t/2 - 2$.

For each pair, we establish a fixed linear ratio between their inverse power maps by defining:

$$\lambda_i = -\frac{M_{m,2i}^{-1}}{M_{m,2i+1}^{-1}} \quad \implies \quad Y_{2i+1} = (\lambda_i)^{\alpha} \cdot Y_{2i} \tag{5}$$

Substituting $Y_{2i+1}^{1/\alpha} = \lambda_i \cdot Y_{2i}^{1/\alpha}$ into the pairwise sum in Equation 4 yields a perfect cancellation:

$$M_{m,2i}^{-1} \cdot Y_{2i}^{1/\alpha} + M_{m,2i+1}^{-1} \cdot \left(-\frac{M_{m,2i}^{-1}}{M_{m,2i+1}^{-1}} Y_{2i}^{1/\alpha} \right) = 0 \tag{6}$$

Thus, the non-linear fractional terms $Y_{2i}^{1/\alpha}$ vanish identically from row m of M^{-1} for all choices of the free variables $Y_{2i} = x_i$.

By expending N_{pairs} DoFs to enforce these constant ratios, all free variables x_i disappear from Equation 4. Setting $Y_{t-2} = 0$, the final variable Y_{t-1} is determined strictly as a fixed constant:

$$Y_{t-1} = \left(\frac{\mathcal{C}_{target}}{M_{m,t-1}^{-1}} \right)^{\alpha}. \tag{7}$$

With Y fully parameterized by the free variables x_i, the intermediate state Z entering the Round 3 S-box layer is directly computed via the affine mapping $Z = M \cdot Y + C^{(3)}$. This structural cancellation absorbs the entirety of Round 2 (constants addition, S-box, and MDS matrix layer) and the constants addition layer of Round 3.

4.2.1 Alternative Interpretation

To build intuition for the generalized $(k+1)$-wise case (presented in Section 4.3), we can alternatively interpret this pairwise cancellation through the lens of linear algebra. For each $i \in \{0, \ldots, N_{pairs} - 1\}$, let $T_i = \{2i, 2i+1\}$ be a pair of state indices. We construct the 1×2 submatrix (effectively a row vector) $\mathbf{M}_{T_i}^{-1}$ formed by the single constraint row m and columns T_i of M^{-1}. Since $\mathbf{M}_{T_i}^{-1}$ is a 1×2 submatrix derived from an MDS matrix, it has full row rank 1. By the Rank-Nullity Theorem, its right nullspace has dimension exactly 1. Let $\boldsymbol{\gamma}_i = (\gamma_{i,0}, \gamma_{i,1})^\top \in \mathbb{F}_p^2$ be a non-trivial vector in $\ker(\mathbf{M}_{T_i}^{-1})$, satisfying:

$$M_{m,2i}^{-1} \cdot \gamma_{i,0} + M_{m,2i+1}^{-1} \cdot \gamma_{i,1} = 0 \tag{8}$$

Parameterizing the paired elements via a single free variable x_i as $Y_{2i} = (\gamma_{i,0})^\alpha \cdot x_i$ and $Y_{2i+1} = (\gamma_{i,1})^\alpha \cdot x_i$ ensures their inverse power maps evaluate to $Y_{2i}^{1/\alpha} = \gamma_{i,0} \cdot x_i^{1/\alpha}$ and $Y_{2i+1}^{1/\alpha} = \gamma_{i,1} \cdot x_i^{1/\alpha}$. Substituting this into the system proves the cancellation:

$$M_{m,2i}^{-1} \cdot Y_{2i}^{1/\alpha} + M_{m,2i+1}^{-1} \cdot Y_{2i+1}^{1/\alpha} = x_i^{1/\alpha} \left(M_{m,2i}^{-1} \cdot \gamma_{i,0} + M_{m,2i+1}^{-1} \cdot \gamma_{i,1} \right) = 0 \tag{9}$$

This perfectly vanishes *all-but-the-last-two* non-linear terms $x_i^{1/\alpha}$ from the homogeneous portion of the backward constraint equation. Geometrically, fixing the algebraic ratio λ_i is entirely equivalent to projecting the state pairs directly into the kernel of the target MDS matrix row. While the N_{pairs} free variables x_i span the nullspace, the non-homogeneous constant $\mathcal{C}_{target}$ requires a particular solution. The remaining two variables Y_{t-2} and Y_{t-1} are excluded from the kernel projection to fulfill this role. By expending one DoF to set $Y_{t-2} = 0$, the final variable Y_{t-1} is deterministically fixed to satisfy the affine shift.

To summarize, we consume 1 DoF per pair out of *all-but-the-last* pair (*i.e.*, $t/2-1$ pairs) to collapse 2 variables into 1 free variable and separately 2 DoFs for the last pair. This is a total of only $t/2+1$ DoFs consumed (out of the original $t-2$ DoFs) to skip 2 full rounds for CICO-1.

4.3 Generalization to $(k+1)$-wise Collapsing for CICO-k

For general CICO-k instances, the attacker must satisfy k simultaneous backward constraints:

$$\sum_{v=0}^{t-1} M_{r,v}^{-1} \cdot Y_v^{1/\alpha} = \mathcal{C}_r \quad \text{for each } r \in I_{in} = \{i_1, \ldots, i_k\} \tag{10}$$

where $\mathcal{C}_r = \left(C_r^{(1)}\right)^\alpha + \sum_{v=0}^{t-1} M_{r,v}^{-1} \cdot C_v^{(2)}$.

To achieve simultaneous cancellation across all k linear constraint equations, variables cannot be grouped in pairs. Instead, we group the state variables into tuples of size $k+1$, forming $(k+1)$**-wise collapsing** (*e.g.,* triplewise for $k=2$, quadruplewise for $k=3$).

For each $i \in \{0, \ldots, N_{tuples} - 1\}$, let $T_i = \{j_{i,0}, j_{i,1}, \ldots, j_{i,k}\}$ be a tuple of $k+1$ state indices. We construct the $k \times (k+1)$ submatrix $\mathbf{M}_{T_i}^{-1}$ formed by rows I_{in} and columns T_i of M^{-1}. Since $\mathbf{M}_{T_i}^{-1}$ is a $k \times (k+1)$ submatrix derived from an MDS matrix, it has full row rank k. By the Rank-Nullity Theorem, its right nullspace has dimension exactly 1. Let $\boldsymbol{\gamma}_i = (\gamma_{i,0}, \gamma_{i,1}, \ldots, \gamma_{i,k})^\top \in \mathbb{F}_p^{k+1}$ be a non-trivial vector in $\ker(\mathbf{M}_{T_i}^{-1})$, satisfying:

$$\sum_{s=0}^{k} M^{-1}_{r,j_{i,s}} \cdot \gamma_{i,s} = 0 \quad \text{for all } r \in I_{in} \tag{11}$$

Parameterizing the tuple elements via a single free variable x_i as $Y_{j_{i,s}} = (\gamma_{i,s})^\alpha \cdot x_i$ ensures their inverse power maps evaluate to $Y_{j_{i,s}}^{1/\alpha} = \gamma_{i,s} \cdot x_i^{1/\alpha}$. Substituting this into the system proves the cancellation:

$$\sum_{s=0}^{k} M^{-1}_{r,j_{i,s}} \cdot Y_{j_{i,s}}^{1/\alpha} = x_i^{1/\alpha} \left(\sum_{s=0}^{k} M^{-1}_{r,j_{i,s}} \cdot \gamma_{i,s} \right) = 0 \quad \text{for all } r \in I_{in} \tag{12}$$

This perfectly vanishes *all-but-the-last-tuple's* non-linear terms $x_i^{1/\alpha}$ simultaneously across all k homogeneous portions of the backward constraint equations.

While the free variables x_i span the nullspace, the k non-homogeneous constants $\mathcal{C}_r$ require a particular solution. Which means we need to reserve exactly k variables from the state before making N_{tuples} tuples. These k variables will eventually be fixed by solving the k independent input constraint equations containing $\mathcal{C}_r$s only (as the rest will be cancelled under Eqn. 12).

This leaves $t - k$ variables available for parameterization. We partition these into $N_{tuples} = \lfloor \frac{t-k}{k+1} \rfloor$ full tuples of size $k+1$. Note that a tuple size strictly greater than k is mathematically required: because the MDS matrix guarantees that any $k \times R$ submatrix with $R \leq k$ has full column rank, smaller tuples yield only the trivial nullspace (which would simply fix variables to constants rather than keeping them free).

Consequently, the leftover $R = (t - k) \bmod (k + 1)$ variables cannot form a valid collapsing tuple. They are excluded from the kernel projection and simply set to arbitrary constants (*e.g.*, 0), directly expending R DoFs. A complete visual representation of this matrix partitioning; including the nullspace-parameterized tuples, the remainder variables, and the particular solution variables; is illustrated in Figure 1.

To summarize, we consume k DoFs per full tuple to collapse $k+1$ variables into 1 free variable, k more DoFs to satisfy the non-homogeneous constants $\mathcal{C}_r$s, and finally R more DoFs to fix the remainder. In total, the number of DoFs consumed to completely absorb Round 2 is:

$$\begin{aligned} \text{Consumed DoFs} &= k \cdot \left\lfloor \frac{t-k}{k+1} \right\rfloor + k + ((t-k) \bmod (k+1)) \\ &= t - \left\lfloor \frac{t-k}{k+1} \right\rfloor \end{aligned} \tag{13}$$

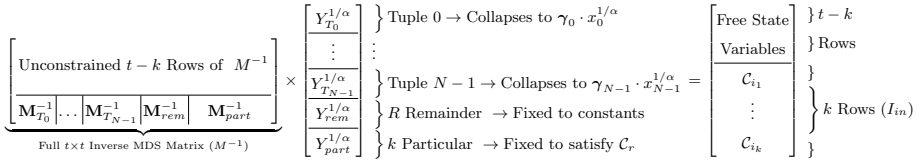

Fig. 1. Visual representation of the $(k+1)$-wise collapsing strategy across the full inverse MDS matrix. The bottom k constraint rows are partitioned into $(k+1)$-sized submatrices $\mathbf{M}_{T_i}^{-1}$. When multiplied by their respective nullspace-parameterized tuples, they evaluate to zero vectors, seamlessly absorbing the non-linear terms. The top $t-k$ rows naturally evaluate to the unconstrained free state variables that propagate forward.

leaving exactly $\lfloor \frac{t-k}{k+1} \rfloor$ free variables to seamlessly propagate forward into the subsequent rounds.

Since we need at least k variables to satisfy the final output k constraints, we require $\lfloor \frac{t-k}{k+1} \rfloor \geq k$ to avoid overdetermining the system. This means the feasibility of skipping 2 full rounds for any value of k only depends on the choice of t.

For example, with $t = 24$, the maximum k for which we can skip 2 full rounds from the front is 4. On the other hand, for small k such as 1 or 2, we can skip 2 full rounds and still have 10 or 5, respectively, more DoFs left to utilize later.

5 Exploiting Residual Degrees of Freedom

5.1 A k-Dimensional Distinguishing Attack for $t - 2k + 1$ Rounds

Under the baseline GSR gadget ($R_{f_0} = 1$), we impose $t-2k$ forward constraints and k backward constraints on Y_1. The total number of linear constraints is:

$$k + (t - 2k) = t - k\,.$$

Subtracting these constraints from the t variables of Y_1 leaves k degrees of freedom. The solution space of Y_1 forms a k-dimensional affine subspace:

$$Y_1(v_1, \ldots, v_k) = X_{base} + \sum_{i=1}^{k} v_i B_i$$

where $X_{base} \in \mathbb{F}_p^t$ is a particular solution to the linear system, $B_i \in \mathbb{F}_p^t$ are the basis vectors of the null space, and v_i are the k free variables. This parameterization traverses one full round and $t-2k$ partial rounds with a polynomial degree of exactly 1. Solving this system requires negligible effort and forms a distinguisher for the middle $t-2k+1$ rounds of Poseidon. To the best of our knowledge, this represents the longest distinguisher identified for Poseidon.

5.2 Extended Partial Round Skipping: $(2, x, y)$ Instances

The residual DoFs remaining after a $(k+1)$-wise collapse can be combined with the GSR partial-layer gadget to target instances of the form $(2, x, y)$. Let X be the vector of $\lfloor \frac{t-k}{k+1} \rfloor$ free variables remaining after the $(k+1)$-wise collapse of Round 1 and Round 2.

Because the non-linear fractional terms have been canceled, the intermediate state entering the non-linear part of the first partial round (Round 3, after round constants addition) is a strictly affine transformation of X. Let this state be $S^{(3)}(X)$. For x consecutive partial rounds ($r = 3, \ldots, 3+x-1$), we sequentially expend 1 DoF per round to constrain the single active S-box input to a chosen field constant $\delta^{(r)}$:

$$S_0^{(r)}(X) = \delta^{(r)} \tag{14}$$

This forces the S-box output to evaluate deterministically to $(\delta^{(r)})^\alpha$. The state transition to the next round becomes:

$$S^{(r+1)}(X) = M \cdot \left(S^{(r)}(X) - \delta^{(r)} \cdot e_0 + (\delta^{(r)})^\alpha \cdot e_0 \right) + C^{(r+1)} \tag{15}$$

where e_0 is the standard basis vector. Because this transition relies solely on constant additions and matrix multiplications, the state remains affine in X through all x partial rounds.

This process consumes exactly x DoFs. To ensure the final k output constraints can still be satisfied without overdetermining the system, we require the remaining free variables to satisfy $\lfloor \frac{t-k}{k+1} \rfloor - x \geq k$. This yields highly optimized, low-degree polynomial systems for instances with exactly $R_{f_0} = 2$ skipped front full rounds.

5.3 Bidirectional Collapsing: Partial Absorption of the Final Round

Alternatively, we can perform $(k+1)$-wise collapsing to absorb the two front full rounds, and utilize the remaining DoFs to re-apply the collapsing strategy at the output boundary to absorb almost the entirety of the last full round.

Let $O_{out} = \{o_1, \ldots, o_k\}$ be the k predefined output constraint indices. Let $V \in \mathbb{F}_p^t$ be the state immediately before the final S-box layer. Tracing backwards through the final MDS matrix layer, the k constraints are given by:

$$\sum_{v=0}^{t-1} M_{o,v} \cdot V_v^\alpha = \mathcal{C}_{out,o} \quad \text{for each } o \in O_{out} \tag{16}$$

To cancel the forward non-linear terms V_v^α with maximum efficiency, we re-apply $(k+1)$-wise collapsing. We group the output state variables into tuples of size $k+1$, denoted $T_i^{out} = \{j_{i,0}, \ldots, j_{i,k}\}$. We identify a non-trivial vector $\boldsymbol{\eta}_i = (\eta_{i,0}, \ldots, \eta_{i,k})^\top \in \mathbb{F}_p^{k+1}$ spanning the kernel of the $k \times (k+1)$ submatrix $\mathbf{M}_{T_i^{out}}$ derived from the target rows of M.

We parameterize the variables of the tuple using a free variable u_i as:

$$V_{j_{i,s}} = (\eta_{i,s})^{1/\alpha} \cdot u_i \quad \Longrightarrow \quad V_{j_{i,s}}^\alpha = \eta_{i,s} \cdot u_i^\alpha \tag{17}$$

By keeping the inverse power map strictly on the field constant $\eta_{i,s}$, the state V remains a strictly linear combination of the free variables u_i. Substituting this into the output constraints proves the cancellation:

$$\sum_{s=0}^{k} M_{o,j_{i,s}} \cdot V_{j_{i,s}}^{\alpha} = u_i^{\alpha} \left(\sum_{s=0}^{k} M_{o,j_{i,s}} \cdot \eta_{i,s} \right) = 0 \quad \text{for all } o \in O_{out} \tag{18}$$

The degree-α term u_i^{α} vanishes simultaneously across all k output constraints. By consuming k DoFs per tuple, we collapse $k+1$ variables into a single free variable u_i, absorbing $k+1$ S-boxes.

From the front $(k+1)$-wise collapse, we have exactly $D_{rem} = \lfloor \frac{t-k}{k+1} \rfloor$ residual DoFs. We use these to form $N_{back} = \lfloor D_{rem}/k \rfloor$ tuples at the output boundary, successfully absorbing $N_{back} \times (k+1)$ S-boxes in the final round.

For instance, in a CICO-1 ($k = 1$) configuration with $t = 24$, the front pairwise collapse leaves $D_{rem} = 11$ residual DoFs. We utilize these to form 11 pairs at the output boundary. This consumes all 11 DoFs but absorbs $11 \times 2 = 22$ S-boxes in the final round. Only $24 - 22 = 2$ S-boxes remain active to satisfy the non-homogeneous constants $\mathcal{C}_{out,o}$. This almost entirely neutralizes the final round's algebraic expansion, resulting in a total of almost 3 full rounds skipped.

We note that Poseidon implementations sometimes consider a defensive countermeasure against round skipping by adding an additional MDS matrix layer at the start and end of the cipher. We highlight that even under this protection, $(k+1)$-wise collapsing with $k = 1$ allows us to skip 1 full round from the front and almost 1 full round from the back (skipping 22 out of 24 S-boxes), successfully restoring the total bypassed full round count to 2 and defeating prepended linear defenses.

5.4 Degree-3 Ideal Reductions for $(3, x, y)$ Instances

When targeting instances of the form $(3, x, y)$, we can use $(k+1)$-wise collapsing to absorb Round 1 and Round 2, and leave the full Round 3 unskipped. Because Round 3 is a full round, its t active S-boxes $\mathsf{S}(x) = x^{\alpha}$ escalate the algebraic degree of the state expressions entering the partial layer.

Rather than consuming our remaining DoFs to force strict linearity through the partial rounds, we allow the degree growth to cascade. Let $N_v = \lfloor \frac{t-k}{k+1} \rfloor$ be the number of free variables parameterized by the $(k+1)$-wise collapse. The state exiting Round 3 consists of polynomials of strict degree α in terms of these N_v variables.

For the subsequent R_p partial rounds, we expend 1 DoF per round to constrain the single active S-box input to an arbitrary field constant $\delta^{(r)}$. Because the state entering the S-box already has degree α, this constraint does not eliminate a variable; rather, it introduces a degree-α polynomial equation into our system:

$$f_r(x_0, \ldots, x_{N_v-1}) = \delta^{(r)} \quad \text{where } \deg(f_r) = \alpha, r \in R_p \tag{19}$$

This yields exactly R_p distinct polynomial equations of degree α defined over all N_v variables. To uniquely solve the system (forming a zero-dimensional ideal),

the total number of constraints must equal the number of free variables. Since the CICO-k problem provides exactly k output constraints, we maximally target $R_p = N_v - k$ partial rounds.

These final k output constraint equations evaluate the state after it passes through the identity maps of the skipped partial rounds and the dense mixing of the R_{f_1} final full rounds. Consequently, these k equations accumulate a maximum algebraic degree of $d_{out} = \alpha \times \alpha^{R_{f_1}} = \alpha^{1+R_{f_1}}$. This reduces the entire cipher to an exactly determined N_v-variable zero-dimensional ideal:

$$\mathcal{I} = \langle f_1, \ldots, f_{R_p}, g_1, \ldots, g_k \rangle \tag{20}$$

6 Solving Methods for Full State Recovery

By successfully bounding the polynomial degree through the GSR gadget and $(k+1)$-wise collapsing, we translate Poseidon's CICO-k solution recovery problem into a highly constrained system of low-degree multivariate polynomials consisting only of the remaining unskipped partial and full rounds. Depending on the subspace dimension k, we deploy specific algebraic techniques to extract the roots and recover the complete initial state.

6.1 Univariate Interpolation ($k = 1, R_{f_0} = 1$)

For $k = 1$ and $R_{f_0} = 1$, the target is parameterized by a single variable v_1, forming a univariate polynomial $f(v_1) = 0$. The forward cipher trajectory is evaluated from S_{in} over $d+1$ numeric points in $\mathbb{F}_p$, and $f(v_1)$ is recovered via Lagrange interpolation. The roots are extracted using the Cantor-Zassenhaus algorithm [8, 17]. The degree of this univariate polynomial is strictly bounded by the remaining unskipped rounds:

$$d \leq \alpha^{R_p + R_{f_1} - t + 2k}.$$

For the root extraction of a d-degree univariate polynomial with coefficients in $\mathbb{F}_p$, the Cantor-Zassenhaus algorithm has a bit complexity of $O(d^2 \log p)$ in runtime and $O(d \log p)$ in memory.

6.2 Bivariate Interpolation using Sylvester Matrix ($k = 2, R_{f_0} = 1$)

For $k = 2$ and $R_{f_0} = 1$, the CICO-2 system is parameterized by exactly two free variables, yielding a bivariate polynomial system $g_1(v_1, v_2) = 0$ and $g_2(v_1, v_2) = 0$ representing the two output constraints. Both g_1 and g_2 have degree $d \leq \alpha^{R_p + R_{f_1} - t + 2k}$. Such a system can be solved using the determinant of the Sylvester matrix [9, 12].

By treating g_1 and g_2 as polynomials in v_2 with coefficients in $\mathbb{F}_p[v_1]$, we compute the determinant of their Sylvester matrix (denoted $\mathrm{Res}(g_1, g_2, v_1)$), which by definition equals zero if and only if the two polynomials share common roots.

This eliminates v_2, producing a univariate polynomial $\mathrm{Res}(g_1, g_2, v_1) = 0$ of degree $\leq d^2$. Standard univariate root-finding algorithms (*e.g.,* Cantor-Zassenhaus) extract the valid candidates for v_1. Back-substituting these roots into g_1 yields a smaller d-degree univariate polynomial in v_2 which is similarly solved for v_2, allowing us to recover the constrained initial state.

For two given d-degree multivariate polynomials over the same set of variables, the cost of computing their Sylvester matrix and defining the determinant as another d^2-degree multivariate polynomial (with strictly one less variable) is $\tilde{O}(d^3)$ bit operations in runtime and $O(d^2 \log p)$ bits in memory [12]. Further, if the resulting polynomial turns out to be a univariate polynomial, its root extraction cost using Cantor-Zassenhaus is $O((d^2)^2 \log p) = O(d^4 \log p)$ bit operations in runtime and $O(d^2 \log p)$ bits in memory. The overall cost becomes time $O(d^4 \log p)$ and memory $O(d^2 \log p)$.

6.3 Generalizing $k > 2$ via Trivial Pairwise Resultants

For generalized instances where $k > 2$ and $R_{f_0} = 1$, we iteratively apply pairwise resultants over the output constraint equations. Let the k output equations be $g_1, g_2, \ldots, g_k$ defined over k variables $v_1, \ldots, v_k$. We group them in pairs and compute the resultants to eliminate v_k:

$$h_i = \mathrm{Res}(g_i, g_{i+1}, v_k) \quad \text{for } i \in \{1, \ldots, k-1\}\,.$$

This yields $k-1$ equations in $k-1$ variables. Recursively applying this pairwise elimination collapses the system into a single univariate polynomial in v_1. The roots are extracted using the Cantor-Zassenhaus algorithm and iteratively back-substituted.

However, we note that this trivial approach is far from optimal. The expected degree of the k-variate system with each polynomial of degree $d \leq \alpha^{R_p + R_{f_1} - t + 2k}$ is $\leq d^k$, but this pairwise method results in a univariate polynomial with degree $d^{2^{k-1}}$. Therefore, for a given multivariate system defining a CICO-k instance, there exists a threshold on k after which the required complexity exceeds that of exhaustive search.

This iteratively nested Sylvester resultant computation has a cost of time $O(d^{2^k} \log p)$ and memory $O(d^{2^{k-1}} \log p)$.

6.4 Gröbner Basis based Solving

For systems that cannot be efficiently reduced via univariate or bivariate resultant methods (such as configurations with $R_{f_0} = 3$ or $k > 2$), we rely on Gröbner basis extraction. A standard algebraic attack utilizing Gröbner bases consists of four stages:

1. **Algebraic Modeling:** Representing the cipher as a multivariate polynomial system.
2. F_4 **Reduction:** Computing the Gröbner basis under a degree-reverse-lexicographic (degrevlex) ordering using Faugère's F_4 [10] algorithm.

3. **FGLM Conversion:** Converting the degrevlex basis into a lexicographical (lex) basis via the FGLM [11] algorithm to isolate variables and recover a univariate polynomial.
4. **Univariate Solving:** Extracting the final numerical roots (*e.g.,* via the Cantor-Zassenhaus algorithm).

The computational bottleneck of this approach lies in the F_4 and FGLM stages, whose complexities are strictly bounded by the number of variables N_v and the degrees d_i of the N_v polynomials in the modeled zero-dimensional system.

Specifically, F_4 constructs and row-reduces the **Macaulay matrix**; a large, sparse matrix where the columns represent all possible monomials up to a maximum degree D_{mac}, and the rows represent the system's original polynomials multiplied by monomials. For an exactly determined zero-dimensional ideal, this maximum degree is theoretically upper bounded by the Macaulay degree of regularity:

$$D_{mac} = 1 + \sum_{i=1}^{N_v} (d_i - 1)\,. \tag{21}$$

The column dimension of the Macaulay matrix is bounded by the total number of monomials up to D_{mac}, giving $M_{mac} = \binom{N_v + D_{mac}}{N_v}$. Consequently, the worst-case runtime complexity for F_4 is $O(M_{mac}^{\omega})$ field operations (where $2 \leq \omega < 2.37$ is the linear algebra exponent), with a memory complexity of $O(M_{mac}^2 \log p)$ bits.

Following F_4, the Sparse FGLM algorithm [11] converts the basis ordering in $O(D_{bezout}^3)$ operations,[6] where $D_{bezout} = \prod_{i=1}^{N_v} d_i$ is the Bézout bound (the maximum number of solutions in the algebraic closure). The memory complexity of FGLM is bounded by the storage of the sparse multiplication matrices, requiring $O(N_v D_{bezout}^2 \log p)$ bits.

Our advanced subspace restriction framework operates at the *modeling* stage. By absorbing initial two full rounds and actively bounding the algebraic cascade within the partial layer, we construct exactly determined, zero-dimensional ideals composed of strictly low-degree equations. This significantly reduces N_v and d_i, thereby exponentially shrinking the initial Macaulay matrix dimension M_{mac} and the Bézout bound D_{bezout}. This theoretically lowers the computation complexities for both the subsequent F_4 and FGLM stages. Furthermore, as we empirically demonstrate in Section 7.3, even these improved theoretical bounds clearly over-estimate the true operational cost of F_4.

[6] The Sparse FGLM algorithm formally operates in $O(mD_{bezout}^2)$ operations, where m is the sparsity parameter of the multiplication matrix. Because $m \leq D_{bezout}$ in the worst case, $O(D_{bezout}^3)$ serves as a strict upper bound.

7 Complexity Analysis and Empirical Benchmarks

7.1 State-of-the-Art vs. GSR+Collapsing on KoalaBear with $R_{f_0} < 3$

We instantiate the bounds using Poseidon's KoalaBear parameters adopted by the Ethereum ecosystem: $p = 2130706433 \approx 2^{31}$, $\alpha = 3$, and $t = 24$. All complexities are converted to base-2 logarithm ($\log_2(3) \approx 1.585$, $\log_2(\log_2(p)) \approx \log_2(31) \approx 4.95$) to facilitate direct comparison.

Table 1 contrasts the computational complexity of extracting valid CICO solutions between state-of-the-art baseline approaches (no partial layer subspace restriction) and our proposed GSR framework using the combination of Sylvester resultant and Cantor-Zassenhaus (SyCZ). Table 2 presents the reach of $(k+1)$-wise collapsing for instances with $R_{f_0} = 2$. Paired entries (x, y) in both tables denote (Time, Memory) bit-complexities, respectively.

Target	(R_{f_0}, R_p, R_{f_1})	Brute-force time	SyCZ (Baseline)	SyCZ (with GSR)
CICO-1 ($k = 1$)	(1, 23, 4) = 28	2^{31}	$(2^{93.7}, 2^{49.3})$	$(\mathbf{2^{20.8}}, \mathbf{2^{12.9}})$
CICO-2 ($k = 2$)	(1, 23, 4) = 28	2^{62}	$(2^{182.5}, 2^{93.7})$	$(\mathbf{2^{49.3}}, \mathbf{2^{27.1}})$
CICO-2 ($k = 2$)	(1, 20, 4) = 25	2^{62}	$(2^{163.4}, 2^{84.2})$	$(\mathbf{2^{30.3}}, \mathbf{2^{17.6}})$
CICO-3 ($k = 3$)	(1, 18, 4) = 23	2^{93}	$(2^{296.6}, 2^{150.8})$	$(\mathbf{2^{55.7}}, \mathbf{2^{30.3}})$
CICO-4 ($k = 4$)	(1, 16, 4) = 21	2^{124}	$(2^{537.5}, 2^{271.2})$	$(\mathbf{2^{106.4}}, \mathbf{2^{55.7}})$
CICO-5 ($k = 5$)	(1, 14, 2) = 17	2^{155}	$(2^{867.2}, 2^{436.1})$	$(\mathbf{2^{106.4}}, \mathbf{2^{55.7}})$
CICO-6 ($k = 6$)	(1, 12, 1) = 14	2^{186}	$(2^{1425.1}, 2^{715.0})$	$(\mathbf{2^{106.4}}, \mathbf{2^{55.7}})$
CICO-7 ($k = 7$)	(1, 10, 1) = 12	2^{217}	$(2^{2439.4}, 2^{1222.2})$	$(\mathbf{2^{207.8}}, \mathbf{2^{106.4}})$

Table 1. Complexity comparison of CICO-k solution recovery on Poseidon's KoalaBear instance ($t = 24, \alpha = 3$) for $R_{f_0} = 1$. Paired values represent (Time, Memory) bit-complexities based on the SyCZ solver. Note that for $k \geq 5$, R_{f_1} is successively reduced to 2 and then to 1 to show the R_{f_1} threshold where SyCZ (with GSR) time complexity remains faster than the respective $2^{31 \times k}$ brute-force bound. R_p is first taken as 23 and then reduced to $t - 2k$ to demonstrate maximum impact.

Analysis for $k = 1$ (with Cantor-Zassenhaus):

- **Baseline:** The algebraic degree scales up to $d_{naive} = 3^{28} \approx 2^{44.4}$. Time and memory complexities are bounded to $(2^{93.7}, 2^{49.3})$, which is far inferior to the 2^{31} brute-force guessing cost.
- **GSR ($R_{f_0} = 1$):** We skip $t - 2k = 22$ partial rounds, leaving $23 - 22 = 1$ active partial round. The degree drops to $d_{opt} = 3^{1+4} = 3^5 \approx 2^{7.9}$. Extraction complexity drops to $(\mathbf{2^{20.8}}, \mathbf{2^{12.9}})$, executing in milliseconds. (See Appendix A).
- **$(k+1)$-wise+GSR ($R_{f_0} = 2$):** Targeting $(2, 14, 4) = 20$ rounds, we skip Round 1 and Round 2, and expend 10 DoFs to linearize 10 partial rounds. The final algebraic degree is defined by the 4 remaining partial rounds and 4 final full rounds ($d = 3^8 \approx 2^{12.7}$), yielding a solving complexity of $(\mathbf{2^{30.3}}, \mathbf{2^{17.6}})$ that beats brute force.

Target Configuration	(R_{f_0}, R_p, R_{f_1})	Brute-force	SyCZ (Baseline)	SyCZ (with $(k+1)$-wise+GSR)
CICO-1 ($k = 1$)	(1, 23, 4) = 28	2^{31}	$(2^{93.7}, 2^{49.3})$	$(\mathbf{2^{20.8}}, \mathbf{2^{12.9}})$
	(2, 11, 4) = 17		$(2^{58.9}, 2^{31.9})$	
	(2, 14, 4) = 20	2^{31}	$(2^{68.4}, 2^{36.6})$	$(\mathbf{2^{30.3}}, \mathbf{2^{17.6}})$
CICO-2 ($k = 2$)	(1, 20, 4) = 25	2^{62}	$(2^{163.4}, 2^{84.2})$	$(\mathbf{2^{30.3}}, \mathbf{2^{17.6}})$
	(2, 5, 4) = 11		$(2^{74.7}, 2^{39.8})$	
	(2, 9, 4) = 15	2^{62}	$(2^{100.1}, 2^{52.5})$	$(\mathbf{2^{55.7}}, \mathbf{2^{30.3}})$

Table 2. Complexity comparison of CICO-k solution recovery on Poseidon's KoalaBear instance ($t = 24, \alpha = 3$) utilizing $(k+1)$-wise collapsing against standard subspace restriction baselines. Target configurations are grouped to demonstrate equivalent root-finding complexities (due to equal leftover unskippable rounds) and the absolute maximum reachable configurations of the form $(2, x, y)$ within brute-force cost.

Analysis for $k = 2$ (with SyCZ combo):

- **Baseline:** For (1, 20, 4), root finding without restriction projects a degree of $(3^{25})^2 = 3^{50} \approx 2^{79.3}$ for the Sylvester resultant univariate polynomial, dictating a time and memory cost of $(2^{163.4}, 2^{84.2})$. Against a 2^{62} brute-force cost, the naive approach provides no practical vector.
- **GSR ($R_{f_0} = 1$):** For (1, 20, 4), we absorb $t - 2k = 20$ partial rounds, leaving 0 active partial rounds. Equation degree drops to $d_{opt} = 3^{0+4} = 3^4 \approx 2^{6.3}$. The Sylvester matrix projection computes roots with a complexity of $(\mathbf{2^{30.3}}, \mathbf{2^{17.6}})$, running in under a minute. (See Appendix B).
- **$(k+1)$-wise+GSR ($R_{f_0} = 2$):** Targeting $(2, 9, 4) = 15$ rounds, triplewise collapsing absorbs Round 1 and Round 2, while expending DoFs to linearize 5 partial rounds. The overall complexity is bounded by $(\mathbf{2^{55.7}}, \mathbf{2^{30.3}})$, beating generic 2^{62} brute force.

7.2 State-of-the-Art vs. GSR+Collapsing on KoalaBear with $R_{f_0} = 3$

As established in Section 6.4, the theoretical complexity of extracting the Gröbner basis is dictated by two core metrics of the modeled algebraic ideal: the Macaulay matrix dimension M_{mac} (which drives the F_4 reduction cost) and the Bézout bound D_{bezout} (which drives the FGLM conversion cost).

To demonstrate the efficacy of combining $(k+1)$-wise collapsing with degree-$3^{1+R_{f_1}}$ ideal reductions at the modeling stage, we evaluate the $(3, x, y)$ bounds on Poseidon's KoalaBear instance ($t = 24, \alpha = 3$). We compare our isolated framework against the state-of-the-art algebraic baseline. In this baseline, the round skipping technique is applied on the cipher that eliminates the algebraic growth of the first two full rounds while reducing the ideal to zero dimension for $k = 1$ [6] and $k = 2$ [19]. To prevent the costly intermediate expression swell of dense polynomial expansion, the baseline also introduces a new intermediate variable for every active/unskipped S-box across the entire permutation. This

bounds all equations to strict degree-3, but inflates the variable count to

$$\begin{aligned} N_{v,naive} &= k + t \cdot (R_{f_0} - 2) + R_p + t \cdot (R_{f_1} - 1) + k \\ &= t \cdot (R_{f_0} + R_{f_1}) + R_p + 2k - 3t \end{aligned}$$

where $R_{f_1} \geq 1$ and $R_{f_0} \geq 2$.

For our framework, we apply $(k+1)$-wise collapsing. For $k = 1$, the pairwise collapse leaves $N_v = 11$ free variables, allowing us to target up to $R_p = 10$ partial rounds. For $k = 2$, triplewise collapsing leaves $N_v = 7$ free variables, allowing us to target up to $R_p = 5$ partial rounds. This transforms the full-state system into an isolated degree-$3^{1+R_{f_1}}$ system with all-but-one equations strictly capped to degree-3.

Table 3 details these algebraic boundaries and their resulting bit-complexities across several $(3, x, y)$ configurations. While we provide the overall improvements in F4 and FGLM cost in runtime and memory, we also report the raw M_{mac} and D_{bezout} bound improvements to show algorithmic metric independent of chosen linear algebra constants ($\omega = 2.37$). The time, memory and matrix size complexities are normalized to bit-complexities by applying the field characteristic factor $\log_2(p) \approx 2^{4.95}$.

Target Config (k)	**State-of-the-Art Baseline** ($N_v \geq 33$)		**Ours: Degree-3 Isolation via Collapsing**	
	F_4 (M_{mac}, **Time**, **Mem**)	**FGLM** (D_{bezout}, **Time**, **Mem**)	F_4 (M_{mac}, **Time**, **Mem**)	**FGLM** (D_{bezout}, **Time**, **Mem**)
(3, 10, 1) ($k = 1$)	($\approx 2^{95.0}, 2^{230.0}, 2^{194.9}$)	($\approx 2^{57.1}, 2^{176.1}, 2^{124.2}$)	($\approx \mathbf{2^{31.1}}, \mathbf{2^{78.7}}, \mathbf{2^{67.2}}$)	($\approx \mathbf{2^{19.0}}, \mathbf{2^{62.0}}, \mathbf{2^{46.5}}$)
(3, 10, 2) ($k = 1$)	($\approx 2^{165.7}, 2^{397.7}, 2^{336.4}$)	($\approx 2^{95.1}, 2^{290.2}, 2^{201.1}$)	($\approx \mathbf{2^{34.1}}, \mathbf{2^{85.7}}, \mathbf{2^{73.1}}$)	($\approx \mathbf{2^{20.6}}, \mathbf{2^{66.8}}, \mathbf{2^{49.6}}$)
(3, 10, 3) ($k = 1$)	($\approx 2^{234.3}, 2^{560.2}, 2^{473.6}$)	($\approx 2^{133.1}, 2^{404.3}, 2^{277.6}$)	($\approx \mathbf{2^{48.9}}, \mathbf{2^{120.9}}, \mathbf{2^{102.8}}$)	($\approx \mathbf{2^{22.2}}, \mathbf{2^{71.5}}, \mathbf{2^{52.8}}$)
(3, 5, 1) ($k = 2$)	($\approx 2^{86.9}, 2^{210.8}, 2^{178.7}$)	($\approx 2^{52.3}, 2^{161.9}, 2^{114.6}$)	($\approx \mathbf{2^{22.4}}, \mathbf{2^{58.0}}, \mathbf{2^{49.7}}$)	($\approx \mathbf{2^{14.3}}, \mathbf{2^{47.7}}, \mathbf{2^{36.3}}$)
(3, 5, 2) ($k = 2$)	($\approx 2^{157.0}, 2^{377.0}, 2^{319.0}$)	($\approx 2^{90.3}, 2^{276.0}, 2^{191.5}$)	($\approx \mathbf{2^{30.2}}, \mathbf{2^{76.4}}, \mathbf{2^{65.3}}$)	($\approx \mathbf{2^{17.4}}, \mathbf{2^{57.3}}, \mathbf{2^{42.6}}$)
(3, 5, 3) ($k = 2$)	($\approx 2^{225.8}, 2^{540.1}, 2^{456.6}$)	($\approx 2^{128.4}, 2^{390.1}, 2^{268.1}$)	($\approx \mathbf{2^{39.9}}, \mathbf{2^{99.5}}, \mathbf{2^{84.7}}$)	($\approx \mathbf{2^{20.6}}, \mathbf{2^{66.8}}, \mathbf{2^{49.0}}$)

Table 3. Gröbner Basis parameters and theoretical bit-complexities for CICO-k recovery on Poseidon ($t = 24, \alpha = 3$).

Because the state-of-the-art baseline approach must introduce an intermediate variable for every active S-box to cap polynomial degrees to 3, the variable count inflates significantly ($N_{v,naive}$ ranges from 33 to 84) despite already successfully skipping the first two full rounds. This increased number of variables drives the combinatorial explosion of both the Macaulay dimension and Bézout bounds to unmanageable extremes, resulting in an F_4 time complexity of, for example, $2^{560.2}$ bit operations for the $(3, 10, 3)$ CICO-1 instance. By contrast, our framework leverages subspace restriction to confine the system strictly to k equations of degree $3^{1+R_{f_1}}$ and the remaining $N_v - k$ equations to degree 3, nicely capping the variable count to a fraction of the baseline ($N_v = 11$ or 7). This limits both M_{mac} and D_{bezout} bounds, improving the F_4 and FGLM time complexities by factors of 2^{440} and 2^{333}, respectively, for the $(3, 10, 3)$ configuration. By strategically avoiding intermediate expression swell while simultaneously compressing the algebraic density into a very small set of variables, our framework strictly

bounds matrix expansion, improving mathematically intractable configurations by a significant margin.

7.3 Theory vs. Practice: Empirical Results on $(3, 8, 0)$ Instance

For a practical benchmark refuting monolithic upper bounds, we considered a reduced CICO-1 configuration of $(3, 8, 0)$. By setting $R_{f_1} = 0$, we restrict d_i to 3 for all equations. By targeting $R_p = 8$ partial rounds instead of the maximum 10, we reserve 2 DoFs to fix 2 variables (out of 11) to arbitrary constants. This leaves an exactly determined system of:

$$9 \text{ variables and } 9 \text{ equations, all of strict degree } 3.$$

This system yields a theoretical maximum Macaulay bound of $D_{mac} = 1 + 8(2) + 1(2) = 19$, corresponding to a monolithic matrix column dimension of $M_{mac} = \binom{28}{9} = 6,906,900 \approx 2^{22.7}$. Theoretically, this dictates an upper bound runtime of $\approx \mathbf{2^{53.8}}$ field ops and a memory footprint of $\approx \mathbf{2^{50.4}}$ bits, placing it seemingly beyond the capability of a single server.

However, executing this $(3, 8, 0)$ instance using `msolve`'s F_4 implementation [7] on an Intel Xeon Gold 6248R AVX-512 server (multi-threaded over 96 threads) surpassed these estimates by a significant margin.

The Memory Gap. The theoretical bound assumes a dense monolithic matrix requiring $O(M_{mac}^2 \cdot \log(p)) \approx 2^{50.4}$ bits of storage. In reality, polynomials in 9 variables have at most $K = \binom{9+3}{3} = 220$ terms, meaning the matrix is highly sparse and contains at most $O(K \cdot M_{mac})$ non-zero entries. Furthermore, modern F_4 implementations construct sparse matrices degree-by-degree. The maximum dimension of the step-matrix does not occur at D_{mac}, but rather peaks at an intermediate degree of regularity before strictly decreasing as the ideal's dimension collapses.

Tracking the F_4 reduction for $(3, 8, 0)$ confirms that the active step-matrix strictly increased in size until peaking at degrees $D_{peak} = 14$, reaching a maximum dimension of only $37,288 \times 51,852$. A complete step-by-step execution log detailing this intermediate peak and the matrix densities is provided in Figure 2. This peak matrix width of $\approx 2^{15.7}$ columns is exponentially smaller than the monolithic bound $M_{mac} \approx 2^{22.7}$. Additionally, `msolve` utilizes AVX-512 vectorization to process word-sized finite fields natively. Consequently, the maximum memory footprint peaked at less than 10 GB (far smaller than the estimate of $2^{50.4}$ bits).

The Runtime Gap. Because the active matrix width is bounded by the intermediate peak and aggressively pruned by F_4's symbolic preprocessing (which removes columns corresponding to reducible monomials), runtime is drastically reduced. The zero-dimensional system gracefully collapsed after the intermediate peak, successfully terminating at a solving degree of 20. The entire Gröbner basis was extracted by F_4 in exactly 248 seconds (4.1 minutes). This significant

runtime gap validates the practicality of `msolve` (a hardware-accelerated sparse matrix framework) and demonstrates that true algebraic solving complexity is heavily over-estimated by monolithic bounds.

```
Legend for f4 information
---------------------------------------------------------
deg       current degree of pairs selected in this round
sel       number of pairs selected in this round
pairs     total number of pairs in pair list
mat       matrix dimensions (# rows x # columns)
density   density of the matrix
new data  # new elements for basis in this round
          # zero reductions during linear algebra
time(rd)  time of the current f4 round in seconds given
          for real and cpu time
---------------------------------------------------------

deg     sel   pairs        mat          density            new data         time(rd) in sec (real|cpu)
-------------------------------------------------------------------------------------------------------
  3       8       8       9 x 220       100.00%        8 new       0 zero         0.00 | 0.19
  4      13      13      89 x 714        29.81%       13 new       0 zero         0.10 | 9.33
  5      39      39     419 x 1921       17.75%       34 new       5 zero         0.23 | 21.42
  6     145     145    1330 x 4315       14.53%       69 new      76 zero         0.47 | 43.28
  7     351     351    3355 x 8495       12.59%      126 new     225 zero         0.96 | 80.28
  8     756     756    6900 x 14641      11.59%      225 new     531 zero         2.10 | 154.86
  9    1497    1497   12340 x 22625      11.81%      371 new    1126 zero         4.98 | 322.26
 10    2597    2597   18838 x 31023      12.10%      464 new    2133 zero        10.57 | 661.04
 11    3480    3480   25443 x 39188      12.53%      603 new    2877 zero        20.11 | 1251.85
 12    4824    4824   31126 x 45228      12.51%      750 new    4074 zero        32.96 | 2026.55
 13    6000    6000   35962 x 49692      13.33%      672 new    5328 zero        43.76 | 2853.21
 14    5376    5376   37288 x 51852      13.40%      468 new    4908 zero        41.91 | 2806.29
 15    3744    3744   36369 x 52511      12.89%      258 new    3486 zero        32.36 | 2059.34
 16    2064    2064   34754 x 52474      12.05%      111 new    1953 zero        20.24 | 1087.78
 17     888     888   33442 x 52272      11.27%       36 new     852 zero        11.99 | 418.78
 18     288     288   32716 x 52119      10.77%        8 new     280 zero         9.38 | 212.48
 19      64      64   32424 x 52044      10.56%        1 new      63 zero         6.85 | 29.28
 20       8       8   32342 x 52017      10.49%        0 new       8 zero         6.80 | 14.60
-------------------------------------------------------------------------------------------------------
reduce final basis     4218 x 23901      66.85%     4218 new       0 zero         2.70 | 6.14
-------------------------------------------------------------------------------------------------------

---------------- TIMINGS ----------------
overall(elapsed)      248.60 sec
overall(cpu)        14060.28 sec
select                 26.88 sec  10.8%
symbolic prep.         52.91 sec  21.3%
update                 25.44 sec  10.2%
convert                 0.53 sec   0.2%
linear algebra        140.84 sec  56.7%
reduce gb               0.00 sec   0.0%
-----------------------------------------

---------- COMPUTATIONAL DATA -----------
size of basis               4218
#terms in basis            67399276
```

Fig. 2. Execution log of `msolve`'s F_4 algorithm for the $(3, 8, 0)$ CICO-1 configuration. The log empirically demonstrates the step-wise nature of the reduction, showing the active matrix dimensions peaking at degrees 14 and 15 before the zero-dimensional system gracefully collapses.

8 Conclusion

In this paper, we introduced an advanced, unified algebraic framework for subspace restriction against the Poseidon hash function. We first established the baseline Generalized S-Box Skipping (GSR) gadget, optimizing state recovery

for configurations with $R_{f_0} = 1$. We then expanded these algebraic capabilities by formalizing non-trivial $(k+1)$-wise collapsing, successfully neutralizing dense MDS matrix mixing across multiple initial full rounds.

These structural exploitation frameworks allowed us to translate mathematically challenging CICO-k problems into tractable, low-degree polynomial systems. We pushed the limits of subspace-restriction-based full state recoveries for $R_{f_0} = 1$ to 28 rounds for CICO-1 and 25 rounds for CICO-2 and for $R_{f_0} = 2$ to 20 rounds for CICO-1 and 15 rounds for CICO-2 on Poseidon's KoalaBear instance with $t = 24$. We further demonstrated how to efficiently utilize residual DoFs via bidirectional skipping to defeat recently proposed linear layer countermeasure, and formalized degree-$3^{1+R_{f_1}}$ ideal reductions to make concrete advancements for $R_{f_0} = 3$ instances.

Finally, our empirical Gröbner basis extraction of a mathematically intractable $(3, 8, 0)$ instance highlights the disconnect between theoretical complexity estimates and practical cryptanalysis. It proves that deriving security margins from monolithic Macaulay bounds is inaccurate, which we will investigate in future work.

In summary, this work exposes some critical vulnerabilities in the partial layer structure of Poseidon (and related AO ciphers), emphasizing the need for a rigorous reevaluation of their algebraic security parameters.

Acknowledgements. This work was supported by the Ethereum Foundation Grant program through grant number *FY26-2457* and the VLAIO Baekeland mandate under project number HBC.2024.0256. An earlier version of this paper was disclosed to the Ethereum Foundation prior to their public dissemination.

A CICO-1 Attack Parameters [Supplementary Material]

In this section, we present four fully verifiable CICO-1 solutions for Poseidon operating over the KoalaBear prime field $p = 2130706433$ with state size $t = 24$, $\alpha = 3$, and $(R_{f_0}, R_p, R_{f_1}) = (1, 23, 4)$ *i.e.,* 28 out of recommended 31 rounds. The MDS matrix used here is a verified secure circulant matrix derived using official poseidon-tools-main repository [2], and the pseudo-random round constants are generated via SHAKE-256 hash derivations [18]. By targeting constraints `[0]` at both the input and output boundaries, our solver successfully extracted multiple solutions satisfying an instance of one full round, 23 partial rounds, and four full rounds (*i.e.,* (1, 23, 4) rounds) of Poseidon.

```
"mds": [
    [1065186607, 131208511, 428720827, 1207080596,
       1738166676, 1988175222, 81239933, 70865233,
       918233319, 807502311, 552595000, 168534697,
       1604493419, 1758837349, 309831590, 673013262,
       192605313, 1106785377, 1409080307, 610426990,
       873018199, 1492781856, 1942555555, 2096981614],
```

```
[2096981614, 1065186607, 131208511, 428720827,
    1207080596, 1738166676, 1988175222, 81239933,
    70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349, 309831590,
    673013262, 192605313, 1106785377, 1409080307,
    610426990, 873018199, 1492781856, 1942555555],
[1942555555, 2096981614, 1065186607, 131208511,
    428720827, 1207080596, 1738166676, 1988175222,
    81239933, 70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349, 309831590,
    673013262, 192605313, 1106785377, 1409080307,
    610426990, 873018199, 1492781856],
[1492781856, 1942555555, 2096981614, 1065186607,
    131208511, 428720827, 1207080596, 1738166676,
    1988175222, 81239933, 70865233, 918233319,
    807502311, 552595000, 168534697, 1604493419,
    1758837349, 309831590, 673013262, 192605313,
    1106785377, 1409080307, 610426990, 873018199],
[873018199, 1492781856, 1942555555, 2096981614,
    1065186607, 131208511, 428720827, 1207080596,
    1738166676, 1988175222, 81239933, 70865233,
    918233319, 807502311, 552595000, 168534697,
    1604493419, 1758837349, 309831590, 673013262,
    192605313, 1106785377, 1409080307, 610426990],
[610426990, 873018199, 1492781856, 1942555555,
    2096981614, 1065186607, 131208511, 428720827,
    1207080596, 1738166676, 1988175222, 81239933,
    70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349, 309831590,
    673013262, 192605313, 1106785377, 1409080307],
[1409080307, 610426990, 873018199, 1492781856,
    1942555555, 2096981614, 1065186607, 131208511,
    428720827, 1207080596, 1738166676, 1988175222,
    81239933, 70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349, 309831590,
    673013262, 192605313, 1106785377],
[1106785377, 1409080307, 610426990, 873018199,
    1492781856, 1942555555, 2096981614, 1065186607,
    131208511, 428720827, 1207080596, 1738166676,
    1988175222, 81239933, 70865233, 918233319,
    807502311, 552595000, 168534697, 1604493419,
    1758837349, 309831590, 673013262, 192605313],
[192605313, 1106785377, 1409080307, 610426990,
    873018199, 1492781856, 1942555555, 2096981614,
    1065186607, 131208511, 428720827, 1207080596,
    1738166676, 1988175222, 81239933, 70865233,
    918233319, 807502311, 552595000, 168534697,
    1604493419, 1758837349, 309831590, 673013262],
[673013262, 192605313, 1106785377, 1409080307,
    610426990, 873018199, 1492781856, 1942555555,
```

```
    2096981614, 1065186607, 131208511, 428720827,
    1207080596, 1738166676, 1988175222, 81239933,
    70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349, 309831590],
[309831590, 673013262, 192605313, 1106785377,
    1409080307, 610426990, 873018199, 1492781856,
    1942555555, 2096981614, 1065186607, 131208511,
    428720827, 1207080596, 1738166676, 1988175222,
    81239933, 70865233, 918233319, 807502311, 552595000,
    168534697, 1604493419, 1758837349],
[1758837349, 309831590, 673013262, 192605313,
    1106785377, 1409080307, 610426990, 873018199,
    1492781856, 1942555555, 2096981614, 1065186607,
    131208511, 428720827, 1207080596, 1738166676,
    1988175222, 81239933, 70865233, 918233319,
    807502311, 552595000, 168534697, 1604493419],
[1604493419, 1758837349, 309831590, 673013262,
    192605313, 1106785377, 1409080307, 610426990,
    873018199, 1492781856, 1942555555, 2096981614,
    1065186607, 131208511, 428720827, 1207080596,
    1738166676, 1988175222, 81239933, 70865233,
    918233319, 807502311, 552595000, 168534697],
[168534697, 1604493419, 1758837349, 309831590,
    673013262, 192605313, 1106785377, 1409080307,
    610426990, 873018199, 1492781856, 1942555555,
    2096981614, 1065186607, 131208511, 428720827,
    1207080596, 1738166676, 1988175222, 81239933,
    70865233, 918233319, 807502311, 552595000],
[552595000, 168534697, 1604493419, 1758837349,
    309831590, 673013262, 192605313, 1106785377,
    1409080307, 610426990, 873018199, 1492781856,
    1942555555, 2096981614, 1065186607, 131208511,
    428720827, 1207080596, 1738166676, 1988175222,
    81239933, 70865233, 918233319, 807502311],
[807502311, 552595000, 168534697, 1604493419,
    1758837349, 309831590, 673013262, 192605313,
    1106785377, 1409080307, 610426990, 873018199,
    1492781856, 1942555555, 2096981614, 1065186607,
    131208511, 428720827, 1207080596, 1738166676,
    1988175222, 81239933, 70865233, 918233319],
[918233319, 807502311, 552595000, 168534697, 1604493419,
    1758837349, 309831590, 673013262, 192605313,
    1106785377, 1409080307, 610426990, 873018199,
    1492781856, 1942555555, 2096981614, 1065186607,
    131208511, 428720827, 1207080596, 1738166676,
    1988175222, 81239933, 70865233],
[70865233, 918233319, 807502311, 552595000, 168534697,
    1604493419, 1758837349, 309831590, 673013262,
    192605313, 1106785377, 1409080307, 610426990,
    873018199, 1492781856, 1942555555, 2096981614,
```

```
        1065186607, 131208511, 428720827, 1207080596,
        1738166676, 1988175222, 81239933],
    [81239933, 70865233, 918233319, 807502311, 552595000,
        168534697, 1604493419, 1758837349, 309831590,
        673013262, 192605313, 1106785377, 1409080307,
        610426990, 873018199, 1492781856, 1942555555,
        2096981614, 1065186607, 131208511, 428720827,
        1207080596, 1738166676, 1988175222],
    [1988175222, 81239933, 70865233, 918233319, 807502311,
        552595000, 168534697, 1604493419, 1758837349,
        309831590, 673013262, 192605313, 1106785377,
        1409080307, 610426990, 873018199, 1492781856,
        1942555555, 2096981614, 1065186607, 131208511,
        428720827, 1207080596, 1738166676],
    [1738166676, 1988175222, 81239933, 70865233, 918233319,
        807502311, 552595000, 168534697, 1604493419,
        1758837349, 309831590, 673013262, 192605313,
        1106785377, 1409080307, 610426990, 873018199,
        1492781856, 1942555555, 2096981614, 1065186607,
        131208511, 428720827, 1207080596],
    [1207080596, 1738166676, 1988175222, 81239933, 70865233,
        918233319, 807502311, 552595000, 168534697,
        1604493419, 1758837349, 309831590, 673013262,
        192605313, 1106785377, 1409080307, 610426990,
        873018199, 1492781856, 1942555555, 2096981614,
        1065186607, 131208511, 428720827],
    [428720827, 1207080596, 1738166676, 1988175222,
        81239933, 70865233, 918233319, 807502311, 552595000,
        168534697, 1604493419, 1758837349, 309831590,
        673013262, 192605313, 1106785377, 1409080307,
        610426990, 873018199, 1492781856, 1942555555,
        2096981614, 1065186607, 131208511],
    [131208511, 428720827, 1207080596, 1738166676,
        1988175222, 81239933, 70865233, 918233319,
        807502311, 552595000, 168534697, 1604493419,
        1758837349, 309831590, 673013262, 192605313,
        1106785377, 1409080307, 610426990, 873018199,
        1492781856, 1942555555, 2096981614, 1065186607]
],
"round_constants": [
    1137173758, 1158021119, 2090848577, 1073741823,
        2115911463, 2049454079,
    1206189706, 100663295, 533807487, 507455476, 674234367,
        1694403083,
    1894906175, 1363247103, 1059583056, 805306367,
        2122976255, 1912160162,
    1098907647, 670322783, 126863869, 168558591, 2034213506,
        2084339279,
    1951424511, 264895764, 201326591, 1067614975,
        1014910952, 1348468735,
```

```
1241322519, 1642328703, 579010559, 2119166112,
    1610612735, 2098468863,
1676836677, 50331647, 1340645567, 253727738, 337117183,
    1920943365,
2021194911, 1755365375, 529791528, 402653183,
    2029821905, 549453823,
335161391, 1137173758, 1158021119, 2090848577,
    1073741823, 2115911463,
2049454079, 1206189706, 100663295, 533807487, 507455476,
    674234367,
1694403083, 1894906175, 1363247103, 1059583056,
    805306367, 2122976255,
1912160162, 1098907647, 670322783, 126863869, 168558591,
    2034213506,
2084339279, 1951424511, 264895764, 201326591,
    1067614975, 1014910952,
1348468735, 1241322519, 1642328703, 579010559,
    2119166112, 1610612735,
2098468863, 1676836677, 50331647, 1340645567, 253727738,
    337117183,
1920943365, 2021194911, 1755365375, 529791528,
    402653183, 2029821905,
549453823, 335161391, 1137173758, 1158021119,
    2090848577, 1073741823,
2115911463, 2049454079, 1206189706, 100663295,
    533807487, 507455476,
674234367, 1694403083, 1894906175, 1363247103,
    1059583056, 805306367,
2122976255, 1912160162, 1098907647, 670322783,
    126863869, 168558591,
2034213506, 2084339279, 1951424511, 264895764,
    201326591, 1067614975,
1014910952, 1348468735, 1241322519, 1642328703,
    579010559, 2119166112,
1610612735, 2098468863, 1676836677, 50331647,
    1340645567, 253727738,
337117183, 1920943365, 2021194911, 1755365375,
    529791528, 402653183,
2029821905, 549453823, 335161391, 1137173758,
    1158021119, 2090848577,
1073741823, 2115911463, 2049454079, 1206189706,
    100663295, 533807487,
507455476, 674234367, 1694403083, 1894906175,
    1363247103, 1059583056,
805306367, 2122976255, 1912160162, 1098907647,
    670322783, 126863869,
168558591, 2034213506, 2084339279, 1951424511,
    264895764, 201326591,
1067614975, 1014910952, 1348468735, 1241322519,
    1642328703, 579010559,
```

```
2119166112, 1610612735, 2098468863, 1676836677,
    50331647, 1340645567,
253727738, 337117183, 1920943365, 2021194911,
    1755365375, 529791528,
402653183, 2029821905, 549453823, 335161391, 1137173758,
    1158021119,
2090848577, 1073741823, 2115911463, 2049454079,
    1206189706, 100663295,
533807487, 507455476, 674234367, 1694403083, 1894906175,
    1363247103,
1059583056, 805306367, 2122976255, 1912160162,
    1098907647, 670322783,
126863869, 168558591, 2034213506, 2084339279,
    1951424511, 264895764,
201326591, 1067614975, 1014910952, 1348468735,
    1241322519, 1642328703,
579010559, 2119166112, 1610612735, 2098468863,
    1676836677, 50331647,
1340645567, 253727738, 337117183, 1920943365,
    2021194911, 1755365375,
529791528, 402653183, 2029821905, 549453823, 335161391,
    1137173758,
1158021119, 2090848577, 1073741823, 2115911463,
    2049454079, 1206189706,
100663295, 533807487, 507455476, 674234367, 1694403083,
    1894906175,
1363247103, 1059583056, 805306367, 2122976255,
    1912160162, 1098907647,
670322783, 126863869, 168558591, 2034213506, 2084339279,
    1951424511,
264895764, 201326591, 1067614975, 1014910952,
    1348468735, 1241322519,
1642328703, 579010559, 2119166112, 1610612735,
    2098468863, 1676836677,
50331647, 1340645567, 253727738, 337117183, 1920943365,
    2021194911,
1755365375, 529791528, 402653183, 2029821905, 549453823,
    335161391,
1137173758, 1158021119, 2090848577, 1073741823,
    2115911463, 2049454079,
1206189706, 100663295, 533807487, 507455476, 674234367,
    1694403083,
1894906175, 1363247103, 1059583056, 805306367,
    2122976255, 1912160162,
1098907647, 670322783, 126863869, 168558591, 2034213506,
    2084339279,
1951424511, 264895764, 201326591, 1067614975,
    1014910952, 1348468735,
1241322519, 1642328703, 579010559, 2119166112,
    1610612735, 2098468863,
```

```
    1676836677, 50331647, 1340645567, 253727738, 337117183,
        1920943365,
    2021194911, 1755365375, 529791528, 402653183,
        2029821905, 549453823,
    335161391, 1137173758, 1158021119, 2090848577,
        1073741823, 2115911463,
    2049454079, 1206189706, 100663295, 533807487, 507455476,
        674234367,
    1694403083, 1894906175, 1363247103, 1059583056,
        805306367, 2122976255,
    1912160162, 1098907647, 670322783, 126863869, 168558591,
        2034213506,
    2084339279, 1951424511, 264895764, 201326591,
        1067614975, 1014910952,
    1348468735, 1241322519, 1642328703, 579010559,
        2119166112, 1610612735,
    2098468863, 1676836677, 50331647, 1340645567, 253727738,
        337117183,
    1920943365, 2021194911, 1755365375, 529791528,
        402653183, 2029821905,
    549453823, 335161391, 1137173758, 1158021119,
        2090848577, 1073741823,
    2115911463, 2049454079, 1206189706, 100663295,
        533807487, 507455476,
    674234367, 1694403083, 1894906175, 1363247103,
        1059583056, 805306367,
    2122976255, 1912160162, 1098907647, 670322783,
        126863869, 168558591,
    2034213506, 2084339279, 1951424511, 264895764,
        201326591, 1067614975,
    1014910952, 1348468735, 1241322519, 1642328703,
        579010559, 2119166112,
    1610612735, 2098468863, 1676836677, 50331647,
        1340645567, 253727738,
    337117183, 1920943365, 2021194911, 1755365375,
        529791528, 402653183,
    2029821905, 549453823, 335161391, 1137173758,
        1158021119, 2090848577,
    1073741823, 2115911463, 2049454079, 1206189706,
        100663295, 533807487,
    507455476, 674234367, 1694403083, 1894906175,
        1363247103, 1059583056,
    805306367, 2122976255, 1912160162, 1098907647,
        670322783, 126863869,
    168558591, 2034213506, 2084339279, 1951424511,
        264895764, 201326591,
    1067614975, 1014910952, 1348468735, 1241322519,
        1642328703, 579010559,
    2119166112, 1610612735
],
```

```
"solutions": [
    {
        "input_state_1": [
            0, 1589364078, 1896430845, 875661494,
                1347204565, 458155487,
            1384987356, 571290314, 330532031, 1770159316,
                1420433581, 1163438003,
            337130899, 1120934139, 1044017398, 1544907671,
                886524037, 605359254,
            744064511, 136779467, 1310164241, 780342668,
                789268467, 1840097141
        ],
        "output_state_1": [
            0, 1597060746, 1364711247, 2024085809,
                1512940021, 639813075,
            2091129989, 806642337, 1844843578, 1440263009,
                756609248, 263448336,
            1618326804, 811620658, 690866717, 933217052,
                378688250, 1799766079,
            1214958415, 1478995146, 222650518, 953822748,
                1832066125, 854422796
        ]
    },
    {
        "input_state_2": [
            0, 1379788884, 1164672055, 1191892707,
                532821958, 461674733,
            1413084694, 822997674, 741936930, 457154561,
                2018368438, 131208049,
            1011428151, 1251905695, 1061341129, 1974638428,
                1250553775, 1354810513,
            752495316, 599296425, 1371731441, 412465195,
                1768055340, 465027489
        ],
        "output_state_2": [
            0, 1008747893, 1419899900, 521561470, 185111988,
                528848150,
            1922979060, 253468514, 1353778182, 769215756,
                1272714572, 55194805,
            1791994868, 313734210, 222115939, 1618161927,
                1750950112, 1898860391,
            1735545169, 1131269616, 1707218116, 78404225,
                354584698, 1992700870
        ]
    },
    {
        "input_state_3": [
            0, 1192173019, 2043837185, 1184509588,
                1362931591, 824908447,
```

```
            1288338852, 335401995, 8741120, 1941566821,
                1849825894, 1630091427,
            289382118, 2048304857, 1752306838, 387944395,
                377511001, 1316881134,
            1645715643, 1061611602, 243851999, 1749172782,
                1273017419, 511139102
        ],
        "output_state_3": [
            0, 496724098, 581343189, 448783366, 1958178900,
                1487404312,
            463366279, 285055731, 682583592, 1845363571,
                465657558, 1074455959,
            275076066, 1954497979, 1949185051, 557858070,
                936141974, 306893588,
            1525066889, 1648361248, 1344575339, 1625371935,
                1961167704, 856000905
        ]
    },
    {
        "input_state_4": [
            0, 1221687458, 142017802, 170800535, 1703008619,
                477515941,
            1643535378, 1274866856, 206044471, 1587794015,
                1143959450, 1300562349,
            1560159219, 657354945, 584529588, 1593984937,
                596667400, 1316480087,
            66852599, 1986668902, 990307644, 1909187780,
                1966059188, 239791987
        ],
        "output_state_4": [
            0, 1651826367, 470722407, 967325238, 948366201,
                1212345581,
            629845574, 302533854, 1805852539, 509050066,
                886610095, 372933298,
            1662560570, 992763186, 1448826727, 1884229999,
                1642795887, 1733990605,
            712258464, 1835007496, 254582648, 1864356433,
                783573503, 946578381
        ]
    }
]
```


B CICO-2 Attack Parameters [Supplementary Material]

CICO-2 attack is a harder attack than CICO-1, and our recovered CICO-2 solution exhibit a vulnerability in 25 rounds out of Poseidon's recommended 31 rounds in the KoalaBear ($t = 24$) setting.

More specifically, we present four fully verifiable CICO-2 solutions mapping for Poseidon operating over the KoalaBear prime field $p = 2130706433$ with state size $t = 24$, $\alpha = 3$, $(R_{f_0}, R_p, R_{f_1}) = (1, 20, 4)$. The MDS matrix used here is also a verified secure circulant matrix derived using official poseidon-tools-main repository [2], and the pseudo-random round constants are generated via SHAKE-256 hash derivations [18]. By targeting constraints `[0, 0]` at both the input and output boundaries, our solver successfully extracted multiple solutions satisfying resulting constraints for this parameter set.


```
"mds": [
    [182029519, 812174379, 799647766, 1902072930,
       1646926355, 473881174, 1583631138, 677760056,
       838789176, 827331457, 1322308383, 1239059097,
       233273648, 55471811, 929516509, 2062955645,
       1362536314, 218724220, 836843832, 14387358,
       1396572830, 1966472522, 1119098349, 1443459654],
    [1443459654, 182029519, 812174379, 799647766,
       1902072930, 1646926355, 473881174, 1583631138,
       677760056, 838789176, 827331457, 1322308383,
       1239059097, 233273648, 55471811, 929516509,
       2062955645, 1362536314, 218724220, 836843832,
       14387358, 1396572830, 1966472522, 1119098349],
    [1119098349, 1443459654, 182029519, 812174379,
       799647766, 1902072930, 1646926355, 473881174,
       1583631138, 677760056, 838789176, 827331457,
       1322308383, 1239059097, 233273648, 55471811,
       929516509, 2062955645, 1362536314, 218724220,
       836843832, 14387358, 1396572830, 1966472522],
    [1966472522, 1119098349, 1443459654, 182029519,
       812174379, 799647766, 1902072930, 1646926355,
       473881174, 1583631138, 677760056, 838789176,
       827331457, 1322308383, 1239059097, 233273648,
       55471811, 929516509, 2062955645, 1362536314,
       218724220, 836843832, 14387358, 1396572830],
    [1396572830, 1966472522, 1119098349, 1443459654,
       182029519, 812174379, 799647766, 1902072930,
       1646926355, 473881174, 1583631138, 677760056,
       838789176, 827331457, 1322308383, 1239059097,
       233273648, 55471811, 929516509, 2062955645,
       1362536314, 218724220, 836843832, 14387358],
    [14387358, 1396572830, 1966472522, 1119098349,
       1443459654, 182029519, 812174379, 799647766,
       1902072930, 1646926355, 473881174, 1583631138,
       677760056, 838789176, 827331457, 1322308383,
       1239059097, 233273648, 55471811, 929516509,
       2062955645, 1362536314, 218724220, 836843832],
    [836843832, 14387358, 1396572830, 1966472522,
       1119098349, 1443459654, 182029519, 812174379,
       799647766, 1902072930, 1646926355, 473881174,
```

```
    1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314, 218724220],
[218724220, 836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355, 473881174,
    1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314],
[1362536314, 218724220, 836843832, 14387358, 1396572830,
    1966472522, 1119098349, 1443459654, 182029519,
    812174379, 799647766, 1902072930, 1646926355,
    473881174, 1583631138, 677760056, 838789176,
    827331457, 1322308383, 1239059097, 233273648,
    55471811, 929516509, 2062955645],
[2062955645, 1362536314, 218724220, 836843832, 14387358,
    1396572830, 1966472522, 1119098349, 1443459654,
    182029519, 812174379, 799647766, 1902072930,
    1646926355, 473881174, 1583631138, 677760056,
    838789176, 827331457, 1322308383, 1239059097,
    233273648, 55471811, 929516509],
[929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355, 473881174,
    1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811],
[55471811, 929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355, 473881174,
    1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648],
[233273648, 55471811, 929516509, 2062955645, 1362536314,
    218724220, 836843832, 14387358, 1396572830,
    1966472522, 1119098349, 1443459654, 182029519,
    812174379, 799647766, 1902072930, 1646926355,
    473881174, 1583631138, 677760056, 838789176,
    827331457, 1322308383, 1239059097],
[1239059097, 233273648, 55471811, 929516509, 2062955645,
    1362536314, 218724220, 836843832, 14387358,
    1396572830, 1966472522, 1119098349, 1443459654,
    182029519, 812174379, 799647766, 1902072930,
    1646926355, 473881174, 1583631138, 677760056,
    838789176, 827331457, 1322308383],
[1322308383, 1239059097, 233273648, 55471811, 929516509,
    2062955645, 1362536314, 218724220, 836843832,
    14387358, 1396572830, 1966472522, 1119098349,
    1443459654, 182029519, 812174379, 799647766,
```

```
    1902072930, 1646926355, 473881174, 1583631138,
    677760056, 838789176, 827331457],
[827331457, 1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355, 473881174,
    1583631138, 677760056, 838789176],
[838789176, 827331457, 1322308383, 1239059097,
    233273648, 55471811, 929516509, 2062955645,
    1362536314, 218724220, 836843832, 14387358,
    1396572830, 1966472522, 1119098349, 1443459654,
    182029519, 812174379, 799647766, 1902072930,
    1646926355, 473881174, 1583631138, 677760056],
[677760056, 838789176, 827331457, 1322308383,
    1239059097, 233273648, 55471811, 929516509,
    2062955645, 1362536314, 218724220, 836843832,
    14387358, 1396572830, 1966472522, 1119098349,
    1443459654, 182029519, 812174379, 799647766,
    1902072930, 1646926355, 473881174, 1583631138],
[1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355, 473881174],
[473881174, 1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379,
    799647766, 1902072930, 1646926355],
[1646926355, 473881174, 1583631138, 677760056,
    838789176, 827331457, 1322308383, 1239059097,
    233273648, 55471811, 929516509, 2062955645,
    1362536314, 218724220, 836843832, 14387358,
    1396572830, 1966472522, 1119098349, 1443459654,
    182029519, 812174379, 799647766, 1902072930],
[1902072930, 1646926355, 473881174, 1583631138,
    677760056, 838789176, 827331457, 1322308383,
    1239059097, 233273648, 55471811, 929516509,
    2062955645, 1362536314, 218724220, 836843832,
    14387358, 1396572830, 1966472522, 1119098349,
    1443459654, 182029519, 812174379, 799647766],
[799647766, 1902072930, 1646926355, 473881174,
    1583631138, 677760056, 838789176, 827331457,
    1322308383, 1239059097, 233273648, 55471811,
    929516509, 2062955645, 1362536314, 218724220,
    836843832, 14387358, 1396572830, 1966472522,
    1119098349, 1443459654, 182029519, 812174379],
```

```
[812174379, 799647766, 1902072930, 1646926355,
    473881174, 1583631138, 677760056, 838789176,
    827331457, 1322308383, 1239059097, 233273648,
    55471811, 929516509, 2062955645, 1362536314,
    218724220, 836843832, 14387358, 1396572830,
    1966472522, 1119098349, 1443459654, 182029519]
],

"round_constants": [
1375440650, 2006515628, 1307113356, 2061079059,
    829381244, 2043791910,
737050957, 1716865126, 65313788, 1835882285, 1830670823,
    824712244,
1734845741, 683800643, 1922270414, 1731540232,
    343232192, 92890539,
1580935014, 1583575646, 613256790, 879354417, 240732658,
    1444904268,
467888043, 2010814686, 1440032228, 1176112725,
    352034784, 814573440,
450694358, 1362701228, 1534880589, 107285413,
    1947419414, 2005577384,
237833663, 822690442, 126334699, 1031058193, 2107440271,
    761070095,
1070662913, 1984467229, 2088826276, 325622001,
    1898397332, 1332515663,
920820281, 1677726488, 1114792620, 1081151082,
    620603566, 808461350,
1509814615, 287758160, 720789559, 1419973571,
    1996429723, 36719816,
717539722, 931940531, 1071096065, 1095595880, 98305972,
    981638424,
2007838239, 730373743, 430004662, 46985655, 357474420,
    821573088,
693263375, 708985213, 208599033, 1175368835, 10508391,
    950292472,
596069772, 443085597, 923381666, 1282977294, 1424291578,
    488920964,
574250972, 1776957247, 1076408408, 1621356654,
    1157544555, 590288075,
325860241, 432417979, 941200143, 1657354181, 910551236,
    1275638028,
1371360015, 1362106565, 1680335838, 1364440033,
    505233334, 1293343700,
627276923, 402859153, 1695790778, 682549161, 321338381,
    1539763154,
1764823365, 306396243, 180691448, 885515620, 1129179406,
    993762752,
146459656, 1155570785, 606245313, 1662214755,
    1860343269, 2017974904,
```

```
2070326376, 1377633836, 1517972258, 1929086552,
    1414558365, 589013796,
1236908192, 998558234, 743004231, 2111405240, 629623168,
    996162474,
211862180, 345661928, 253626003, 2039999680, 2105392940,
    872384229,
471913542, 2110957082, 130108089, 1824547067,
    1347908457, 1991761319,
1692527235, 1764500136, 275000619, 2032830522,
    1329714789, 190791201,
511694953, 1135573488, 1203386287, 1877649206,
    1274316507, 1220435904,
1955715157, 2081830053, 1777628021, 231655987,
    913935311, 678467935,
1440237105, 1215616240, 249147289, 25907082, 395409231,
    1754816108,
1848825488, 108161224, 1127654043, 11339609, 37478131,
    727168338,
926995068, 1310216028, 1095330553, 1988363527,
    669165917, 12469358,
1047404280, 1248230267, 901950508, 1484443862,
    1672374637, 652485505,
2113858549, 906249491, 1403459234, 892620539, 99968009,
    420493463,
2023585028, 1877272890, 964797184, 8595434, 305063999,
    1327428287,
1751443201, 1445983929, 2079988979, 407304094,
    378561405, 676222850,
676803779, 1325826764, 1764629096, 1124614894,
    1680282674, 1554746382,
2006589267, 1618902148, 36411953, 814619374, 947990061,
    1311185440,
1404942956, 216453746, 1501181937, 533539981, 443913418,
    1748605094,
11482849, 1894477904, 2102306526, 1490782527,
    2073456856, 114400667,
1497402479, 1495978443, 218234611, 494314761, 405188137,
    942530856,
1253066212, 1162654937, 362378118, 109564743, 420406686,
    1782971115,
148628368, 519191204, 234387588, 1886520298, 1640958398,
    1249164509,
1999771584, 102585347, 1217737970, 1167194973,
    1348906627, 2130034015,
881184819, 321211653, 1018313300, 2011368674, 828176348,
    2100982384,
81805248, 2032435089, 84018003, 231473643, 1772527487,
    1582071992,
92105546, 1469174772, 251837818, 554432516, 651850171,
    1254258715,
```

```
1254860762, 641920399, 1094671081, 1657576435,
    1921195356, 54626259,
1689633176, 744090968, 2083439999, 804721851,
    1361845900, 1474891367,
1061154393, 722232637, 516655453, 550091387, 10305708,
    281690027,
1190922736, 1638056084, 2010514454, 1627336277,
    1446427699, 1852221049,
1539630548, 1075052118, 1749060219, 434471592,
    1558818829, 773646847,
706598390, 1305523254, 1568147305, 1822448013,
    742433648, 238685717,
2020376837, 60012576, 1921742053, 1358172156, 697004195,
    1365666929,
1139828836, 159880438, 660411841, 1562029037,
    1024080620, 178739176,
796107648, 349773919, 1788806713, 548709606, 2103812072,
    458408978,
1754148496, 1605938887, 3298629, 1445719214, 2129883753,
    1175148922,
1301741306, 2000376032, 1791198614, 113054932,
    1345627296, 1713978880,
65509615, 1308720311, 1111942061, 324275240, 1331690460,
    48432528,
520018454, 716593084, 1054159915, 1925970212, 387095363,
    1292696544,
314127324, 771914575, 40532818, 1702299875, 2123077605,
    621470000,
1169140545, 911666495, 454335736, 372845031, 1907769840,
    613599248,
1479857277, 1611952666, 1419223255, 1760539472,
    783952320, 2088665843,
1576079773, 868737497, 174133596, 963588053, 1103831901,
    1612317056,
1736180226, 965154870, 42789145, 757051103, 985625319,
    854097174,
898969063, 206236596, 1451715271, 1015830025, 195566940,
    749520872,
605309623, 13981821, 947792113, 1825231464, 930257988,
    870724375,
1478055724, 1160384837, 1069586057, 1141537369,
    406729809, 912249570,
1249480558, 2128618080, 907881272, 218889889,
    1042544559, 136147037,
1255454138, 2032652291, 1207678284, 821806785,
    1689794697, 565958525,
476980311, 1468526601, 1722570174, 2052232893,
    1825326926, 1798489061,
1013184006, 1693498661, 636203913, 1216067890,
    1370241538, 228546411,
```

```
1956505242, 618402883, 782884991, 216700420, 1012556063,
    3643448,
709301117, 2116542810, 1776043605, 1802777563,
    1645980560, 1907855712,
669166925, 1546285702, 1701061417, 1604297602,
    1097511123, 989149709,
311364994, 1312076463, 882544017, 75048958, 1420512927,
    1579241323,
1939921351, 1880031112, 1910481233, 1321072638,
    1452938924, 45492890,
843210927, 146754154, 634142020, 1367707073, 1509994810,
    128697645,
47469873, 1480950334, 343105918, 1523876070, 707650115,
    1458820982,
1240120374, 1807749801, 1942884119, 1352548065,
    685009928, 1281646668,
621186234, 1646785313, 547850798, 1069019293,
    1304155729, 2075833097,
1653613623, 1947973011, 1188240372, 2035988797,
    1877337004, 199007455,
1523326776, 1385483299, 345208431, 1297649191,
    1158762927, 666828462,
1981587139, 1100153662, 769987809, 156844742,
    1720092270, 1799725859,
472097656, 905025350, 1851338145, 1129260141,
    1360456412, 361810061,
306736169, 1402527689, 1938706968, 551959222, 686109020,
    425275558,
1409610888, 959383538, 1894712763, 1184756353,
    1225177246, 103664933,
809257319, 1896928540, 1664257053, 943399262,
    1258020770, 1725179027,
619305449, 1273537183, 1910903253, 1277881484,
    662522103, 1831420635,
276870919, 705678512, 1964371725, 1760667213, 543338949,
    195573550,
1412127106, 2115287651, 1310648071, 1626673627,
    192110421, 114547034,
893378778, 1491429020, 2006168100, 1270089369,
    559087702, 508235099,
1381839369, 1259767241, 417697784, 1815283603,
    1199316140, 866345115,
689084659, 948284217, 2068734572, 478429979, 1740694864,
    1580514451,
621447509, 351797563, 1031725855, 547523025, 542433454,
    895845170,
1907517130, 808478325, 1829646990, 1986050017,
    502312932, 769240818,
1358629449, 1915261593, 509078963, 783301797,
    1513995405, 1945663361,
```

```
1597856784, 1349258375, 1854446343, 96450788, 509938789,
    1860257441,
2118310526, 579341490, 1261393803, 2118081247,
    1161594066, 729489583,
910069621, 893899107, 1175481257, 800258287, 1834117446,
    684418230,
300158745, 1075001136, 531291773, 870477804, 920834289,
    760149702,
1155933929, 226992181, 1957867554, 1473584116,
    1274358644, 334156855
],

"solutions": [
{
"input_state_1": [0, 0, 1457795403, 691853604,
    652846219, 972492827, 54730415, 1033489081,
    1057487049, 55332218, 2008831162, 803391250,
    60120468, 1877141981, 1101310796, 1015008570,
    2118294146, 350967989, 1805506356, 1978491379,
    1050641144, 1321122138, 2061189515, 792705724
],
"output_state_1": [0, 0, 261851485, 1271016679,
    794917173, 1945908895, 1771553859, 1256786360,
    2032727456, 1970456320, 1792835212, 1291679903,
    1087102302, 1976454217, 118112817, 39907320,
    575450740, 439309906, 817024562, 447250730,
    1723355579, 1728055708, 2101636512, 416863670
]
},
{
"input_state_2": [0, 0, 1735314245, 1841352588,
    1079178731, 1449777892, 1830693190, 519455216,
    1141715241, 1761495728, 534241941, 988816726,
    1032938004, 481855515, 1852367678, 2014031206,
    1116881964, 825418428, 588224898, 901960092,
    1562923937, 568780707, 2074165747, 662978623
],
"output_state_2": [0, 0, 1305529411, 1982269961,
    606831843, 1348257348, 1513694667, 966533932,
    1346936940, 686267266, 1049888163, 1639374355,
    1812747808, 763239598, 117202136, 1878699937,
    26650871, 802062188, 2027995774, 1428085040,
    1999265276, 30557785, 1201092072, 220358831
]
},
{
"input_state_3": [0, 0, 2058781021, 853366124,
    888172973, 1990114849, 390842797, 697717536,
    1726573462, 1056007407, 1294542897, 1677343919,
    61747403, 1363254340, 473218111, 2068308516,
```

```
        1547465630, 732655583, 688164922, 54801640,
        396092569, 1252126150, 1816585639, 1243034690
    ],
    "output_state_3": [0, 0, 1930039217, 1552856035,
        2034048381, 1788949406, 676840371, 77317022,
        1270121892, 329514189, 1362912084, 1646626772,
        1312187768, 37355629, 495440028, 691511502,
        1462575672, 1670616428, 909975655, 1340346915,
        1342699598, 1163776715, 2015885502, 2008245875
    ]
    },
    {
    "input_state_4": [0, 0, 56566861, 1176088222,
        1183329990, 1302601453, 1381433739, 634021088,
        101079865, 903260014, 1805536288, 1515839611,
        1466319052, 1007921967, 134480078, 1831918734,
        497050818, 1335120547, 456112231, 1404183782,
        627671124, 1700489335, 672315599, 577291986
    ],
    "output_state_4": [0, 0, 2015846302, 1202134334,
        1841379149, 1626676516, 1611882641, 1190521662,
        1871170171, 2034915643, 1182082188, 1843739253,
        1733199620, 1798918715, 891106801, 1839398893,
        197910175, 525776976, 535306888, 1417632967,
        114754208, 895888103, 99623796, 1592597732
    ]
  }
]
```


References

1. Linear layer tool (2021), https://extgit.isec.tugraz.at/krypto/linear-layer-tool
2. Poseidon-tools (2026), https://github.com/khovratovich/poseidon-tools
3. Ashur, T., Buschman, T., Mahzoun, M.: Algebraic cryptanalysis of the HADES design strategy: Application to Poseidon and Poseidon2. In: Australasian Conference on Information Security and Privacy. pp. 225–244. Springer (2024)
4. Bak, A., Bariant, A., Boeuf, A., Briaud, P., Øygarden, M., Phanse, A.: The Algebraic CheapLunch: Extending FreeLunch Attacks on Arithmetization-Oriented Primitives Beyond CICO-1. Cryptology ePrint Archive (2025)
5. Bak, A., Bariant, A., Boeuf, A., Hostettler, M., Jazeron, G.: Claiming bounties on small scale Poseidon and Poseidon2 instances using resultant-based algebraic attacks. Cryptology ePrint Archive, Paper 2026/150 (2026), https://eprint.iacr.org/2026/150
6. Bariant, A., Bouvier, C., Leurent, G., Perrin, L.: Algebraic attacks against some arithmetization-oriented primitives. IACR Transactions on Symmetric Cryptology pp. 73–101 (2022)

7. Berthomieu, J., Eder, C., Safey El Din, M.: msolve: A library for solving polynomial systems. In: Proceedings of the 46th International Symposium on Symbolic and Algebraic Computation (ISSAC '21). pp. 51–58. ACM (2021). https://doi.org/10.1145/3452143.3465545
8. Cantor, D.G., Zassenhaus, H.: A new algorithm for factoring polynomials over finite fields. Mathematics of Computation **36**(154), 587–592 (1981). https://doi.org/10.2307/2007663
9. Collins, G.E.: The calculation of multivariate polynomial resultants. Journal of the ACM (JACM) **18**(4), 515–532 (1971)
10. Faugere, J.C.: A new efficient algorithm for computing gröbner bases (f4). Journal of pure and applied algebra **139**(1-3), 61–88 (1999)
11. Faugère, J.C., Mou, C.: Sparse FGLM algorithms. Journal of Symbolic Computation **80**, 538–569 (2017)
12. Giorgi, P., Jeannerod, C.P., Villard, G.: On the complexity of polynomial matrix computations. In: Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (ISSAC). pp. 135–142. ACM (2003). https://doi.org/10.1145/860854.860889
13. Grassi, L., Khovratovich, D., Rechberger, C., Roy, A., Schofnegger, M.: Poseidon: A new hash function for Zero-Knowledge proof systems. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 519–535 (2021)
14. Grassi, L., Khovratovich, D., Schofnegger, M.: Poseidon2: A faster version of the Poseidon hash function. In: International Conference on Cryptology in Africa. pp. 177–203. Springer (2023)
15. Grassi, L., Koschatko, K., Rechberger, C.: Poseidon and Neptune: Gröbner Basis Cryptanalysis Exploiting Subspace Trails. IACR Transactions on Symmetric Cryptology **2025**(2), 34–86 (Jun 2025). https://doi.org/10.46586/tosc.v2025.i2.34-86, https://tosc.iacr.org/index.php/ToSC/article/view/12244
16. Grassi, L., Rechberger, C., Schofnegger, M.: Proving resistance against infinitely long subspace trails: How to choose the linear layer. IACR Transactions on Symmetric Cryptology pp. 314–352 (2021)
17. Kaltofen, E., Shoup, V.: Subquadratic-time factoring of polynomials over finite fields. In: Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing. pp. 398–407 (1995). https://doi.org/10.1145/225058.225167
18. National Institute of Standards and Technology: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Tech. Rep. Federal Information Processing Standards Publication (FIPS) 202, National Institute of Standards and Technology (2015). https://doi.org/10.6028/NIST.FIPS.202
19. Sanso, A., Vitto, G.: Top Gun: Degree Annihilation Attacks on Poseidon. Cryptology ePrint Archive, Paper 2026/1254 (2026), https://eprint.iacr.org/2026/1254
20. Vitto, G.: Slipway: Accessing Finite Subspace Trails in Poseidon. Cryptology ePrint Archive, Paper 2026/1579 (2026), https://eprint.iacr.org/2026/1579
21. Zhao, Z., Sanso, A., Vitto, G., Ding, J.: Graeffe-Based Attacks on Poseidon and NTT Lower Bounds. Cryptology ePrint Archive, Paper 2025/1916 (2025), https://eprint.iacr.org/2025/1916