Mind Small Integers in MDS Matrix: Collision Attacks on Round-Reduced Reinforced Concrete

Jiamin Cui[1,5,6], Fukang Liu[2(✉)], Jianqiang Ni[3], and Willi Meier[4]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
cuijiamin@sdu.edu.cn
[2] Institute of Science Tokyo, Tokyo, Japan
liufukangs@gmail.com
[3] Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, China
jianqiangni0213@163.com
[4] University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland
willimeier48@gmail.com
[5] State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Qingdao, 266237, Shandong, China
[6] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China

Abstract. The rapid advancement of Zero-Knowledge Proofs (ZKP) has motivated the design of ZK-friendly hash functions. A relatively new design strategy for ZK-friendly hash functions is using the composition of small look-up tables (LUTs) to build a nonlinear transform (called `Bars` layer) over a large prime field $\mathbb{F}_p$. This not only improves the plain performance but also enhances its security against algebraic attacks since the `Bars` layer is equivalent to a complex and high-degree polynomial over $\mathbb{F}_p$. This design strategy was first proposed at CCS 2022 for the ZK-friendly hash function Reinforced Concrete. However, there has been no third-party collion attacks of Reinforced Concrete since then, and all existing attacks on such LUT-based ZK-friendly ciphers like Tip5, Monolith and Skyscraper mainly exploit the differential or linear properties of `Bars`.

In this paper, we demonstrate that the `Bars` layer surrounded by MDS matrices (called `Concrete` layers) with small integers, i.e., `Concrete` $\circ$ `Bars` $\circ$ `Concrete`, might lead to weaker differential properties of the round function. This key observation leads to a novel collision attack on 4.5 out of 7 rounds of Reinforced Concrete, successfully bypassing the `Bars` layer as well as the subsequent `Concrete` layer and power-map-based nonlinear layer (called `Bricks` layer). The attack is verified by providing a practical collision for 3.5-round Reinforced Concrete where the last 4 layers are `Bricks` $\circ$ `Concrete` $\circ$ `Bars` $\circ$ `Concrete`.

Furthermore, we provide constructive countermeasures by designing an algorithm to generate an improved `Concrete` layer that thwarts this specific attack without increasing the number of arithmetic constraints compared to the original design. Finally, we also applied our framework

to Tip5 but found that its MDS matrix can effectively prevent this attack. This is the first time to exploit the weak combination of `Bars` layer and `Concrete` layer to mount efficient attacks on LUT-based ZK-friendly hash functions. We believe that it sheds new insight into the security of these hash functions.

Keywords: ZK-friendly · Hash functions · Lookup table · MDS matrix · Collision attacks · Reinforced Concrete

1 Introduction

Zero-knowledge proofs (ZKP) have attracted significant attention in recent years due to their growing deployment in real-world systems such as blockchain, verifiable computation, and digital identity. In these protocols, cryptographic hash functions are fundamental building blocks, particularly in set membership proofs based on Merkle tree accumulators and verifiable computation based on recursive proofs. While classical hash functions like SHA-3 [26] benefit from extensive community scrutiny and offer strong security guarantees optimized for conventional software and hardware, they are not optimized for arithmetic circuits and therefore tend to underperform in ZKP settings. To bridge this gap, a novel class of ZK-friendly hash functions operating natively over large finite fields has been proposed. One common design strategy is to minimize the number of multiplications, as seen in MiMC [3], GMiMC [2], Poseidon [22], Poseidon2 [23] and Neptune [24]. Another approach aims to achieve low-degree descriptions for round functions that enable highly efficient constraint systems. Examples are Vision/Rescue [4], Rescue-Prime [5,31] Anemoi [11], and Griffin [19].

While these primitives are efficient in ZKP protocols, their plain performance remains unsatisfactory as field multiplications are still relatively expensive compared to simple bit operations. Recently, starting with the introduction of Plookup [18,28], the set of constraints expanded beyond simple arithmetic expressions to include efficient lookup tables (LUT) [27,30]. This new type of constraints inspires a novel design strategy for ZK-friendly hash functions based on LUTs, which simultaneously increases the algebraic degree while maintaining high efficiency in native implementations. The core idea is to split a field element over $\mathbb{F}_p$ into small chunks over fields or rings, apply small LUT-based permutations to each chunk, and then recombine the results into an element of the initial field. The small LUT-based permutations often translate into extremely complicated algebraic representations when lifted back to $\mathbb{F}_p$, posing significant challenges for cryptanalysis. This strategy was first introduced in Reinforced Concrete [20], and is subsequently applied in other primitives such as Monolith [21], Tip5 [32], Tip4/Tip4' [29] and Skyscraper [12,13].

Related works. The exploration of LUT-based designs necessitates advanced cryptanalytic methods to establish strong security guarantee. These hash functions, often instantiated in sponge construction, require to satisfy preimage

and collision resistance. The common cryptanalytic strategy is to model the attacks as a system of equations and try to solve it. However, when constructing the equation system to represent collision attacks, the major bottleneck arises from the inherently complicated and high-degree algebraic representation of the LUT-based S-boxes. Instead of directly handling these LUT-based S-boxes by guess-and-determine, Liu et al. introduced an efficient algorithm for analyzing the differential properties of LUT-based S-boxes in FSE 2025 [25]. By efficiently identifying non-zero input/output difference pairs for LUT-based S-boxes satisfying specific conditions, more degrees of freedom can be strategically exploited, thereby significantly enhancing the efficiency of collision attacks. Using this framework, they successfully presented 3-round collision attacks on Tip5/Tip4/Tip4' and 2-round collision attacks on Monolith. For semi-free-start (SFS) collision attacks, they further achieved 4-round attacks on Tip4 and 3-round attacks on Monolith-64.

The linear property of the LUT-based S-boxes was evaluated in ToSC 2025 [7]. For the S-box of Monolith-64, a linear approximation with an unexpectedly high probability of $2^{-22.24}$ was identified, which was subsequently exploited in the limited-birthday attacks in the work [6]. A root-finding-aided rebound attack was also proposed in [6]. For Skyscraper-v1, this attack results in a full-round truncated differential distinguisher for the permutation, and a collision attack on the compression mode for 9-round permutation. In [7], the authors refined a technique for handling the first round. By combining this technique with the discovery of a partial preimage, they mounted a collision attack on 4-round Tip5, albeit in the setting where the capacity is reduced from six to five field elements. In particular, for the first proposal Reinforced Concrete, no practical progress has been reported so far.

Motivation. For primitives over smaller fields, such as 64-bit or 31-bit prime fields, achieving 128 bits of security requires a larger number of field elements for the capacity and state size. This provides additional degrees of freedom when constructing the corresponding equation systems, as demonstrated in recent attacks on Tip5, Tip4, Tip4' and Monolith [6,7,25]. In contrast, Reinforced Concrete operates on prime fields of approximately 256 bits while targeting the same 128-bit security level. It employs a highly compact state of only 3 elements with a capacity of a single field element. While this structure restricts the problem to a univariate equation system, the `Bars` layer which is composed of three 256-bit LUT-based S-boxes is extremely difficult to bypass advantageously. Although previous work [7] has attempted to evaluate the linear approximation of `Bars`, the analysis was limited to fixed points. For the specific case of prime $p = p_{\text{BLS381}}$, the best-known linear approximation holds with a probability of approximately $2^{-114.25}$, which is far too low to mount an effective attack.

Since this compactness of the state severely restricts the ability of the attacker, it becomes a rational design choice to rely primarily on the complicated algebraic representations of `Bars` to provide cryptographic strength, and instead optimize the linear layer for performance. In this paper, we reveal that this trade-off

introduces a subtle structural vulnerability: the security of the LUT-based S-boxes is highly contingent upon the surrounding linear layer, even when the algebraic representation of the S-box itself remains complicated.

Our contribution. We establish a novel attack framework based on the observation of `Bars` surrounded by MDS matrices to evaluate the security of LUT-based ZK-friendly primitives. Our contributions are summarized as follows.

1. We show that the security of LUT-based S-boxes highly depends on the surrounding linear layer. We reveal a structural vulnerability specific to Reinforced Concrete: the small-integer construction of the `Concrete` layer fails to provide adequate diffusion across the mixed-radix decomposition. This allows an attacker to strategically find high-probability difference transitions across `Bars` surrounded by MDS matrices.
2. Based on this insight, we propose a novel attack framework for Reinforced Concrete by activating a minimal number of LUTs within `Bars`. By precisely controlling difference propagation to enforce sparse LUT activation, we successfully give the first collision attacks on 4.5-rounds (out of 7 rounds) of Reinforced Concrete and the first practical collision on 3.5-rounds Reinforced Concrete. By extending the attack to utilize a multi-block strategy, we increase the success probability from around 0.6 to over 0.9 with only a threefold increase in time complexity. Our results are shown in Table 1.
3. We propose a searching algorithm to evaluate the vulnerability of the `Concrete` layer against our attack framework. By explicitly computing the set of insecure parameters, we provide a mathematical guarantee that no valid single-active S-box differential characteristic exists for the refined parameters. This allows us to generate improved `Concrete` layers that thwart this specific attack.
4. We apply our framework to Tip5, evaluating its security bounds based on the minimum number of active LUTs. We show that the strong diffusion of the MDS linear layer effectively prevents the exploitation of partial S-boxes, forcing a dense activation pattern with a lower bound of 12 active LUTs, thereby rendering the attack computationally infeasible.

Outline. The rest of the paper is organized as follows. In Section 2, we introduce some background knowledge needed in this paper. In Section 3, we present our new insight into the linear layer used in Reinforced Concrete, and show how its small-integer structure can be exploited to find high-probability difference transitions for `Bars`. In Section 4, we show how to activate a minimal number of LUTs in the `Bars` layer. Then, we propose our attack framework for 4.5-round collision attacks on Reinforced Concrete in Section 5. In Section 6, we investigate countermeasures that can thwart our attacks. In Section 7, we apply the proposed framework to Tip5 to evaluate its security. The paper is concluded in Section 8.

Table 1: Summary of our collision attacks on Reinforced Concrete.

Instance	#Rounds	Time Complexity	$\omega = 2.37$	Success Prob.
p_{BLS381}	3.5	$\max\{2^{11.99\omega+23.93}, 2^{52.47}\}$ (practical)	$2^{52.47}$	0.66
	4.5	$\max\{2^{19.62\omega+23.93}, 2^{76.23}\}$	$2^{76.23}$	0.66
	4.5	$\max\{2^{19.62\omega+25.52}, 2^{77.81}\}$	$2^{77.81}$	0.96
p_{BN254}	3.5	$\max\{2^{11.99\omega+23.80}, 2^{52.35}\}$ (practical)	$2^{52.35}$	0.66
	4.5	$\max\{2^{19.62\omega+23.80}, 2^{76.10}\}$	$2^{76.10}$	0.66
	4.5	$\max\{2^{19.62\omega+25.39}, 2^{77.68}\}$	$2^{77.68}$	0.96
p_{ST}	3.5	$\max\{2^{10.59\omega+24.98}, 2^{50.34}\}$ (practical)	$2^{50.34}$	0.61
	4.5	$\max\{2^{15.35\omega+24.98}, 2^{64.60}\}$	$2^{64.60}$	0.61
	4.5	$\max\{2^{15.35\omega+26.56}, 2^{66.19}\}$	$2^{66.19}$	0.94

2 Preliminary

2.1 Notation

The following notations will be used throughout this paper.

- $\mathbb{F}_p$ denotes the finite field where p is a prime.
- $\mathbb{Z}_s$ denotes the ring of integers modulo s.
- $\mathsf{R} = 2^{64} \bmod 2^{64} - 2^{32} + 1$ and $\mathsf{R}^{-1} = (2^{64})^{-1} \bmod 2^{64} - 2^{32} + 1$ are constants.
- For a matrix M of size $n \times n$ and a vector $x = (x_0, \cdots, x_{n-1})^T$, $(M(x))_i$ represents the i-th entry of $M \times x$ for $0 \leq i < n$.
- $\text{Diag}_{a,n,s}$ denotes a diagonal matrix $\text{Diag}(\underbrace{a, \cdots, a}_{s}, \underbrace{1, \cdots, 1}_{n-s})$.

2.2 Sponge Construction

The sponge construction [9] operates on an internal state consisting of n field elements over $\mathbb{F}_p$, which can be divided into two parts: an outer part with a rate of r elements and an inner part with a capacity of c elements. In the absorbing phase, the message is split into chunks $M_i \in \mathbb{F}_p^r$ and added to (or used to overwrite) the first r field elements of the internal state. The squeezing phase produces the hash value $h_i \in \mathbb{F}_p^d$.

For Reinforced Concrete, odd-length messages are padded with a single 0 element (over $\mathbb{F}_p$). Then, the very first capacity value is initialized with a length-depending constant.

2.3 General Description of LUT-based S-box

To unify the algebraic framework, we provide a general description of the LUT-based S-box [20, 21, 32] following the work of [7, 25]. We define S as a LUT-based S-box over $\mathbb{F}_p$. Let $(s_1, s_2, \cdots, s_m)$ denote a tuple of integer parameters

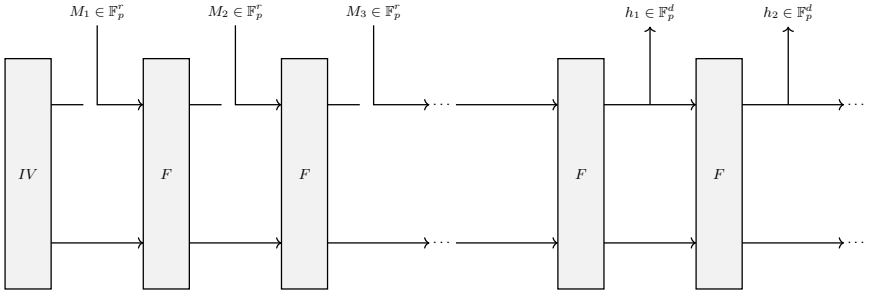

Fig. 1: Sponge construction.

satisfying the condition $\prod_{i=1}^{m} s_i > p$ and $\lceil \sum_{i=1}^{m} \log_2(s_i) \rceil = \lceil \log_2(p) \rceil$. The bases $(b_1, b_2, \cdots, b_m)$ are defined as

$$b_m = 1,\ b_i := \prod_{j>i} s_j = s_{i+1} s_{i+2} \cdots s_m, 1 \leq i < m.$$

The LUT-based S-box $y = S(x) = \mathcal{C} \circ \mathcal{S} \circ \mathcal{D}(x)$ over $\mathbb{F}_p$ is constructed via the following three steps:

- **Decomposition** $\mathcal{D}$: Let $\mathcal{D}(x) := (x_1, x_2, \cdots, x_m)$. The element $x \in \mathbb{F}_p$ is represented as $(x_1, x_2, \ldots, x_m) \in \mathbb{Z}_{s_1} \times \mathbb{Z}_{s_2} \times \cdots \times \mathbb{Z}_{s_m}$ such that

$$x = x_1 \cdot \prod_{i=2}^{m} s_i + x_2 \cdot \prod_{i=3}^{m} s_i + \cdots + x_{m-1} \cdot s_m + x_m = \sum_{i=1}^{m} x_i b_i,$$

  where $0 \leq x_i < s_i$.

- **Permutation** $\mathcal{S}$: Let $(y_1, \cdots, y_m) = \mathcal{S}(x_1, \cdots, x_m)$. For each component i, we compute $y_i = S_i(x_i)$, where S_i is defined as a LUT-based permutation over $\mathbb{Z}_{s_i}$. The permutations $S_1, \ldots, S_m$ are chosen carefully so that S is a permutation over $\mathbb{F}_p$.

- **Composition** $\mathcal{C}$: Let $y = \mathcal{C}(y_1, \cdots, y_m)$. To compose $(y_1, \cdots, y_m) \in \mathbb{Z}_{s_1} \times \mathbb{Z}_{s_2} \times \cdots \times \mathbb{Z}_{s_m}$ back into an element $y \in \mathbb{F}_p$, we have $y = \sum_{i=1}^{m} y_i \prod_{j>i} s_j = \sum_{i=1}^{m} y_i b_i$.

2.4 Description of Reinforced Concrete and Tip5 Family

Reinforced Concrete. The Reinforced Concrete hash function was proposed at CCS 2022 [20]. It applies the sponge construction with $(r, c, d) = (2, 1, 2)$ and $IV = (0, 0, 0)$. The underlying permutation operates over $\mathbb{F}_p^3$. In the specification, the authors instantiate $\mathbb{F}_p$ using 3 options for prime p, namely $p = p_{\text{BLS381}}$, $p = p_{\text{BN254}}$ and $p_{\text{ST}} = 0x3fa000...001$. For the Reinforced Concrete hash function, the security claim against collision attacks is 2^{128} field operations.

The underlying permutation, hereafter referred to as Reinforced Concrete, is illustrated in Figure 2. It can be viewed as a modified 7-round SPN-based permutation over $\mathbb{F}_p^3$ such that

$$RC := (M_8 \circ B) \circ (M_7 \circ B) \circ (M_6 \circ B) \circ (M_5 \circ \texttt{Bars} \circ M_4) \circ (B \circ M_3) \circ (B \circ M_2) \circ (B \circ M_1).$$

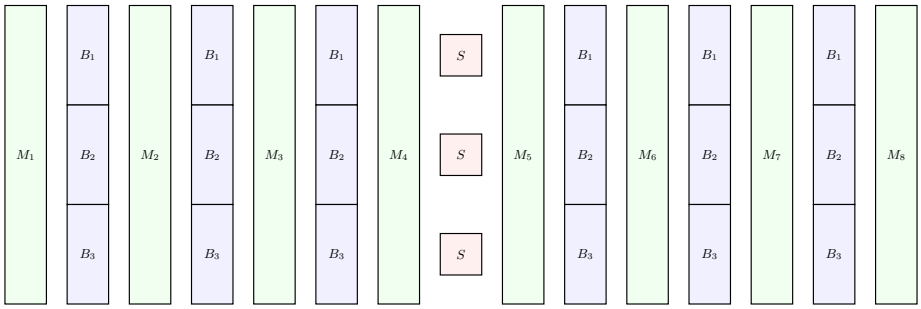

Fig. 2: Overview of Reinforced Concrete.

In this paper, we equivalently regard the middle layer $(M_5 \circ \texttt{Bars} \circ M_4)$ as a single round. Accordingly, each $(B \circ M_i)$ for $i < 4$ is regarded as one round before this middle round, and each $(M_i \circ B)$ for $i > 5$ is regarded as one round after it. The permutation consists of three types of layers: the `Concrete` layer M_i, the Bricks layer B, and the Bars layer `Bars`:

- **`Concrete` layer** M_i: $M_i \colon \mathbb{F}_p^3 \to \mathbb{F}_p^3$ is an affine layer defined as $M_i(x) = M \cdot x + c_i$, where M is a 3×3 MDS matrix defined as

$$M = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix},$$

  and $c_i \in \mathbb{F}_p^3$ denotes the i-th round constant.

- **`Bricks` layer** B: The Bricks layer $(y_1, y_2, y_3) = B(x_1, x_2, x_3) \in \mathbb{F}_p^3$ is defined as:

$$\begin{aligned} y_1 = B_1(x_1) &= x_1^d, \\ y_2 = B_2(x_1, x_2) &= x_2(x_1^2 + \alpha_1 x_1 + \beta_1), \\ y_3 = B_3(x_2, x_3) &= x_3(x_2^2 + \alpha_2 x_2 + \beta_2), \end{aligned}$$

  where $\gcd(p-1, d) = 1$, $\alpha_i, \beta_i \in \mathbb{F}_p$. $\alpha_i^2 - 4 \cdot \beta_i$ is a non-quadratic residue mod p for $i = 1, 2$.

– **Bars layer `Bars`**: The Bars layer $\texttt{Bars} : \mathbb{F}_p^3 \to \mathbb{F}_p^3$ is defined as $(y_1, y_2, y_3) = (S(x_1), S(x_2), S(x_3))$. $S(x) = \mathcal{C} \circ \mathcal{S} \circ \mathcal{D}(x)$ denotes the LUT-based S-box described previously. The parameters $(s_1, s_2, \cdots, s_m)$ are defined separately for different primes p such that $\prod_{i=1}^{m} s_i > p$. Let $(v_1, v_2, \cdots, v_m) = \mathcal{D}(p-1)$. p' is a prime that satisfies $p' \leq \min_{1 \leq i \leq m} v_i$. Then for each i, the local S-box $S_i(x_i)$ is defined as:

$$y_i = S_i(x_i) = \begin{cases} f(x_i), & \text{if } x_i < p'; \\ x_i, & \text{if } x_i \geq p'. \end{cases}$$

f denotes a permutation over $\mathbb{F}_{p'}$ with high degree and dense polynomial representation.

The specific parameters for the `Bricks` layer and the `Bars` layer depend on the chosen prime field $\mathbb{F}_p$. We summarize these parameter sets in Table 2.

Table 2: Parameters for different Reinforced Concrete instantiations.

Instance	d	$(\alpha_1, \alpha_2, \beta_1, \beta_2)$	p'	m	$(s_m, \ldots, s_1)$
p_{BLS381}	5	$(1, 3, 2, 4)$	659	27	693, 696, 694, 668, 679, 695, 691, 693, 700, 688, 700, 694, 701, 694, 699, 701, 701, 701, 695, 698, 697, 703, 702, 691, 688, 703, 679
p_{BN254}	5	$(1, 3, 2, 4)$	641	27	651, 658, 656, 666, 663, 654, 668, 677, 681, 683, 669, 681, 680, 677, 675, 668, 675, 683, 681, 683, 683, 655, 680, 683, 667, 678, 673
p_{ST}	3	$(1, 2, 3, 4)$	1013	25	$s_{25} = 1023$, $s_i = 1024$ for $i \in \{1, \ldots, 24\}$

Tip5 family. Tip5 is a family of hash functions optimized for zero-knowledge proof systems [32]. It operates over $\mathbb{F}_p^n$, where $p = 2^{64} - 2^{32} + 1$. The original Tip5 utilizes a 5-round permutation and targets a security level of 160 bits with $(r, c, d) = (10, 6, 5)$. In [29], two additional instantiations, denoted as Tip4 and Tip4′, were proposed to target a 128-bit security level. We have $(r, c, d) = (12, 4, 4)$ for Tip4 and $(r, c, d) = (8, 4, 4)$ for Tip4′.

Following the equivalent description provided in [25], the j-th round function $F_j : \mathbb{F}_p^n \to \mathbb{F}_p^n$ can be defined as

$$F_j(x) := c_j + M_j(\mathbf{S}(x)), 1 \leq j \leq 5.$$

The components of the round function are defined as follows:

– **S-box layer S**: The S-box layer $\mathbf{S}\colon \mathbb{F}_p^n \to \mathbb{F}_p^n$ is defined as $(y_1, \cdots, y_n) = \mathbf{S}(x_1, \cdots, x_n)$. For $1 \leq i \leq n$ we have

$$y_i = \begin{cases} S(x_i), & \text{for } 1 \leq i \leq s, \\ T(x_i) = x_i^7, & \text{for } s < i \leq n. \end{cases}$$

s denotes the number of lookup-based S-boxes per round. $S(x) = \mathcal{C} \circ \mathcal{S} \circ \mathcal{D}(x)$ represents the LUT-based S-box described previously, instantiated with parameters $s_k = 2^8$ for $1 \leq k \leq m$. The internal permutation S_k applied to each decomposed chunk $z \in \mathbb{Z}_{2^8}$ is defined as:

$$S_k(z) = (z+1)^3 - 1 \pmod{2^8+1}.$$

– **Linear layer** M_i: $M_i(x) := M_i \times x$, $x = (x_1, \cdots, x_n)^T$. $M_i = \mathrm{Diag}_{\mathsf{R},n,s} \times M_0 \times \mathrm{Diag}_{\mathsf{R}^{-1},n,s}$ for $1 \leq i < 5$ and $M_5 = M_0 \times \mathrm{Diag}_{\mathsf{R}^{-1},n,s}$. $M_0 \in \mathbb{F}_p^{n \times n}$ is a circulant MDS matrix.

– **Round constants** c_i: $c_i = (c_{i,1}, c_{i,2}, \cdots, c_{i,n}) \in \mathbb{F}_p^n$ denotes the i-th round constants.

2.5 Gröbner Basis Attacks

Consider a system with n polynomials $\{f_1, \cdots, f_n\}$ in m variables, where $\deg(f_i)$ denotes the degree of f_i. Let $\mathcal{I} := \langle f_1, \cdots, f_n \rangle$ be the ideal generated by these polynomials and let $d_{\mathcal{I}}$ denote the degree of the ideal. We assume $\mathcal{I}$ to be zero-dimensional. In this paper, we focus on the case $m = n$.

The general approach to solving a multivariate system of equations using Gröbner basis follows the three main steps:

1. Compute a *grevlex* order Gröbner basis using the F4 algorithm [17]. The complexity of this step is bounded by $\binom{n+d_{\mathrm{reg}}}{d_{\mathrm{reg}}}^{\omega}$ arithmetic operations over $\mathbb{F}_p$, where d_{reg} denotes the *degree of regularity* and $2 \leq \omega < 3$ is the linear algebraic exponent.

2. Convert the *grevlex* order Gröbner basis to a *lexicographic* order one using the FGLM algorithm [15], which is bounded by $n \cdot {d_{\mathcal{I}}}^3$ arithmetic operations over $\mathbb{F}_p$. The complexity of this step is bounded by $n \cdot {d_{\mathcal{I}}}^{\omega}$ arithmetic operations over $\mathbb{F}_p$ when using a sparse FGLM algorithm [16].

3. Find the roots of the Gröbner basis polynomials using univariate system resolution [14] and substitution. For a univariate polynomial f of degree d, the complexity is approximately $d \log(d) \cdot (\log(d) + \log(p)) \cdot \log(\log(d))$ arithmetic operations over $\mathbb{F}_p$.

Under the hypothesis that the polynomial system is regular, the degree of regularity and the degree of the zero-dimensional ideal $\mathcal{I}$ are upper bounded by

the *Macaulay bound* and the *Bézout bound*, respectively:

$$d_{\text{reg}} \leq 1 + \sum_{i=1}^{n}(\deg(f_i) - 1),\ d_{\mathcal{I}} \leq \prod_{i=1}^{n} \deg(f_i).$$

On the choice of ω. Choosing a concrete value of ω is a challenging problem for algebraic attacks. Generally, it is common practice to use $\omega = 2.8$ or $\omega = 2.37$. In some cases, $\omega = 2$ also hold in practice [1, 10]. Our experiments suggest that $\omega = 2.37$ does not appear to underestimate. For FGLM, we use $\omega = 3$ although optimized algorithms may achieve $\omega < 3$.

3 Difference Transitions through $\texttt{Concrete} \circ \texttt{Bars} \circ \texttt{Concrete}$

The main algebraic barrier in Reinforced Concrete is $\texttt{Concrete} \circ \texttt{Bars} \circ \texttt{Concrete}$ and bypassing it with a high-probability difference transition is technical. Without loss of generality, we define

$$(y_1, \ldots, y_n) = M'_{n\times n} \circ \texttt{Bars}_n \circ M_{n\times n}(x_1, \ldots, x_n)$$

where

- $M_{n\times n}$ and $M'_{n\times n}$ denotes the $\texttt{Concrete}$ layers over $\mathbb{F}_p^n$. Specifically, $M_{n\times n}(x) = M \cdot x + c$ and $M'_{n\times n} = M \cdot x + c'$, where M is an $n \times n$ MDS matrix over $\mathbb{F}_p$ and $c, c' \in \mathbb{F}_p^n$.
- $\texttt{Bars}_n : \mathbb{F}_p^n \mapsto \mathbb{F}_p^n$ is defined as

  $$(u_1, \ldots, u_n) \in \mathbb{F}_p^n \mapsto (S(u_1), \ldots, S(u_n)) \in \mathbb{F}_p^n,$$

  where $S(u_i) : \mathbb{F}_p \mapsto \mathbb{F}_p$ is the LUT-based S-box, i.e.,

  $$(u_{i,1}, \ldots, u_{i,m}) = \mathcal{D}(u_i),\ S(u_i) = \mathcal{C}(S_1(u_{i,1}), \ldots, S_m(u_{i,m})).$$

- $(x_1, \ldots, x_n) \in \mathbb{F}_p^n, (y_1, \ldots, y_n) \in \mathbb{F}_p^n$ are the input and output of the 3 consecutive layers.

The overview of the construction is shown in Figure 3. For Reinforced Concrete, we have $n = 3$, $\texttt{Bars}_3 = \texttt{Bars}$. For $M'_{3\times 3}$ and $M_{3\times 3}$,

$$M = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

There are several attacks on such ZK-friendly hash functions exploiting the weak differential property of the LUT-based S-box S to only bypass $\texttt{Bars}_n$. Specifically, attackers can efficiently find a valid input difference $(\Delta u_1, \ldots, \Delta u_n)$ and output difference $(\Delta v_1, \ldots, \Delta v_n)$ for the layer $\texttt{Bars}_n$, i.e.,

$\exists (u_1, \ldots, u_n) \in \mathbb{F}_p^n$:

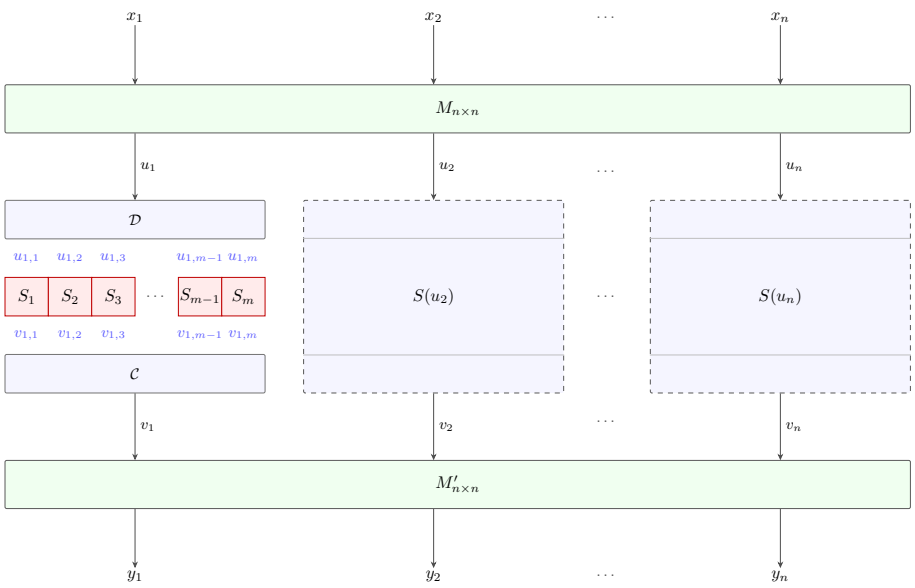

Fig. 3: Overview of our construction $M'_{n\times n} \circ \texttt{Bars}_n \circ M_{n\times n}$.

$$(\Delta v_1, \ldots, \Delta v_n) = (S(u_1 + \Delta u_1), \ldots, S(u_n + \Delta u_n)) - (S(u_1), \ldots, S(u_n)).$$

Therefore, for any given input difference $(\Delta x_1, \ldots, \Delta x_n)$ and output difference $(\Delta y_1, \ldots, \Delta y_n)$ of these 3 consecutive layers, we can efficiently check whether they correspond to a valid difference transition. Specifically, we can compute $M_{n\times n}(\Delta x_1, \ldots, \Delta x_n)$ and $M'^{-1}_{n\times n}(\Delta y_1, \ldots, \Delta y_n)$ to get the input and output difference of $\texttt{Bars}_n$. Then, we can use the algorithm in [25] to efficiently check whether they are valid.

The technical issue and our new insight into the `Concrete` layer. However, although it is possible to find a valid difference transition through these 3 consecutive layers by trying enough random input-output difference pairs (or specially chosen pairs), the differential probability might be quite low.

For our collision attack on round-reduced Reinforced Concrete, we want to find a valid input-output difference transition through these 3 consecutive layers satisfying

$$\Delta x_1 = 0, \Delta y_2 = 0, \Delta y_3 = 0, \Delta y_1 \neq 0. \tag{1}$$

Then, the input difference and output difference of `Bars` become

$$(\Delta u_1, \Delta u_2, \Delta u_3) = M(0, \Delta x_2, \Delta x_3),\ (\Delta v_1, \Delta v_2, \Delta v_3) = M^{-1}(\Delta y_1, 0, 0).$$

Since M is an MDS matrix, if it is a valid difference transition, there must be

$$\Delta v_i \neq 0, \Delta u_i \neq 0 \text{ for } 1 \leq i \leq 3.$$

This implies that almost all LUT-based S-boxes over $\mathbb{F}_p$ must be active (i.e., with a nonzero input difference) for any such valid difference transition. If such a difference transition is required to hold with high probability Pr, this might be quite challenging.

Let us elaborate more on why it is challenging. Then, it will be clear why our attack on Reinforced Concrete is nontrivial and is based primarily on our new insight into the `Concrete` layer $M_{n\times n}$. As illustrated in Figure 4, we consider the decomposition and $(u_i + \Delta u_i, v_i + \Delta v_i)$:

$$(u_{i,1}, \ldots, u_{i,m}) = \mathcal{D}(u_i)\ ,\ (v_{i,1}, \ldots, v_{i,m}) = \mathcal{D}(v_i),$$
$$(u'_{i,1}, \ldots, u'_{i,m}) = \mathcal{D}(u_i + \Delta u_i)\ ,\ (v'_{i,1}, \ldots, v'_{i,m}) = \mathcal{D}(v_i + \Delta v_i).$$

By definition, we have

$$S_j(u_{i,j}) = v_{i,j}, S_j(u'_{i,j}) = v'_{i,j} \text{ for } 1 \le j \le m.$$

In addition, S_j is a permutation over $\mathbb{Z}_{s_j}$ where $v_{i,j} \in \mathbb{Z}_{s_j}$.

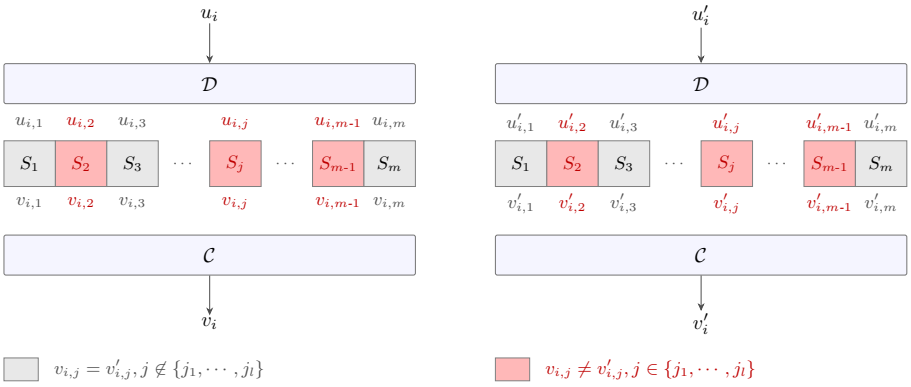

Fig. 4: Decomposition of (u_i, v_i) and (u'_i, v'_i).

On the challenge. Assume that there are ℓ indices $j_1, \ldots, j_\ell$ such that

$$v'_{i,j} \neq v_{i,j} \text{ if and only if } j \in \{j_1, \ldots, j_\ell\}.$$

We must have

$$u'_{i,j} \neq u_{i,j} \text{ if and only if } j \in \{j_1, \ldots, j_\ell\}.$$

And we say that there are ℓ active LUTs when computing $v_i = S(u_i)$ and $v_i + \Delta v_i = S(u_i + \Delta u_i)$ (seen Figure 4). In this case, there will be about $\prod_{j\in\{j_1,\ldots,j_\ell\}} s_j$ possible values of Δu_i and Δv_i since $u_{i,j}, v_{i,j} \in \mathbb{Z}_{s_j}$.

Note that the integers $s_1, \ldots, s_m$ are almost equal and close to $p^{1/m}$. Let ℓ_i be the number of active LUTs when computing $v_i = S(u_i)$ and $v_i + \Delta v_i = S(u_i + \Delta u_i)$ as shown in Figure 5. Hence, there are about

$$\binom{m}{\ell_1} \times \binom{m}{\ell_2} \times \binom{m}{\ell_3} \times (p^{1/m})^{\ell_1+\ell_2+\ell_3}$$

possible values of such $(\Delta u_1, \Delta u_2, \Delta u_3)$ and $(\Delta v_1, \Delta v_2, \Delta v_3)$. On the other hand, $(\Delta v_1, \Delta v_2, \Delta v_3) = M^{-1}(\Delta y_1, 0, 0)$ can take at most $p-1$ different values for $\Delta y_1 \neq 0$. Hence, if M^{-1} is a random matrix, the probability that we can expect to have $\ell_1 + \ell_2 + \ell_3$ active LUTs (even if we do not care about the condition $\Delta x_1 = 0$) is about

$$(p-1) \cdot \frac{\binom{m}{\ell_1} \times \binom{m}{\ell_2} \times \binom{m}{\ell_3} \times (p^{1/m})^{\ell_1+\ell_2+\ell_3}}{(p-1)^3} \approx \frac{\binom{m}{\ell_1} \times \binom{m}{\ell_2} \times \binom{m}{\ell_3} \times (p^{1/m})^{\ell_1+\ell_2+\ell_3}}{p^2}.$$

To have the highest differential probability, we have $\ell_1 = \ell_2 = \ell_3 = 1$. In this case, the probability will be about $m^3 \cdot p^{\frac{3}{m}-2}$, which is much smaller than p^{-1} in the case of Reinforced Concrete. Hence, if M is a random matrix, we do not expect to find a valid input-output difference of the 3 consecutive layers that have only 3 active LUTs.

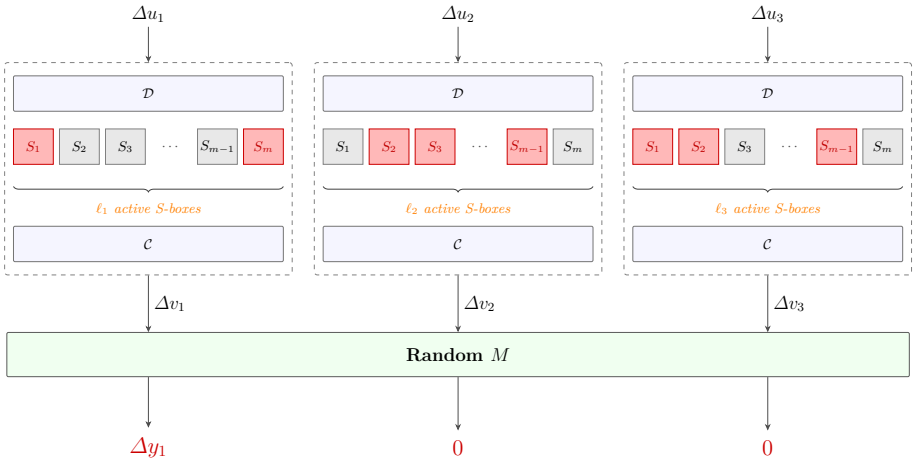

Fig. 5: Differential propagation in `Concrete` $\circ$ `Bars` when activating (ℓ_1, ℓ_2, ℓ_3) LUTs. $(\Delta v_1, \Delta v_2, \Delta v_3) \to (\star, 0, 0)$ through a random M holds with probability $\ll p^{-1}$.

The general case. The above analysis also applies to the general case where the matrix is of size $n \times n$. Then, for the output difference of the 3 consecutive layers:

$$(\Delta y_1 \neq 0, \underbrace{0, \ldots, 0}_{n-1}),$$

the probability that there are $\ell_1 + \ldots + \ell_n$ active LUTs is estimated as

$$\frac{(p^{1/m})^{\sum_{i=1}^{n} \ell_i} \cdot \prod_{i=1}^{n} \binom{m}{\ell_i}}{p^{n-1}}.$$

To find a valid difference transition, we have not yet considered the condition on the input difference $(\Delta x_1, \ldots, \Delta x_n)$, which will only make such a probability much lower. In particular, there it should be impossible to have a valid high-probability difference transition that only activates n LUTs, i.e., $\ell_i = 1$ for $1 \leq i \leq n$.

***M* is not random in Reinforced Concrete.** However, the MDS matrix M used in Reinforced Concrete is fixed and composed of small integers 1 and 2. This might make it different from the random case. And we will exploit this feature to construct a high-probability difference transition through the middle 3 consecutive layers satisfying Section 3. The details of how to construct such a high-probability transition with activating the lowest number of LUTs will be explained in Section 5. Here, we can give an intuitive example for better understanding. For example, we assume

$$\forall v_2 \in \mathbb{F}_p : \; M(\Delta v_1, \Delta v_2, \Delta v_3) = M(-3\Delta v_2, \Delta v_2, \Delta v_2) = (\Delta y_1, 0, 0)$$

Then, we can choose $\Delta v_2 = 1$. In this way, we have $\Delta v_1 = -3$ and $\Delta v_3 = 1$, which allows to activate only 3 LUTs.

Then, for $(\Delta u_1, \Delta u_2, \Delta u_3) = M(0, \Delta x_2, \Delta x_3)$, we have

$$\Delta u_1 = \Delta x_2 + \Delta x_3, \Delta u_2 = 2\Delta x_2 + \Delta x_3, \Delta u_3 = \Delta x_2 + 2\Delta x_3.$$

Hence, to possibly match the above output difference $(\Delta v_1, \Delta v_2, \Delta v_3)$, we can also choose a small integer for $(\Delta x_2, \Delta x_3)$. In this way, $(\Delta u_1, \Delta u_2, \Delta u_3)$ also takes small integers, and there is a high probability that we can find such valid difference transition $(\Delta u_1, \Delta u_2, \Delta u_3) \rightarrow (\Delta v_1, \Delta v_2, \Delta v_3)$ through `Bars`. The intuition above suggests that the security of Reinforced Concrete depends critically on the interaction between the `Concrete` layer and the mixed-radix decomposition of the LUT-based S-box.

4 Activating One LUT for LUT-based S-box

In this section, we show how to precisely control and activate a minimal number of LUTs within the LUT-based `Bars` layer. To have the highest differential probability, we focus on the specific scenario in which exactly one LUT is activated.

Let $y = S(x)$, where $x, y \in \mathbb{F}_p$. We denote $\mathcal{D}(x) = (x_1, \ldots, x_m)$ and $\mathcal{D}(y) = (y_1, \ldots, y_m)$, respectively. To activate only the i-th LUT S_i, we require the difference to be localized within the i-th component of the decomposition vector.

Formally, for an input pair (x, x'), let δ_i denote the integer difference at the i-th component:

$$\delta_i = x'_i - x_i, \quad \text{where } \delta_i \in \mathbb{Z}, 0 < |\delta_i| < s_i.$$

A strictly single-active difference pattern implies that

$$\mathcal{D}(x') - \mathcal{D}(x) = (x'_1 - x_1, \ldots, x'_m - x_m) = (0, \ldots, \delta_i, \ldots, 0), \text{ where } \delta_i \neq 0,$$

which equivalently satisfies $\delta_i \neq 0$ and $\delta_j = 0$ for all $j \neq i$.

To capture all such possible differences, we introduce the following definition.

Definition 1 (Single-Active Difference Candidate Set). *Let $y = S(x)$ be the LUT-based S-box over $\mathbb{F}_p$ as defined in Section 2.3, where $x, y \in \mathbb{F}_p$. Let the input and output be represented as $x = \sum_{i=1}^{m} x_i b_i$ and $y = \sum_{i=1}^{m} y_i b_i$, where $y_i = S_i(x_i)$ is a LUT of size s_i. The bases are defined as $b_i = \prod_{j=i+1}^{m} s_j$ for $1 \leq i < m$, and $b_m = 1$. The **Single-Active Difference Candidate Set** $\mathcal{J}$ consists of all candidate input differences for activating exactly one LUT. It is defined as*

$$\mathcal{J} = \bigcup_{i=1}^{m} \mathcal{J}_i, \ \mathcal{J}_i = \{\delta_i \cdot b_i \bmod p \mid \delta_i \in \mathbb{Z}, \ 0 < |\delta_i| < s_i\}.$$

$\mathcal{J}_i$ corresponds to the set of candidate differences for activating only the i-th LUT in $S(x)$.

Removing effects of carry propagation. When constructing algebraic equations, it is natural to work with differences directly in $\mathbb{F}_p$. However, writing $x' = x + \delta_i \cdot b_i$ with $0 < |\delta_i| < s_i$ does not ensure that only S_i is active. Adding $\delta_i \cdot b_i$ may produce carries (or borrows) in the base-b mixed radix representation, thereby modifying multiple components.

Example 1. Let us consider an instance of Reinforced Concrete using p_{BLS381}. The decomposition consists of $m = 27$ small S-boxes, where $s_{27} = 693$ and $s_{26} = 696$. Let $\delta_{27} = 280$ and $\mathcal{D}(x) = (\ldots, 549, 138, 443)$. Then $\mathcal{D}(x') = (\ldots, 549, 139, 30)$. Although we intend to activate only S_{27}, both S_{26} and S_{27} become active due to a carry.

We now quantify the probability that a random pair avoids this carry propagation.

Proposition 1. *For any fixed index i with $1 \leq i \leq m$, let $b_i = \prod_{j=i+1}^{m} s_j$, with $b_m = 1$. Choose $x \in \{0, \ldots, p-1\}$ and $\delta_i \in \{\pm 1, \ldots, \pm(s_i - 1)\}$ independently and uniformly, and set $x' = (x + \delta_i b_i) \bmod p$. The average probability that only S_i is active with input difference δ_i is*

$$P_i = \frac{q_i b_i s_i (s_i - 1) + b_i a_i (a_i - 1) + 2 a_i r_i}{2p(s_i - 1)},$$

where

$$p = q_i s_i b_i + a_i b_i + r_i, \qquad q_i \geq 0, \quad 0 \leq a_i < s_i, \quad 0 \leq r_i < b_i.$$

Proof. Let $\mathcal{D}(x) = (x_1, \ldots, x_m)$ and $\mathcal{D}(x') = (x'_1, \ldots, x'_m)$, where $x' = (x + \delta_i b_i) \bmod p$. Then the difference δ_i affects only the i-th component if and only if

$$0 \leq x_i + \delta_i < s_i \quad \text{and} \quad 0 \leq x + \delta_i b_i < p.$$

In total, there are $2p(s_i - 1)$ possible choices of (x, δ_i). We count pairs that satisfying the above conditions.

Partition $[0, p) \cap \mathbb{Z}$ into q_i complete blocks of length $s_i b_i$ and a final block of length $a_i b_i + r_i$. Within each complete block, the higher components are fixed. For each of the b_i possible lower-component values, both x_i and x'_i can take any of the s_i digit values. Requiring $x'_i \neq x_i$ gives $s_i(s_i - 1)$ choices. Thus, complete blocks contribute $q_i b_i s_i (s_i - 1)$ pairs.

In the final block, let $x = q_i s_i b_i + x_i b_i + t, 0 \leq t < b_i$, where t denotes the lower components. For each $0 \leq t < r_i$, the possible values of x_i are $0, \ldots, a_i$, giving $a_i(a_i + 1)$ ordered choices of distinct digits. For each $r_i \leq t < b_i$, there are a_i possible digit values, giving $a_i(a_i - 1)$ ordered choices. Hence, the final block contributes

$$r_i a_i (a_i + 1) + (b_i - r_i) a_i (a_i - 1) = b_i a_i (a_i - 1) + 2 a_i r_i.$$

Hence, we have

$$P_i = \frac{q_i b_i s_i (s_i - 1) + b_i a_i (a_i - 1) + 2 a_i r_i}{2p(s_i - 1)}, 1 \leq i \leq m.$$

□

Remark 1. Notice that using $p = q_i s_i b_i + a_i b_i + r_i$, we obtain

$$P_i = \frac{1}{2} - \frac{a_i (s_i - a_i) b_i + (s_i - 1 - 2a_i) r_i}{2p(s_i - 1)}.$$

For all three RC parameter sets, the deviation of P_i from $1/2$ is less than 2^{-8}. Hence, we have $P_i \approx 1/2$ for $1 \leq i \leq m$.

5 Collision Attacks on 4.5-round Reinforced Concrete

In this section, we turn the insights of Section 3 into an explicit collision attack for 4.5-round Reinforced Concrete using the framework stated in Section 4. As shown in Figure 6, the 4.5-round Reinforced Concrete is composed of

$$(B) \circ (M_5 \circ \texttt{Bars} \circ M_4) \circ (B \circ M_3) \circ (B \circ M_2) \circ (B \circ M_1).$$

While `Concrete` employs an MDS matrix, we exploit a critical structural feature: the matrix is composed entirely of small integers (1 and 2). This helps us to effectively find high probability difference transitions for `Bars` by activating a minimal number of LUTs.

5.1 Exploiting the Weak Diffusion of `Concrete`

Let $(m_1, m_2, 0)$ and $(m_1 + \Delta m_1, m_2 + \Delta m_2, 0)$ denote the input pair. As shown in Figure 6, Δh denotes the difference of B_3 after the final `Bricks` layer and we have $\Delta h = 0$. We use $\Delta u = (\Delta u_1, \Delta u_2, \Delta u_3)$ and $\Delta v = (\Delta v_1, \Delta v_2, \Delta v_3)$ to denote the input and output difference of `Bars`, respectively. Let $\Delta x = (\Delta x_1, \Delta x_2, \Delta x_3)$ and $\Delta y = (\Delta y_1, \Delta y_2, \Delta y_3)$ denote the input difference and the output difference of the middle layer $M_5 \circ \texttt{Bars} \circ M_4$, respectively. Since $\Delta h = 0$, we have

$$(y_3 + \Delta y_3)((y_2 + \Delta y_2)^2 + \alpha_2(y_2 + \Delta y_2) + \beta_2) - y_3(y_2^2 + \alpha_2 y_2 + \beta_2) = 0.$$

Let $\Delta y_3 = 0$, then we have $\Delta y_2 y_3(2y_2 + \Delta y_2 + \alpha_2) = 0$. To make the difference easier to control, we focus on the case $\Delta y_2 = 0$.

When $\Delta y_2 = \Delta y_3 = 0$, we obtain the following linear system:

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} \Delta v_1 \\ \Delta v_2 \\ \Delta v_3 \end{pmatrix} = \begin{pmatrix} \Delta y_1 \\ 0 \\ 0 \end{pmatrix}$$

After performing Gaussian elimination, we can get

$$\Delta v_3 = \Delta v_2, \ \Delta v_1 = -3\Delta v_2.$$

Our goal is to activate only a single LUT in $S(x)$. Thus, each output difference Δv_i must take values from the set $\mathcal{J} = \bigcup_{j=1}^{m} \mathcal{J}_j$.

Although $\mathcal{J}$ is a discrete set, the structure of `Concrete` significantly increases the number of valid output differences. Specifically, to determine whether a given output difference is valid, it is sufficient to check whether $-3\Delta v_2 \in \mathcal{J}$. Notice that when condition $0 < |-3\delta_j| < s_j$ holds, we have $-3\Delta v_2 \in \mathcal{J}_j$, which implies that the three components $\Delta v_1, \Delta v_2, \Delta v_3$ only activate the j-th LUT when passing through S. Consequently, the total number of potential correct output differences for S_j is $N_{out,j} = 2 \times \lfloor (s_j - 1)/3 \rfloor$, $1 \le j \le m$.

5.2 An Additional Constraint for Δu

The input difference Δu does not require any restrictions other than belonging to the set $\mathcal{J}$. However, the Bricks layer exhibits an unbalanced algebraic degree $(5, 3, 3)$ for the three coordinates. Since there remain unused degrees of freedom in the input pairs, we impose the additional constraint $\Delta x_1 = 0$ on the input difference of M_4 to reduce the overall algebraic degree of the resulting system of equations.

After performing Gaussian elimination, we have

$$\begin{cases} \Delta x_2 + \Delta x_3 = \Delta u_1, \\ 2\Delta x_2 + \Delta x_3 = \Delta u_2, \\ \Delta x_2 + 2\Delta x_3 = \Delta u_3. \end{cases}$$

Let $\Delta u_1, \Delta u_3 \in \mathcal{J}_j$, if $\Delta u_2 = 3\Delta u_1 - \Delta u_3 \in \mathcal{J}_j$, $(\Delta u_1, \Delta u_2, \Delta u_3)$ all only activate the j-th LUT in $S(x)$. The number of potential correct input differences $(\Delta u_1, \Delta u_2, \Delta u_3)$ for S_j is denoted by $N_{in,j}$, $1 \leq j \leq m$.

Given Δu and Δv, when each $\Delta u_i, \Delta v_i$ activate the LUT at the same place in $S(x)$, i.e., the j-th LUT, we can efficiently determine whether it is a valid transition by constructing the Difference Distribution Table (DDT) of each S_j. The DDT of S_j is of size $(2s_j - 1) \times (2s_j - 1)$. We can construct the DDT by going through all the possible $s_j(s_j - 1)$ input pairs that satisfy the single-active condition. The number of valid input/output differences for S_j is denoted by N_j.

5.3 Our Attack Framework

In this subsection, we present our framework for collision attacks of 4.5-round Reinforced Concrete. An overview of the attack framework is illustrated in Figure 6. Specifically, value pairs are depicted in red, while differences are shown in blue. The whole procedure can be divided into the following three stages.

Stage 1: Find the valid $\Delta u \to \Delta v$ for the Bars layer.

1. To satisfy condition $\Delta h = 0$, we require $\Delta y_2 = \Delta y_3 = 0$, which implies:
$$\Delta v_3 = \Delta v_2, \quad \Delta v_1 = -3\Delta v_2. \tag{2}$$
2. We impose an additional constraint $\Delta x_1 = 0$ on the input of M_4, which implies:
$$\Delta u_2 = 3\Delta u_1 - \Delta u_3. \tag{3}$$
3. By searching the set $\mathcal{J}_j$ for candidates that satisfy Equation (2) and Equation (3), $1 \leq j \leq m$, we identify the specific valid values for Δu and Δv associated with S_j. The number of potential correct input differences and output differences for S_j are denoted by $N_{in,j}$ and $N_{out,j}$, respectively.
4. The set of candidate Δu and Δv is then further filtered by checking the DDT of each S_j. The number of valid $\Delta u \to \Delta v$ for S_j is denoted by N_j, and the number of valid input differences after filter is denoted by $N'_{in,j}$, $1 \leq j \leq m$.

Stage 2: Find the differential trail.

1. For each valid Δu, we compute the inverse of the linear layer to obtain the corresponding input difference $(0, \Delta x_2, \Delta x_3)$.
2. Set $m_1, m_2, \Delta m_1, \Delta m_2$ as 4 unknown variables. We further assume that the input difference of B_1 in the 1st round be 0, then we have $2\Delta m_1 + \Delta m_2 = 0$, i.e., $\Delta m_2 = -2\Delta m_1$.
3. For each derived $(0, \Delta x_2, \Delta x_3)$, we construct a system of 3 equations in $(m_1, m_2, \Delta m_1)$ as:

   $$f_3 \circ f_2 \circ f_1(m_1 + \Delta m_1, m_2 - 2\Delta m_1, 0) - f_3 \circ f_2 \circ f_1(m_1, m_2, 0) = (0, \Delta x_2, \Delta x_3),$$

   where $f_i(\cdot) = B \circ M_i(\cdot)$. After solving the equation system, we now have $(m_1, m_2, \Delta m_1)$ for each possible Δx.

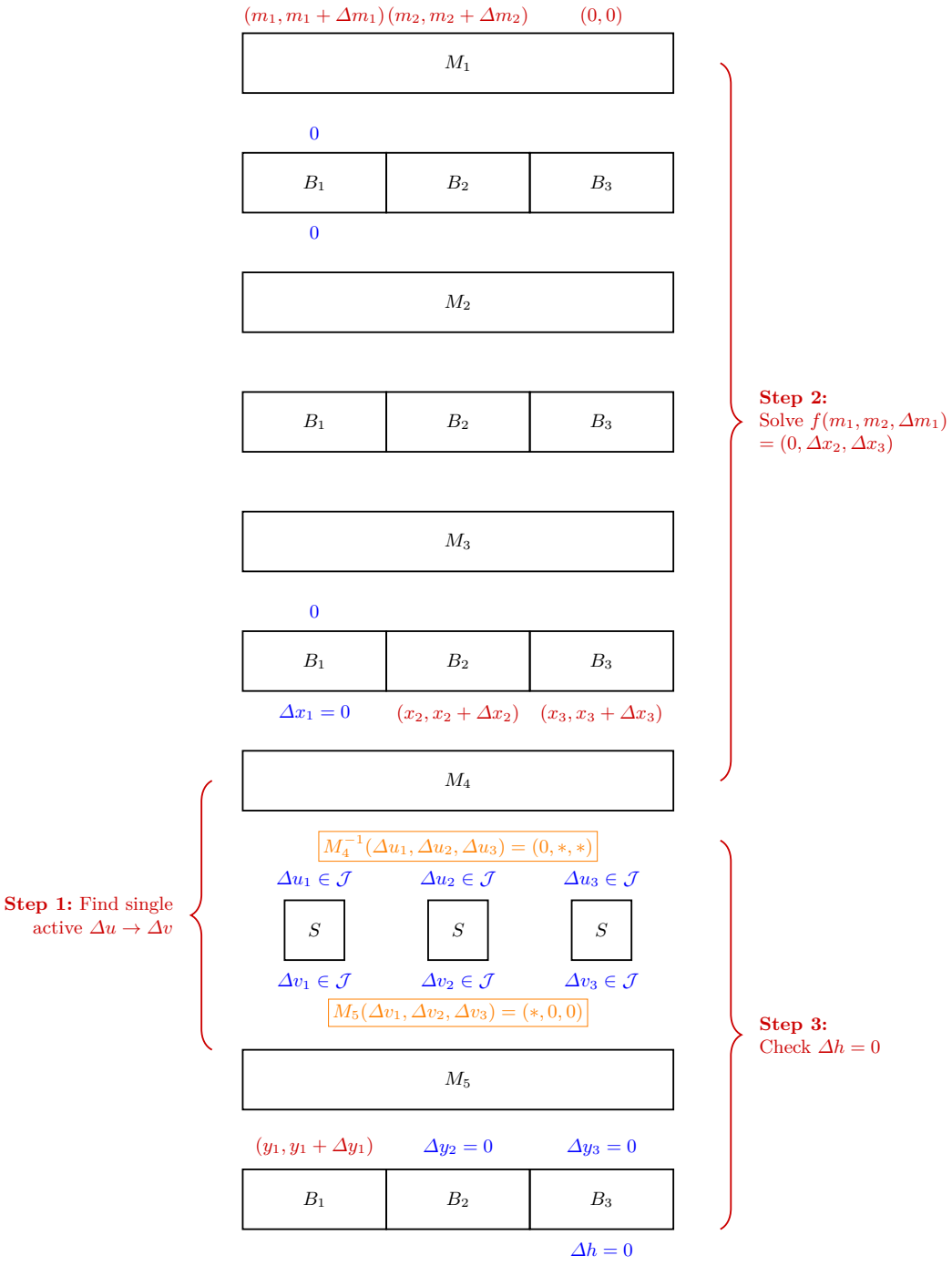

Fig. 6: The framework of 4.5-round collision attacks for Reinforced Concrete.

Stage 3: Find the collision. For each solution derived from the equation system in Stage 2, we obtain specific candidate values for input pairs u and input differences Δu to the Bars layer. As discussed in Proposition 1, a given pair satisfies the required single S-box activation condition with a probability of approximately $(P_i)^3$. Consequently, we first filter the candidate pairs based on this activation condition. For the remaining pairs, we compute the output difference to verify if $\Delta h = 0$ holds.

Complexity and success probability evaluation. In Stage 1, the time complexity is negligible. For the number of valid $\Delta u \to \Delta v$, we obtain $N_{out,j}$ possible output differences and $N_{in,j}$ possible input differences for S_j, $1 \le j \le m$ These candidate sets are subsequently filtered by checking the DDT of S_j, yielding a total of N_j valid $\Delta u \to \Delta v$ with $N'_{in,j}$ valid input differences. Summing over all m S-boxes, the total number of valid $\Delta u \to \Delta v$ is $N = \sum_{j=1}^{m} N_j$, with $N'_{in} = \sum_{j=1}^{m} N'_{in,j}$ valid input differences.

In Stage 2, we are required to solve a total of $N'_{in} = \sum_{j=1}^{m} N'_{in,j}$ equation systems. For p_{BLS381} and p_{BN254}, a trivial upper bound on the degree of the three polynomials is $(d^2, d^2 \times 3, d^2 \times 3) = (25, 75, 75)$ when $d = 5$. However, due to the cancellation, the actual degrees reduce to $(23, 73, 73)$. This reduction is caused by the zero input difference of B_1 in the first round. Specifically, while the output of M_2 has degrees of $(5, 5, 5)$, the output differences of M_2 are only of degree $(3, 3, 3)$. Consequently, after the Bricks layer in the second round, the observed degree drops. Under the regularity assumption, the Macaulay bound is $d_{mac} = 167$ and the Bézout bound is $B = 23 \times 73 \times 73 = 122567$. Therefore, the overall complexity of solving equation systems is

$$\begin{aligned} T &= N'_{in} \times \max\{\tbinom{3+d_{mac}}{3}^{\omega}, 3 \cdot 122567^3\} \\ &\approx N'_{in} \times \max\{2^{19.62\omega}, 2^{52.29}\}. \end{aligned}$$

For p_{ST}, the upper bound of the degree is $(3^2, 3^3, 3^3) = (9, 27, 27)$. We have $d_{mac} = 61$ and the Bézout bound is $B = 6561$. Therefore, the overall complexity of solving equation systems is

$$T_{\mathrm{ST}} \approx N'_{in} \times \max\{2^{15.35\omega}, 2^{39.62}\}.$$

In Stage 3, evaluating the success probability requires determining the average number of solutions u for a valid input/output difference pair $(\Delta u, \Delta v)$. We can use the DDT of S_i to calculate the number of possible solutions for each given input/output difference. Suppose that the probability that a randomly selected valid input pair satisfies $\Delta u \to \Delta v$ is given by Q_j for S_j, the total success probability can be computed as

$$P \approx \sum_{j=1}^{m} N'_{in,j} \times P_j^3 \times \frac{N_j}{N'_{in,j}} \times Q_j \approx \sum_{j=1}^{m} \frac{N_j \times Q_j}{8}.$$

On different instances. In the case of p_{BLS381}, to estimate the number of valid $\Delta u \to \Delta v$, we take S_{27} as an example, since the first p' values are identical across all S-boxes. For S_{27} we obtain $N_{out,27} \approx 2^{8.85}$ and $N_{in,27} \approx 2^{19.28}$. After filtered by checking the DDT of S_{27}, we have a total of $N_{27} \approx 2^{23.31}$ valid $(\Delta u, \Delta v)$ with $N'_{in,27} \approx 2^{19.17}$ and $N'_{out,27} \approx 2^{8.85}$. Summing over all m S-boxes, the total number of valid $\Delta u \to \Delta v$ is $N = \sum_{j=1}^{m} N_j \approx 2^{28.07}$.

In Stage 2, we need to solve a total of $N'_{in} = \sum_{j=1}^{27} N'_{in,j} \approx 2^{23.93}$ equation systems. Therefore, the overall complexity of solving the equation systems is

$$\begin{aligned} T &\approx 2^{23.93} \times \max\{2^{19.62\omega}, 2^{52.29}\} \\ &\approx \max\{2^{19.62\omega+23.93}, 2^{76.23}\} \end{aligned}$$

In Stage 3, we can use the DDT of S_j to calculate the number of possible solutions for each given input/output difference. Take S_{27} as an example, the average number of input pairs for each Δu is $2^{26.93}$. For a specific transition $\Delta u \to \Delta v$, the average number of solutions is approximately $2^{1.26}$. Consequently, the probability that a randomly selected valid input pair satisfies $\Delta u \to \Delta v$ is given by $Q_{27} \approx 2^{-25.67}$. Finally, the total success probability can be computed as

$$P \approx \sum_{j=1}^{27} \frac{N_j \times Q_j}{8} \approx 0.66.$$

Comparisons with the p_{BN254} and p_{ST} instances are summarized in Table 3, where $\bar{Q} = (\sum_{j=1}^{m} Q_j)/(8m)$.

Table 3: Complexity and success probability for different instances of 4.5-round Reinforced Concrete.

Instance	N	N'_{in}	**Degree**	$\bar{Q}$	T	P
p_{BLS381}	$2^{28.07}$	$2^{23.93}$	$(23, 73, 73)$	$2^{-28.68}$	$\max\{2^{19.62\omega+23.93}, 2^{76.23}\}$	0.66
p_{BN254}	$2^{27.93}$	$2^{23.80}$	$(23, 73, 73)$	$2^{-28.54}$	$\max\{2^{19.62\omega+23.80}, 2^{76.10}\}$	0.66
p_{ST}	$2^{29.67}$	$2^{24.98}$	$(9, 27, 27)$	$2^{-30.38}$	$\max\{2^{15.35\omega+24.98}, 2^{64.60}\}$	0.61

Remark 2. While our attack framework targets a single active LUT per S-box, generalizing to multiple active LUTs per S-box might appear as a natural extension. However, this approach triggers an exponential explosion in time complexity. Specifically, for $l \geq 2$ active LUTs per S-box, the average success probability $\bar{Q}$ reduces from around $2^{-3} \cdot p^{-3/m}$ to around $2^{-3l} \cdot p^{-3l/m}$. The required number of candidate differential paths N should increase to around $p^{3l/m} \times 2^{3l}$ to maintain the success rate. Since evaluating each candidate requires solving an equation system, the overall time complexity explodes exponentially.

Enhancing success probability via two input blocks. To further enhance the success probability, we can extend the attack to utilize a multi-block strategy. After the first input block (p_1, p_1') is absorbed, let the resulting internal state be denoted by (e_1, e_2, e_3) and (e_1', e_2', e_3'). The inputs for the second permutation are represented by $(m_1 + e_1, m_2 + e_2, e_3)$ and $(m_1 + \Delta m_1 + e_1', m_2 + \Delta m_2 + e_2', e_3')$. We further assume that the input difference of B_1 in the 1st round to be zero. This implies

$$2(\Delta m_1 + \Delta e_1) + (\Delta m_2 + \Delta e_2) + \Delta e_3 = 0,$$

i.e., $\Delta m_2 = -2\Delta m_1 - (2\Delta e_1 + \Delta e_2 + \Delta e_3)$.

We then repeat the collision search procedure described in the previous section to find the valid $m_1, m_2, \Delta m_1$. By repeating this process up to three times, the success probability can be estimated as $P \approx 1 - (1 - 0.66)^3 \approx 0.96$. The time complexity slightly increases to $T' = 3 \cdot T \approx \max\{2^{19.62\omega + 25.52}, 2^{77.81}\}$ for p_{BLS381}. For p_{ST}, the success probability can be estimated as $P \approx 1 - (1 - 0.61)^3 \approx 0.94$. The time complexity slightly increases to $T' = 3 \cdot T \approx \max\{2^{15.35\omega + 26.56}, 2^{66.19}\}$.

5.4 Experimental Verification

To empirically validate the effectiveness of our framework, we provide a concrete collision pair for the 3.5-round Reinforced Concrete of all the 3 instances, which is composed of

$$(B) \circ (M_4 \circ \texttt{Bars} \circ M_3) \circ (B \circ M_2) \circ (B \circ M_1).$$

All the source code are available at `https://github.com/iljido/Look-up-Based-Cipher`.

For the case of p_{BLS381}, in Stage 2, the constructed algebraic system consists of three polynomials. The degrees of these polynomials are upper-bounded by $(3, 13, 13)$. Consequently, the Macaulay bound is $d_{mac} = 27$, and the Bézout bound is $B = 507$. As N_{in}' remains the same, the overall complexity for solving the equation system is

$$\begin{aligned} T &\approx N_{in}' \cdot \max\{\binom{27+3}{3}^{\omega}, 3 \cdot 507^3\} \\ &\approx N_{in}' \cdot \max\{2^{11.99\omega}, 2^{28.54}\} \\ &\approx \max\{2^{11.99\omega + 23.93}, 2^{52.47}\}. \end{aligned}$$

The colliding input states are specified in hexadecimal format as follows:

$m_1 =$ `0x3a165266fa58dc6c178bf9186f16e5968facb2b444f642eee140b095677964c9`

$m_2 =$ `0x4eaf1522a644995fff69343992223da07b49308bf9e59cf6c452f1b35b9919ca`

$m_1' =$ `0x6a09ba024607789f41eb235e8cbf1a5f15bc2bd2b79d7d500ccaf7629ed98121`

$m_2' =$ `0x62b5ed3f3884de41dde4b7b56073ac14c2e7e252149584336d3e6417ecd8e11b`

The corresponding differences of the input of `Bars` are given as follow, which precisely activates only the 19-th LUT:

$\Delta u_1 =$ `0x73eda753299d7d483339d80809a1d80553bda402fffd90b21a947b519b461141`

$\Delta u_2 =$ `0x73eda753299d7d483339d80809a1d80553bda402fff4943103fc3fa785cab981`

$\Delta u_3 =$ `0x765e74bc1324f4c077a40`

6 Searching for Secure `Concrete` Layer

The collision attack presented in the previous section demonstrates that the small integers of the specific `Concrete` layer matrix in Reinforced Concrete allow the attacker to restrict the difference to a single LUT. This vulnerability significantly reduces the complexity required to mount the attack. While replacing it with a dense, random matrix would effectively render such attacks infeasible, it would introduce prohibitive overhead in zero-knowledge proof systems. Therefore, it is natural to investigate countermeasures that can thwart this attack while preserving the efficiency of Reinforced Concrete in zero-knowledge proof systems.

6.1 Constraints on the Output Difference Δv

Notice that the additional constraint imposed on the input difference Δu (i.e., $\Delta x_1 = 0$) serves primarily to reduce the algebraic degree of the resulting Gröbner basis system. Removing this constraint would merely increase the time complexity of Stage 2, but it simultaneously implies that all input differences $\Delta u_i \in \mathcal{J}$ for $i = 1, 2, 3$ become feasible candidates.

If we relax this condition, the algebraic degrees of the three equations increase to $(5^3, 3 \cdot 5^2, 3 \cdot 5^2) = (125, 75, 75)$ when $d = 5$. Under the assumption of regularity, the Macaulay bound becomes $d'_{mac} = 273$, and the Bézout bound reaches $B' = 125 \times 75^2 \approx 2^{19.42}$. Consequently, the time complexity for solving a single equation system is estimated as:

$$T_{GB} = \max\left\{ \binom{3 + d'_{mac}}{3}^{\omega},\ 3 \cdot (B')^3 \right\} \approx \max\left\{ 2^{21.72\omega},\ 2^{59.84} \right\}.$$

If we take a closer look at the DDT of each LUT S_i, we can reveal that while each valid non-zero output difference corresponds to at least one input difference, there exist exactly two specific output differences, $\pm(s_i - 1)$, for which the corresponding input difference is unique. Furthermore, for these specific differences, the input/output pair satisfying the difference transition is also unique.

Let us consider a specific `Concrete` layer that allows for a valid output difference where each component of the output difference Δv takes one of these special values, i.e. $\Delta v_j \in \{\pm(s_i - 1)b_i\}$, while maintaining $\Delta h = 0$. Under this assumption, the corresponding input difference Δu is uniquely determined. We can then proceed to construct and solve the equation system as described in Stage 2. Since the input/output pair satisfying the difference is unique, the success probability for a random trial can be estimated as $(1/\max(s_i))^3 \approx 2^{-28.37}$.

To enhance the success probability, we employ the multi-block strategy. By treating the second message block as variables, we can perform l independent

trials. To ensure the success probability exceeds 90%, we require

$$(1 - 2^{-28.37})^l \leq 0.1.$$

Solving for l, the required data is approximately $l \approx 2^{29.57} \ll 2^{128}$. Consequently, the total time complexity for the attack is estimated as:

$$T_{total} = l \cdot T_{GB} \approx 2^{29.57} \cdot \max\left\{2^{21.72\omega},\ 2^{59.84}\right\} \approx \max\{2^{21.72\omega+29.57},\ 2^{89.41}\}.$$

This result indicates that the primitive remains vulnerable to collision attacks. Thus, the key point for preventing the attack is to select the `Concrete` layer such that no valid output difference Δv can satisfy the linear constraints.

6.2 Finding a Secure `Concrete` Layer

We consider the circulant matrix $M_c = \mathrm{circ}(c, 1, 1)$, where $c \in \mathbb{F}_p \setminus \{0, 1, -1, -2\}$. Then M_c is MDS and invertible with $\det(M_c) = (c-1)^2(c+2) \neq 0$. Specifically, it does not increase the number of arithmetic constraints required in zero-knowledge proof systems compared to the original design.

To satisfy the collision condition $\Delta h = 0$, we impose the constraint $\Delta y_2 = \Delta y_3 = 0$. The resulting linear system $M_c \cdot \Delta v = (\Delta y_1, 0, 0)^T$ implies:

$$\begin{cases} \Delta v_1 + c\Delta v_2 + \Delta v_3 = 0, \\ \Delta v_1 + \Delta v_2 + c\Delta v_3 = 0. \end{cases}$$

Subtracting these equations yields the following relationship:

$$\Delta v_3 = \Delta v_2, \quad \Delta v_1 = -(c+1)\Delta v_2. \tag{4}$$

The attack becomes feasible only if there exists a tuple $(\Delta v_1, \Delta v_2, \Delta v_3)$ such that all three components belong to the set of single-LUT activations $\mathcal{J}$. From Equation (4), selecting $\Delta v_2 \in \mathcal{J}$ automatically implies $\Delta v_3 \in \mathcal{J}$. Consequently, the security of the design depends on whether $\Delta v_1 = -(c+1)\Delta v_2$, also falls within $\mathcal{J}$.

To thwart the attack, the parameter c must be selected such that $\lambda = -(c+1)$ disrupts the sparse structure of the mixed-radix representation. Formally, we require

$$\forall \Delta v_2 \in \mathcal{J}, \quad -(c+1) \cdot \Delta v_2 \notin \mathcal{J}.$$

Stated in terms of the basis elements, there must not exist a pair of indices (i, j) and integers (k_i, k_j) satisfying $0 < |k_i| < s_i$ and $0 < |k_j| < s_j$, such that:

$$-(c+1) \cdot (k_i b_i) \equiv k_j b_j \pmod{p}. \tag{5}$$

We observe that the set of single-LUT differences, $\mathcal{J}$, has a relatively small cardinality. For p_{BLS381}, $|\mathcal{J}| = \sum_{j=1}^{m} 2 \times (s_j - 1) \approx 2^{15.19}$. The generalized `Concrete` layer allows a valid differential transition if and only if the parameter c satisfies the linear relation $\delta_{out} \equiv -(c+1) \cdot \delta_{in} \pmod{p}$ for some pair $(\delta_{in}, \delta_{out}) \in$

$\mathcal{J} \times \mathcal{J}$. By inverting this relationship, we define the set of vulnerable parameters, denoted by $\mathcal{C}_{bad}$, as:

$$\mathcal{C}_{bad} = \left\{ -\left(\delta_{out} \cdot \delta_{in}^{-1} + 1\right) \bmod p \mid \delta_{in}, \delta_{out} \in \mathcal{J} \right\} \cup \{0, 1, -1, -2\}.$$

The time complexity to construct $\mathcal{C}_{bad}$ is quadratic in the size of $\mathcal{J}$, specifically $O(|\mathcal{J}|^2)$. Given the parameters of Reinforced Concrete, the maximum size of this set is approximately $|\mathcal{J}|^2 \approx 2^{30.3}$. Comparing this to the vast size of the field p, the fraction of insecure parameters is negligible:

$$\frac{|\mathcal{C}_{bad}|}{|\mathbb{F}_p|} \leq \frac{2^{30.38}}{p_{\text{BLS381}}} \approx 2^{-224.47}.$$

This formulation allows for a rigorous selection procedure: by computing $\mathcal{C}_{bad}$ and selecting any $c \in \mathbb{F}_p \setminus \mathcal{C}_{bad}$, we can mathematically guarantee that no valid single-LUT differential characteristic exists for the 4.5-round attack.

7 Security Evaluation of Tip5 by Activating LUTs

The attack of Section 5 establishes that the security of a LUT-based ZK-friendly primitive depends critically on whether its linear layer preserves the mixed-radix sparsity of single-active differences. To evaluate the security level provided by the Tip5 matrix, we focus on semi-free-start (SFS) collision attacks on the 4-round Tip5, activating the minimal number of LUTs to maximize available degrees of freedom.

7.1 A SFS Collision Attack Framework for 4-round Tip5

We present a generalized SFS collision attack framework, which is illustrated in Figure 7. In the following, we give a detailed description of the two steps of the framework.

Step 1: Active LUTs in the 2nd non-linear layer. Let s_1 S-boxes and $t_1 \leq t$ T-boxes be active in the 2nd round. To ensure that there is no difference in the inner part, we can construct a system of c linear equations in $s_1 + t_1$ variables:

$$(M^{-1} \times \Delta x)_i = 0, \text{ for } n - c < i \leq n, \tag{6}$$

where $\Delta x = (\Delta x_1, \cdots, \Delta x_n)$. To make sure that Equation (6) has at least one non-trivial solution, the following equation should hold:

$$s_1 + t_1 > c.$$

By solving Equation (6), we determine the valid input differences Δx.

We then randomly choose the input values for the t_1 active T-boxes, which fixes their output values and output differences. Simultaneously, we randomly select the output differences for the s_1 active S-boxes. In this way, the total output difference of the 2nd non-linear layer is fully determined.

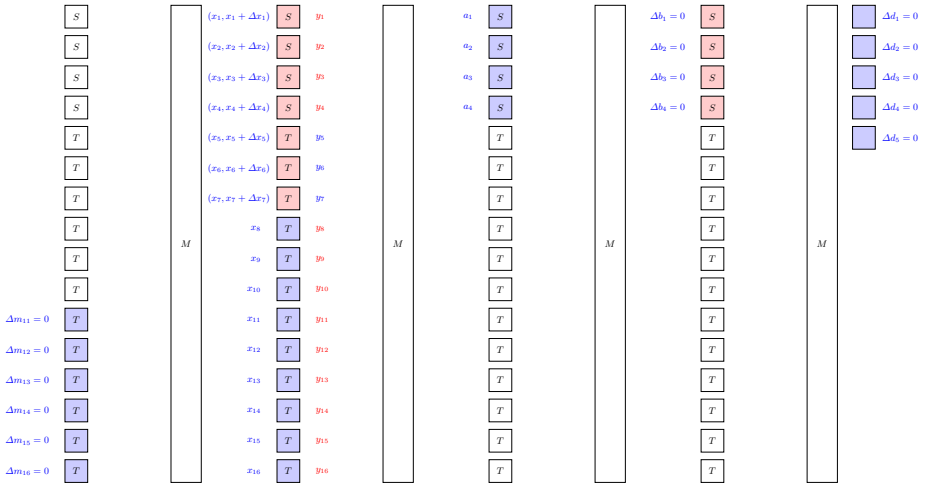

Fig. 7: Attack framework of 4-round Tip5.

Step 2: Establish equation system to find the collision. Denote the output of the s_1 active S-boxes and $t-t_1$ inactive T-boxes as s_1+t-t_1 unknown variables $(y_1, y_2, y_3, \cdots, y_{s_1+t-t_1})$. We can get the $2s+d$ equations as follows:

- s linear equations to ensure that the input value of the S-boxes in the 3rd round is a random chosen constant (a_1, a_2, a_3, a_4):

$$\big(M(y+\Delta y)+c^{(2)}\big)_i = a_i,\ 1 \le i \le s.$$

  Since we already known the full Δy, we can easily get the input/output differences of the s S-boxes in the 3rd round.
- s degree-6 equations to ensure that the input difference of S-boxes in 4th round is 0:

$$\Big(M\Big(S\big(M(y+\Delta y)+c^{(2)}\big) - S\big(M(y)+c^{(2)}\big)\Big)\Big)_i = 0,\ 1 \le i \le s.$$

- d degree-48 equations to ensure that the difference of the hash value is 0. For $1 \le i \le d$, we have

$$\left(M_f\left(S\Big(M\Big(S\big(M(y+\Delta y)+c^{(2)}\big)\Big)+c^{(3)}\Big) - S\Big(M\Big(S\big(M(y)+c^{(2)}\big)\Big)+c^{(3)}\Big)\right)\right)_i = 0.$$

By performing Gaussian elimination on the s linear equations, we only need to solve s degree-6 equations and d degree-48 equations in $s_1 - s + t - t_1$ variables.

Complexity evaluation. This attack framework only works when

$$s_1 - s + t - t_1 \ge s + d$$

For Tip5, we have $s_1 = 4, t_1 = 3$. Under the regularity assumption, the Macaulay bound is given by $d_{mac} = 1 + 5 \times 4 + 47 \times 5 = 256$. The Bézout bound is given by $6^4 \cdot 48^5 \approx 2^{38.26}$. Then the time complexity to solve each equation system can be estimated as

$$T_{tip5} = \max\{\binom{9+256}{9}^{\omega}, 9 \cdot 2^{3\times 38.26}\} \approx \max\{2^{53.78\omega}, 2^{117.94}\}.$$

Once we obtain the solution, the output of the 2nd S-box layer is fully determined. Let w_i denote the number of LUTs with non-zero input differences corresponding to the i-th S-box. For a valid collision to occur, we need to check the validity of the LUT. For a random output of the 8-bit LUT, each solution is valid with probability 2^{-7}. Hence, the probability of satisfying the full output difference of the 2nd S-box layer is around $2^{-7\times W}$, where $W = \min(\sum_{i=1}^{4} w_i)$ represents the total number of active LUTs. Consequently, the efficiency of the attack is maximized when W is minimized.

7.2 Search Strategy for Minimal Active LUTs

To evaluate the security bound rigorously, we aim to identify the minimum number of active LUTs, denoted $W = \min(\sum_{i=1}^{4} w_i)$, required by the linear constraints derived in Step 1.

After Gaussian elimination, the system is reduced to the following form:

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & -a_7 \\ 0 & 1 & 0 & 0 & 0 & 0 & -a_6 \\ 0 & 0 & 1 & 0 & 0 & 0 & -a_5 \\ 0 & 0 & 0 & 1 & 0 & 0 & -a_4 \\ 0 & 0 & 0 & 0 & 1 & 0 & -a_3 \\ 0 & 0 & 0 & 0 & 0 & 1 & -a_2 \end{array}\right) \begin{pmatrix} \Delta x_7 \\ \Delta x_6 \\ \Delta x_5 \\ \Delta x_4 \\ \Delta x_3 \\ \Delta x_2 \\ \Delta x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Specifically, all differences Δx_i can be expressed as scalar multiples of a single free variable as:

$$\Delta x_i - a_i \Delta x_1 = 0, 2 \leq i \leq s_1 + t_1.$$

To perform a comprehensive search, we employ a heuristic strategy inspired by [8]. We iteratively set each variable Δx_k, $1 \leq k \leq s$ as the free variable by applying column permutations to the coefficient matrix before Gaussian elimination. This transformation allows us to express the system for each $k \in \{1, \cdots, s\}$ as:

$$\Delta x_i - a_{i,k} \Delta x_k = 0, \quad 1 \leq i \leq s_1 + t_1, \quad i \neq k.$$

For each chosen base Δx_k, we enumerate all values with a restricted number of active bytes and compute the resulting number of total active LUTs. Let w_{thresh} denote the threshold of the number of the active 8-bit LUTs for the free variable x_k. We enumerate all values of Δx_k such that the number of active bytes

satisfies $w_k \leq w_{thresh}$ and find the local minimum number of total active LUTs, denoted as W_{thresh}. The size of this search space is given by:

$$|\Omega| = s \times \binom{t}{t_1} \times \sum_{j=1}^{w_{thresh}} \binom{8}{j} (2^8 - 1)^j.$$

Based on the search results described above, we can establish a lower bound for W. If the minimum total weight W_{thresh} obtained within the search threshold w_{thresh} satisfies the condition:

$$W_{thresh} > s_1 \times w_{thresh},$$

the lower bound of the number of active 8-bit LUTs is $s_1 \times (w_{thresh} + 1)$.

Application to Tip5. We conducted a heuristic search for Tip5 with parameters $s_1 = 4, t = 12$, and $t_1 = 3$. The search threshold was set to $w_{thresh} = 2$, The total search space size is approximately $|\Omega| \approx 2^{30.6}$. The experimental search yielded a minimum total weight of:

$$W_2 = 21 > 8.$$

So the theoretical lower bound is estimated as $s_1 \times (w_{thresh} + 1) = 12$. The lower bound of time complexity for our attack is

$$\max\{2^{53.78 \times 2}, 2^{117.94}\} \times 2^{7 \times 12} \approx 2^{201.94} \gg 2^{160}.$$

This by far exceeds the generic security bound for collision resistance. Consequently, we conclude that 4-round Tip5 is secure under our SFS collision attack framework. The strong diffusion of the linear layer effectively provides a robust security margin.

8 Conclusion

The results on Reinforced Concrete and Tip5 provide new insight into the design of LUT-based ZK-friendly hash functions. While the idea of employing high-degree LUT-based S-boxes (such as the `Bars` layer) was introduced to provide resistance against algebraic attacks, their accompanying linear layers were often optimized primarily for performance. The core of our attack relies on exploiting the structural vulnerability arising from the weak combination of the `Bars` layer and the `Concrete` layers. Specifically, the small integers of `Concrete` layer did not properly mix the internal components, allowing the differences to propagate in a way that preserves their sparsity. By decomposing the large field element into smaller chunks, an attacker can attempt to bypass the LUT-based S-box surrounded by MDS matrices with high probability. Although this structural weakness does not compromise the security of the full-round Reinforced Concrete, it reduces the security margin, directly facilitating our collision attack on the

4.5-round variant. In Tip5, the MDS matrix forced a dense activation. Although the Bars structure is similar to Reinforced Concrete, the strong linear diffusion of the MDS matrix effectively prevents this attack. We hope these findings shed new light on the role of linear diffusion in the future design of ZK-friendly hash functions.

Acknowledgements. We thank the anonymous reviewers for their valuable comments and constructive suggestions.

This work is supported by the National Key R&D Program of China (Grant No. 2024YFA1013000), the National Natural Science Foundation of China (Grant No. U2336207), the National Cryptologic Science Fund of China (Grant No. 2025NCSF01013), the Key R&D Program of Shandong Province, China (Grant No. 2026CXPT037), and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM114). Fukang Liu has been supported by JSPS KAKENHI Grant Number 26K14705.

References

1. M. R. Albrecht, C. Cid, L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, and M. Schofnegger. Algebraic cryptanalysis of stark-friendly designs: Application to marvellous and mimc. In *ASIACRYPT 2019*, volume 11923 of *Lecture Notes in Computer Science*, pages 371–397. Springer, 2019.
2. M. R. Albrecht, L. Grassi, L. Perrin, S. Ramacher, C. Rechberger, D. Rotaru, A. Roy, and M. Schofnegger. Feistel structures for mpc, and more. In *ESORICS 2019*, volume 11736 of *Lecture Notes in Computer Science*, pages 151–171. Springer, 2019.
3. M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, 2016.
4. A. Aly, T. Ashur, E. Ben-Sasson, S. Dhooghe, and A. Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symmetric Cryptol.*, 2020(3):1–45, 2020.
5. T. Ashur, A. Kindi, W. Meier, A. Szepieniec, and B. Threadbare. Rescue-prime optimized. *IACR Cryptol. ePrint Arch.*, page 1577, 2022.
6. A. Bak, G. Jazeron, P. Galissant, and L. Perrin. Attacking split-and-lookup-based primitives using probabilistic polynomial system solving applications to round-reduced monolith and full-round skyscraper. *IACR Trans. Symmetric Cryptol.*, 2025(3):337–367, 2025.
7. A. Bak and L. Perrin. On the security of split-and-lookup-based zk-friendly primitives. *IACR Trans. Symmetric Cryptol.*, 2025(2):87–123, 2025.
8. A. Bar-On, I. Dinur, O. Dunkelman, V. Lallemand, N. Keller, and B. Tsaban. Cryptanalysis of SP networks with partial non-linear layers. In *EUROCRYPT 2015*, LNCS, pages 315–342. Springer, 2015.
9. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. On the indifferentiability of the sponge construction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 181–197. Springer, 2008.

10. L. Bettale, J. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *J. Math. Cryptol.*, 3(3):177–197, 2009.
11. C. Bouvier, P. Briaud, P. Chaidos, L. Perrin, R. Salen, V. Velichkov, and D. Willems. New design techniques for efficient arithmetization-oriented hash functions: ttanemoi permutations and ttjive compression mode. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539. Springer, 2023.
12. C. Bouvier, L. Grassi, D. Khovratovich, K. Koschatko, C. Rechberger, F. Schmid, and M. Schofnegger. Skyscraper: Fast hashing on big primes. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2025(2):743–780, 2025.
13. C. Bouvier, L. Grassi, D. Khovratovich, K. Koschatko, C. Rechberger, F. Schmid, and M. Schofnegger. Skyscraper-v2: Fast hashing on big primes. Cryptology ePrint Archive, Paper 2025/058, 2025.
14. D. G. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, pages 587–592, 1981.
15. J. Faugère, P. M. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *J. Symb. Comput.*, 16(4):329–344, 1993.
16. J. Faugère and C. Mou. Sparse FGLM algorithms. *J. Symb. Comput.*, 80:538–569, 2017.
17. J.-C. Faugère, M. Bardet, and B. Salvy. On the complexity of gröbner basis computation of semi-regular overdetermined algebraic equations. 2004.
18. A. Gabizon and Z. J. Williamson. plookup: A simplified polynomial protocol for lookup tables. *IACR Cryptol. ePrint Arch.*, page 315, 2020.
19. L. Grassi, Y. Hao, C. Rechberger, M. Schofnegger, R. Walch, and Q. Wang. Horst meets fluid-spn: Griffin for zero-knowledge applications. In H. Handschuh and A. Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 573–606. Springer, 2023.
20. L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch. Reinforced concrete: A fast hash function for verifiable computation. In *CCS 2022*, pages 1323–1335. ACM, 2022.
21. L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch. Monolith: Circuit-friendly hash functions with new nonlinear layers for fast and constant-time implementations. *IACR Trans. Symmetric Cryptol.*, 2024(3):44–83, 2024.
22. L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In M. D. Bailey and R. Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 519–535. USENIX Association, 2021.
23. L. Grassi, D. Khovratovich, and M. Schofnegger. Poseidon2: A faster version of the poseidon hash function. In N. E. Mrabet, L. D. Feo, and S. Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings*, volume 14064 of *Lecture Notes in Computer Science*, pages 177–203. Springer, 2023.
24. L. Grassi, S. Onofri, M. Pedicini, and L. Sozzi. Invertible quadratic non-linear layers for mpc-/fhe-/zk-friendly schemes over fnp application to poseidon. *IACR Trans. Symmetric Cryptol.*, 2022(3):20–72, 2022.

25. F. Liu, K. Koschatko, L. Grassi, H. Yan, S. Chen, S. Banik, and W. Meier. Opening the blackbox: Collision attacks on round-reduced tip5, tip4, tip4' and monolith. *IACR Trans. Symmetric Cryptol.*, 2024(4):97–137, 2024.
26. NIST and M. J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 00:08:00 2015.
27. S. Papini and U. Haböck. Improving logarithmic derivative lookups using GKR. *IACR Cryptol. ePrint Arch.*, page 1284, 2023.
28. L. Pearson, J. B. Fitzgerald, H. Masip, M. Bellés-Muñoz, and J. L. Muñoz-Tapia. Plonkup: Reconciling plonk with plookup. *IACR Cryptol. ePrint Arch.*, page 86, 2022.
29. R. Salen. Two additional instantiations from the tip5 hash function construction, 2023.
30. S. T. V. Setty, J. Thaler, and R. S. Wahby. Unlocking the lookup singularity with lasso. In *EUROCRYPT 2024*, volume 14656 of *LNCS*, pages 180–209. Springer, 2024.
31. A. Szepieniec, T. Ashur, and S. Dhooghe. Rescue-prime: a standard specification (sok). *IACR Cryptol. ePrint Arch.*, page 1143, 2020.
32. A. Szepieniec, A. Lemmens, J. F. Sauer, and B. Threadbare. The tip5 hash function for recursive starks. *IACR Cryptol. ePrint Arch.*, page 107, 2023.