Quantum Periodic Distinguisher Construction: Symbolization Method and Automated Tool

Qun Liu[1,4,5], Haoyang Wang[2], Jinliang Wang[1,4,5], Boyun Li[1,4,5], and Meiqin Wang[1,3,4,5]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, China
{qunliu,jinliangwang,boyunli}@mail.sdu.edu.cn
[2] Shanghai Jiao Tong University, Shanghai, China
haoyang.wang@sjtu.edu.cn
[3] Quan Cheng Shandong Laboratory, Jinan, China
mqwang@sdu.edu.cn
[4] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[5] State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Qingdao, 266237, China

Abstract. As one of the famous quantum algorithms, Simon's algorithm enables the efficient derivation of the period of periodic functions in polynomial time. However, the complexity of constructing periodic functions has hindered the widespread application of Simon's algorithm in symmetric-key cryptanalysis. Currently, aside from the exhaustive search-based testing method introduced by Canale et al. at CRYPTO 2022, there is no unified model for effectively searching for periodic distinguishers. Although Xiang et al. established a link between periodic function and truncated differential theory at ToSC 2024, their approach lacks the ability to construct periods using unknown differentials and does not provide automated tools. This limitation underscores the inadequacy of existing methods in identifying periodic distinguishers for complex structures. In this paper, we address the challenge of advancing periodic distinguishers for symmetric-key ciphers. First, we propose a more generalized theory for constructing periodic distinguishers, addressing the limitations of Xiang et al.'s theory in handling unknown differences. We further extend our theory to probabilistic periodic distinguishers, thereby extending the separability property proposed by Hodžić et al. in 2020. As a result, our theory can cover a wider range of periodic distinguishers. Second, we introduce a novel symbolic representation to simplify the search of periodic distinguishers. Based upon this representation, we propose the first fully automated SMT-based search model, which efficiently addresses the challenges of manual searching in complex structures. Finally, we extend the model to SPN structures based on our new theory. Our model has broad applicability through significant advancements in analyzing generalized Feistel structures (GFSs) and SPN-based ciphers. As a general model, we have achieved new quantum distinguishers with the following round configurations: 10 rounds for GFS-4F, 10 rounds for LBlock, 10 rounds for TWINE, and 16 rounds for Skipjack-B, improving the previous best results by 2, 2, 2, and 3 rounds,

respectively. In the domain of SPN-based ciphers, our model has enabled the identification of novel periodic distinguishers, including the first 9-round distinguisher for SKINNY and the first 12-round distinguisher for CRAFT. These achievements lay the foundation for quantum cryptanalysis of SPN-based ciphers using Simon's algorithm.

Keywords: Quantum cryptanalysis · Automated search model · Simon's algorithm · Generalized Feistel structure · SPN structure

1 Introduction

Quantum computing represents a significant paradigm shift in computation, with the potential to solve specific classes of problems with exponential speedup compared to classical computing systems. Among its most significant implications is its impact on cryptography, where quantum algorithms challenge the foundation of classical cryptographic security.

Grover's algorithm [12], which provides a quadratic speedup for unstructured search problems, has played an important role in evaluating the quantum threat to symmetric-key ciphers. The theoretical perspective suggests that the primary quantum vulnerability for symmetric-key ciphers arises from Grover's algorithm, necessitating a doubling of key lengths to maintain security levels equivalent to those in classical systems. However, this strategy may not fully address the broader spectrum of quantum attacks. Simon's algorithm [26], on the other hand, poses a more substantial challenge by reducing the complexity of the exponential-time periodicity search algorithm in classical computing to polynomial time in the quantum setting. Despite its theoretical importance, Simon's algorithm has not been extensively applied in the cryptanalysis of primitives. This gap is largely attributable to the incomplete theoretical framework and the lack of robust methodologies for constructing periodic functions of symmetric-key ciphers based on Simon's approach.

Addressing this gap is essential for advancing the theoretical foundations of symmetric cryptography. In recent years, the cryptographic community has directed significant efforts toward exploring the construction of periodic functions for various cryptographic primitives based on Simon's algorithm. A seminal contribution was made at CRYPTO 2016 by Kaplan et al. [18], who demonstrated for the first time that widely deployed modes of operation for authentication and authenticated encryption, such as CBC-MAC, GCM, and OCB etc., are completely compromised utilizing Simon's algorithm. Subsequently, at ASIACRYPT 2017, Leander and May [21] made a pioneering advance by proposing the Grover-meet-Simon algorithm. They showed that the use of whitening keys, such as FX construction, does not enhance quantum security. Another progress was achieved at ASIACRYPT 2019 by Bonnetain et al. [5], who refined the Grover-meet-Simon algorithm to achieve an optimized trade-off between quantum time complexity and classical data requirements for EM and FX constructions. At

ASIACRYPT 2021, Bonnetain et al. [6] introduced a novel methodology for applying Simon's algorithm, termed quantum linearization attacks. This approach successfully compromised numerous parallelizable MACs that were previously considered secure under classical beyond-birthday-bound security assumptions. The core principle underlying these attacks is that the successful construction of a periodic function enables the execution of an attack on the cryptographic structure with significantly reduced computational complexity.

The central challenge in quantum cryptanalysis with Simon's algorithm lies in the construction of periodic functions for complex structures. For the Feistel structure with two branches, it is relatively easy to construct the periodic function. Kuwakado and Morii [20] proposed a 3-round periodic distinguisher, while Ito et al. [16] introduced a 4-round periodic distinguisher for the Feistel structure using the quantum chosen-ciphertext attack. In contrast, constructing periodic functions for more intricate designs, such as the Generalized Feistel Structure (GFS) with increased number of branches, presents significant challenges. While several heuristic manual deduction methods [11,10,8,27] have been used to identify periodic functions of various GFS instances, these approaches are inherently limited. Such manual methods become increasingly impractical for identifying periodic functions as the number of branches in the structure grows. Furthermore, these techniques are unsuitable for analyzing newly designed structures.

There is an urgent need for a generalized and automated framework to efficiently evaluate the security of various cryptographic structures against quantum attacks with Simon's algorithm. Currently, three main approaches have been explored in the literature. First, in 2020, Hodžić et al. [14] attempted to summarize the general properties of periodic function construction. However, their work was not able to encompass many effective periodic functions. Second, at CRYPTO 2022, Canale et al. [7] proposed the first automated algorithm to identify periodic functions. Their method involved examining all possible circuits and testing each one for periodicity by instantiating the function in small dimensions. Although this approach is effective for structures with few branches, it suffers from scalability issues as the number of branches increases, making it impractical for more complex designs. Third, in order to simplify the construction of periodic distinguishers, Xiang et al. [30] established the links between periodic functions and truncated differentials at ToSC 2024. Although their framework can be used to identify periodic distinguishers for various GFSs, it lacks the ability for constructing periods using unknown differentials and does not provide automated tools, limiting its practical utility.

Substitution-Permutation Network (SPN) is one fundamental structure for block ciphers, prominently exemplified by AES [9] and SKINNY [2]. SPN structures exhibit stronger avalanche properties compared to Feistel networks. However, it is more difficult to identify effective periodic distinguishers for SPN-based ciphers. Consequently, existing methods for searching effective periodic distinguishers are difficult to be applied to SPN structures. For example, for SKINNY block cipher with 4×4 branch configuration, the fact that all branches pass through the S-box in each round leads to unknown differentials to appear very

quickly. This renders the truncated differential theory [30] proposed by Xiang et al. inapplicable, as it relies on predictable differential propagation patterns. Additionally, the use of Maximum Distance Separable (MDS) matrices in SPN structures further complicates the search for periodic distinguishers by introducing strong diffusion properties that disrupt traditional analytical approaches. Furthermore, the automated method proposed by Canale et al. [7], which involves examining all input circuits, instantiating them, and testing each for periodicity, becomes impractical for SPN structures due to their increased complexity. This limitation underscores the inadequacy of existing methods in addressing the unique challenges of searching periodic distinguishers for SPN-based ciphers.

1.1 Our Contributions

In this paper, we address the challenges of constructing periodic distinguishers for symmetric-key primitives. We propose a new generalized theoretical framework for building periodic distinguishers and, based on this framework, present the first efficient automated search model tailored for Generalized Feistel Structures (GFSs). Additionally, we extend this model to Substitution-Permutation Network (SPN)-based block ciphers. Our key contributions are summarized as follows:

More generalized theory for constructing periodic distinguishers. Our constructed theory consists of two parts, the polynomial-time periodic distinguisher (see Theorem 3) and the probabilistic periodic distinguisher (see Theorem 4). The polynomial-time periodic distinguisher not only accounts for the input difference value of x and α_b (α_0 and α_1 are two distinct constants) for the input branches, but also incorporates the values and difference value after the round functions. This advancement addresses the limitation of information loss inherent in Xiang et al.'s truncated differential theory [30], enabling the construction of periodic functions even in the presence of certain unknown differences. As a direct consequence, we achieve a one-round improvement in the distinguisher for GFS-2F [24]. Furthermore, we extend the theory to probabilistic periodic distinguishers based on the collisions of α_b. While previous approaches typically involved an arbitrary selection of α_b, our work demonstrates that when α_b satisfies certain properties (see Definition 3), it becomes possible to identify the potential new periodic function. This extends the separability property in [14]. The periodicity occurs only for particular choices of α_0 and α_1, which we can search for using Grover's algorithm. This turns the distinguisher into a Grover-meet-Simon algorithm [21,5].

First fully automated SMT-based search model for periodic distinguishers. Based on our generalized theory, we implement the first fully automated SMT-based search model for periodic distinguishers, which significantly differs from the circuit instantiations method in [7]. Our model utilizes simplified symbols representations to identify the differences of x, α_b, and the output

values of the round functions, while ensuring that the tail satisfies the periodicity. As a broad application of our model, we discover the first quantum periodic distinguishers with the following round configurations: 10 rounds for GFS-4F [24], 16-round Skipjack-B [19,4,8], 10 rounds for LBlock [29], and 10 rounds for TWINE [28], improving the previous best results by 2, 3, 2 and 2 rounds, respectively. As the first example, the 16-round periodic distinguisher we present for Skipjack-B outperforms existing classical distinguishers in the classical setting. Based on the birthday problem, we prepare $2^{\frac{16}{2}}$ values of x. For each x, we encrypt x for α_0 and α_1. We store the corresponding outputs in order to find a pair of x that satisfies the possible period. The final complexity is about 2^9. However, the classic impossible differential distinguisher requires the complexity at least 2^{32} [4].

Table 1: The periodic distinguishers of different structures. qCPA is the quantum chosen-plaintext attack. qCCA is the quantum chosen-ciphertext attack. N is the size of the internal state of the structure.

Structure	Type	Attack	#Rounds	Complexity	Reference
GFS-2F	GFS	qCPA	5	$O(N)$	[31]
			6	$O(2^{\frac{N}{4}})$	[31]
			6	$\boldsymbol{O(N)}$	**Section 5.1**
GFS-4F	GFS	qCPA	8	$O(N)$	[31]
			10	$\boldsymbol{O(N)}$	**Section 5.1**
Skipjack-B-type	GFS	qCPA	13	$O(N)$	[8]
			16	$\boldsymbol{O(N)}$	**Section 5.2**
LBlock	Feistel	qCPA	3	$O(N)$	[20]
		qCCA	4	$O(N)$	[16]
		qCPA	8	$O(N)$	[30]
		qCPA	**8**	$O(N)$	**Section 5.3**
		qCPA	**10**	$\boldsymbol{O(2^{\frac{N}{16}})}$	**Section 5.3**
TWINE	GFS	qCPA	**8**	$\boldsymbol{O(N)}$	**Section 5.3**
			10	$\boldsymbol{O(2^{\frac{N}{16}})}$	**Section 5.3**
SKINNY	SPN	qCPA	**7**	$\boldsymbol{O(N)}$	**Section 6.1**
			9	$\boldsymbol{O(2^{\frac{5N}{32}})}$	**Section 6.1**
CRAFT	SPN	qCPA	**8**	$\boldsymbol{O(N)}$	**Section 6.2**
			12	$\boldsymbol{O(2^{\frac{5N}{16}})}$	**Section 6.2**
Piccolo-type	Feistel-SP	qCPA	**4**	$\boldsymbol{O(N)}$	**Section 7**

Extending the periodic distinguishers to SPN structure. As another significant application of the automated model, this paper presents the first construction of periodic distinguishers for SPN structure. For SKINNY [2], we discover the first 7-round polynomial-time periodic distinguisher and the first 9-round exponential-time periodic distinguisher. For CRAFT [3], we achieve the first 8-round polynomial-time periodic distinguisher and the first 12-round exponential-time periodic distinguisher. Furthermore, for the Feistel-SP structure, such as Piccolo [25] with MDS matrices, we extend the models to in-

corporate the propagation through the MDS. These advancements significantly broaden the applicability of Simon's algorithm in symmetric-key cryptanalysis and are expected to influence the design of future structures. Table 1 summarizes these results.

1.2 Organization

In Section 2, we provide essential definitions and an introduction to quantum algorithms. In Section 3, we present the approach for constructing periodic distinguishers. The automated SMT-based searching model is introduced in Section 4. Section 5 and Section 6 introduce the application of the model. Section 7 discusses the model for MDS matrices. Section 8 concludes the paper with a summary of our theory.

2 Preliminaries

2.1 Notation

We define $E : \{0,1\}^N \to \{0,1\}^N$ as a block cipher with a block size of N. We also define $f : \{0,1\}^n \to \{0,1\}^n$ as a Boolean function. The unitary operator U_f is defined as $U_f : \sum_{x,y} |x\rangle|y\rangle \to \sum_{x,y} |x\rangle|y \oplus f(x)\rangle$. For a cipher, f_j^i denotes the j-th round function applied in the i-th round. When there is only one function in each round, we denote it as f^i instead of f_0^i.

2.2 Simon's Algorithm

Consider a function $f : \{0,1\}^n \to \{0,1\}^n$ that is guaranteed to be periodic with period s, meaning $f(x) = f(x \oplus s)$ for a non-zero s. The goal of Simon's algorithm is to determine the period s. The algorithm is shown in Algorithm 1. Kaplan et al. [18] show that after cn queries, the period s can be recovered.

Theorem 1 ([18]). *If* $p_0 = \max_{t \in \{0,1\}^n \setminus \{0,s\}} \Pr_x[f(x) = f(x \oplus t)] < 1$, *then Simon's algorithm returns* s *with* cn *queries and* $O(n)$ *qubits, with probability at least* $1 - (2(\frac{1+p_0}{2})^c)^n$.

Usually, we have $p_0 < \frac{1}{2}$. Otherwise, f will exhibit highly non-random characteristics. Under this assumption, when we apply Simon's algorithm to f, it returns s with a probability of at least $1 - 2^n \cdot \left(\frac{3}{4}\right)^{cn}$.

Distinguisher without recovering the period. Ito et al. [16] further relax the condition by focusing on the dimension of the vector space. Let C be a block cipher E or a random permutation Π ($\{0,1\}^n \to \{0,1\}^n$). Obtaining $\eta = O(n)$ vectors, if the dimension of Y is not n, we can distinguish E from Π with a high probability. Algorithm 2 shows the process. The algorithm simplifies the analysis by removing the requirement to bound the probability of collisions other

Algorithm 1 The process of Simon's algorithm

1: Prepare two n-qubit registers initialized to the zero state $|0\rangle^{\otimes n}|0\rangle^{\otimes n}$.
2: Apply a Hadamard transform to the first register:

$$(H^{\otimes n} \otimes I)\,|0\rangle^{\otimes n}|0\rangle^{\otimes n}\,.$$

3: Apply the oracle U_f, we have

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle\,|f(x)\rangle\,.$$

4: Measure the second register in the computational basis yields a value $f(z)$ and collapses the first register to the state:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|z\rangle + |z \oplus s\rangle).$$

5: Apply again the Hadamard transform $H^{\otimes n}$ to the first register yields

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2^n}}\sum_{y\in\{0,1\}^n}(-1)^{y\cdot z}\,(1+(-1)^{y\cdot s})\,|y\rangle.$$

6: Measure the first register to get the output value y, which is guaranteed to satisfy $y \cdot s = 0$.

than the period, which is adopted by this paper to search for the period of the function.

Truncated outputs of quantum oracles. Hosoyamada and Sasaki [15] introduced the truncated outputs of quantum oracles, which can be used to obtain truncated outputs $E|_u$, where $E|u$ represents the output of E truncated to the branch u. This paper focuses more on how to search for periods and therefore does not emphasize circuit implementation.

2.3 Grover-Meet-Simon Algorithm

At ASIACRYPT 2017, Leander and May [21] proposed the Grover-meet-Simon algorithm, which combines Simon's algorithm with Grover's algorithm. Based on this work, Bonnetain et al. [5] reduced the number of queries by reusing internal states. We now provide a brief introduction to the problem.

Definition 1 (Grover-meet-Simon problem). *Let $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^d$ be a function such that there exists some $u \in \{0,1\}^m$ for which $f(u,\cdot)$ hides a non-trivial period s_u with a probability of 2^{-m}. The goal is to find any tuple $(u, s_u) \in U_s$, where $U_s := \{(u, s_u) : u \in \{0,1\}^m, s_u$ is the period of $f(u,\cdot)\}$.*

The Grover-meet-Simon algorithm operates as follows:

– Attacker makes a guess for u (this forms the Grover part of the algorithm).

Algorithm 2 Distinguisher without recovering the period

Prepare an empty set Y.
for $1 \leq i \leq \eta$ **do**
    Following the process of Simon's algorithm, measure the first register and add the obtained vector y to Y.
end for
Calculate the dimension d of the vector space spanned by Y.
if $d = n$ **then**
    return C is Π.
else
    return C is E.
end if

– For the correct guess, the attacker identifies a periodic function, which is then detected using Simon's algorithm.

Grover's algorithm serves as an outer loop with a running time of approximately $O(2^{m/2})$, while Simon's algorithm acts as an inner loop with polynomial complexity.

Theorem 2 (Proposition 2 in [5]). *Suppose that m is in $O(n)$. The computation of finding u is done in time $O\left((n^3 + nT_f)2^{m/2}\right)$, where T_f is the time required to evaluate f once.*

In this paper, we are more focused on how to automate the search for periodic distinguishers. Thus, we assume that there exists a $O(1)$-time quantum circuit capable of efficiently implementing a function f or a structure E and simplify the complexity as $O(2^{\frac{m}{2}})$, ignoring the polynomial factors in Theorem 2. We leave other details, such as how to implement the function s on a quantum computer, for future work.

2.4 Link between Periodic Functions and Truncated Differentials

We first introduce the distinguisher from Kuwakado and Morii [20], and then provide a simple example of Xiang et al.'s technique below. Kuwakado and Morii's work distinguishes the Feistel structure (see Fig. 1) from a random permutation. The three-round Feistel structure is defined by $(x_L^3, x_R^3) = E(x_L^0, x_R^0)$, where $E : \{0,1\}^{2n} \to \{0,1\}^{2n}$ and the two halves of the input are denoted as x_L^0 and x_R^0, then for round i, the transformation is given by:

$$x_L^i = x_R^{i-1} \oplus f^i(x_L^{i-1}), \quad x_R^i = x_L^{i-1},$$

where f^i is the round function. They define a function $f(b, x)$ that takes as input a bit b and a string x, XORed with two arbitrary constants α_0 and α_1:

$$\begin{aligned} f : \{0,1\} \times \{0,1\}^n &\to \{0,1\}^n, \\ b, x &\mapsto x_R^3 \oplus \alpha_b, \\ f(b, x) &= f^2(x \oplus f^1(\alpha_b)). \end{aligned}$$

$f(b, x)$ satisfies the periodic property required by Simon's algorithm and $s = 1\|(f^1(\alpha_0) \oplus f^1(\alpha_1))$ is the period of f:

$$f(b, x) = f(b \oplus 1, x \oplus f^1(\alpha_0) \oplus f^1(\alpha_1)).$$

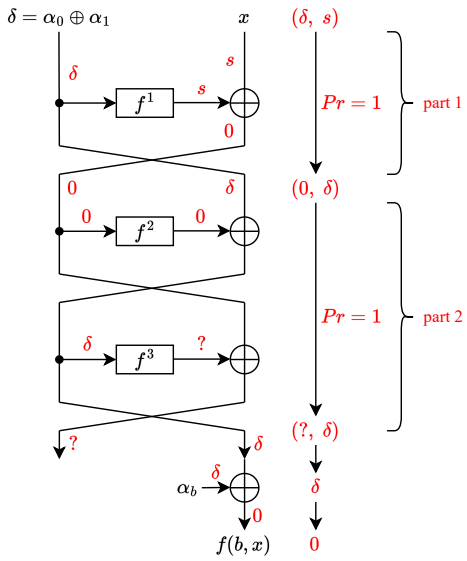

Fig. 1: 3-round Feistel structure with the truncated differential periodicity.

At ToSC 2024, Xiang et al. studied the link between periodic functions and truncated differentials [30]. In Xiang et al.'s theory [30], the distinguisher is divided into two parts. Part 1 is a truncated differential $(\delta, s) \to (s \oplus f^1(\delta), \delta)$, where $\delta = \alpha_0 \oplus \alpha_1$ and Part 2 is a truncated differential $(0, \delta) \to (?, \delta)$, where $f^1(\delta)$ is defined by $f^1(\alpha_0) \oplus f^1(\alpha_1)$. By setting $x = f^1(\alpha_0) \oplus f^1(\alpha_1)$, the output of the truncated differential in Part 1 will connect with the input of the truncated differential in Part 2. Then, we can find a 3-round periodic distinguisher $(\delta, s) \to (?, \delta)$. The second δ is used to construct the periodic function in Xiang et al.'s truncated differential theory.

3 Generalized Method of Periodic Construction

In this section, we present a generalized method for constructing periodic distinguishers. It consists of two theorems:

– Theorem 3: A unified method to construct quantum periodic distinguishers in polynomial time.

– Theorem 4: New probabilistic periodic distinguisher. The distinguisher holds only when α_0 and α_1 satisfy certain properties, where α_0 and α_1 are distinct constants.

3.1 A General Method for Constructing Periodic Distinguishers

We note that there are various methods for constructing periodic functions, which complicates the automatic search for suitable distinguishers. To address this, we first define the concepts of periodic functions and periodic distinguishers.

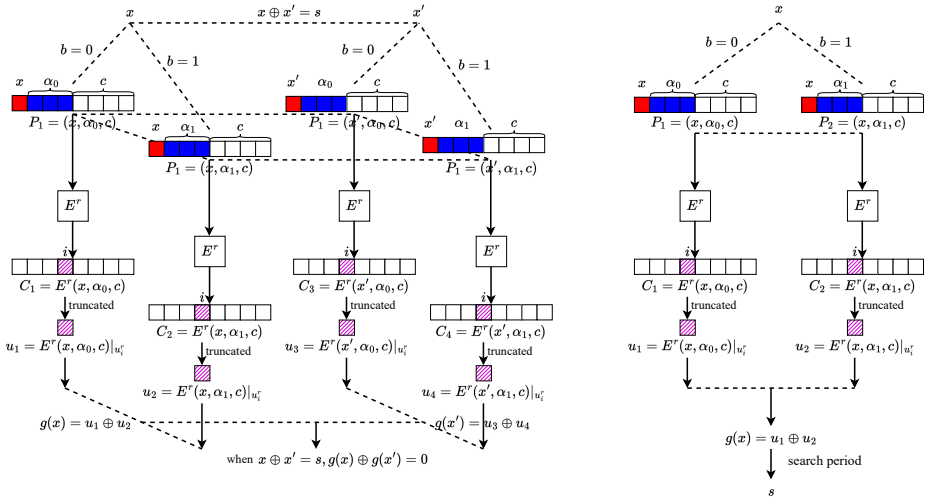

Fig. 2: The process of the construction of periodic distinguisher.

Let $E^r : (u_0^0, u_1^0, \ldots, u_{t-1}^0) \to (u_0^r, u_1^r, \ldots, u_{t-1}^r)$ denote an r-round function with t branches, where each branch is n bits, and u_j^i represents the j-th branch during the i-th round. Assume that the input of E^r can be written with disjoint variables as

$$(x, \alpha_b, c) \in \{0,1\}^n \times \{0,1\}^{e_1} \times \{0,1\}^{e_2}, \;\; n + e_1 + e_2 = tn = N, \;\; e_1 \geq 1, e_2 \geq 0,$$

where x is a variable, while α_b and c are constants. For convenience, we use $\Delta\alpha_b$ and $\Delta f(\alpha_b)$ to represent the differences of $\alpha_0 \oplus \alpha_1$ and $f(\alpha_0) \oplus f(\alpha_1)$, respectively. For any constant t, we have $\Delta t = 0$. Fig. 2-left illustrates the core idea of the periodic distinguisher. Define the periodic function g:

$$\begin{aligned} g : \{0,1\}^n &\to \{0,1\}^n, \\ x &\mapsto \Delta u_i^r, \\ g(x) &= E^r(x, \alpha_0, c)|_{u_i^r} \oplus E^r(x, \alpha_1, c)|_{u_i^r}, \end{aligned} \tag{1}$$

where Δu_i^r is the difference of the values u_i^r when $b = 0$ and $b = 1$. The construction method of the periodic function will be explained in Theorem 3. We note that a similar function also appears in [27].

When the period s is known, Algorithm 3 can be applied in the distinguishing attack.

Algorithm 3 The process of the distinguishing attack.

1: Choose $\alpha_0 \neq \alpha_1$ and c randomly.
2: Construct a quartet of plaintexts (P_1, P_2, P_3, P_4). For $b = 0$, $P_1 = (x, \alpha_0, c)$; for $b = 1$, $P_2 = (x, \alpha_1, c)$. Assume that $s \in \{0,1\}^n$ is the possible period. Let $x' = x \oplus s$. Then, $P_3 = (x', \alpha_0, c)$ and $P_4 = (x', \alpha_1, c)$.
3: The ciphertexts are $C_1 = E^r(x, \alpha_0, c), C_2 = E^r(x, \alpha_1, c), , C_3 = E^r(x', \alpha_0, c)$, and $C_4 = E^r(x', \alpha_1, c)$.
4: We focus solely on the i-th branch, thus truncating the ciphertext to this branch. $u_1 = C_1|_{u_i^r}$, $u_2 = C_2|_{u_i^r}$, $u_3 = C_3|_{u_i^r}$, and $u_4 = C_4|_{u_i^r}$.
5: Compute $g(x) \oplus g(x')$. If s is the period of g, we have $g(x) \oplus g(x') = 0$ for any $x \in \{0,1\}^n$. Otherwise, this probability is negligible.

In the quantum setting, this process can be simplified (see Fig. 2-right). The period s does not need to be predetermined, and the quartet of plaintexts is not required. For $b = 0$ and $b = 1$, the encryption function E is accessed separately, and the outputs are XORed to obtain the periodic function g. Simon's algorithm can then be used to search for s from $g(x)$.

Based on the method shown in Fig. 2-right, we define the periodic distinguisher represented by

$$(s, \delta, 0) \to (?, \ldots, ?, 0, ?, \ldots, ?). \tag{2}$$

In the input, s denotes the site of x as $g(x) \oplus g(x \oplus s) = 0$, $\delta = \Delta\alpha_b$ represents the difference of α_b, and $0 = \Delta c$ is the difference of c. In the output, 0 corresponds to the branch u_i^r and can be used to construct the periodic function $g(x)$, while ? indicates a branch without periodicity.

In this paper, each periodic function corresponds to a unique periodic distinguisher. Therefore, unless otherwise specified, we assume that the two are interchangeable without ambiguity. Next, we introduce how to search for periodic distinguishers, or construct periodic functions.

3.2 Polynomial-Time Periodic Distinguisher

At PQCrypto 2020, Hodžić et al. [14] presented the separability property based on an observation regarding the construction of Simon's algorithm, including the strong separability property, semi-strong separability property, and weak separability property. However, the types of periodic functions are still limited, and it remains unclear which weak separability properties can be used for periodic construction.

We define four functions: $F, G, P, Q : \{0,1\}^* \to \{0,1\}^n$. Each function takes an input of arbitrary length and produces an n-bit output. $F(x, y)$ is defined as

a function that depends only on the input x and y. The same definition applies to G, P and Q. We now present the following definition.

Definition 2 (Differential separability property). *A branch u_i^r satisfies the differential separability property if u_i^r is represented as*

$$u_i^r = E^r(x, \alpha_b, c)|_{u_i^r} = F(x \oplus G(\alpha_b, c), c) \oplus P(x, c) \oplus Q(\alpha_b, c).$$

Compared with Hodžić et al.'s definition [14], our definition allows any unknown difference caused individually by α_b or x. It directly leads to their inability to construct a polynomial-time 5-round distinguisher (only 4 rounds in [14]), whereas our definition allows for the 5-round distinguisher.

In the following, Theorem 3 establishes that if a branch u_i^r satisfies the differential separability property, it can always be used to construct a periodic function. Next, we focus solely on the existence of a branch that satisfies the differential separability property.

Theorem 3 (Polynomial-time periodic distinguisher). *Given a block cipher E^r, if an output branch u_i^r satisfies the differential separability property, $g(x)$ is always a periodic function.*

Proof. $g(x)$ is defined by Equation (1). We have

$$\begin{aligned} g(x) &= E^r(x, \alpha_0, c)|_{u_i^r} \oplus E^r(x, \alpha_1, c)|_{u_i^r} \\ &= \Delta F(x \oplus G(\alpha_b, c), c) \oplus \Delta P(x, c) \oplus \Delta Q(\alpha_b, c) \\ &= \Delta F(x \oplus G(\alpha_b, c), c) \oplus \Delta Q(\alpha_b, c). \end{aligned}$$

Set $s = \Delta G(\alpha_b, c) = G(\alpha_0, c) \oplus G(\alpha_1, c)$, then for any x:

$$\begin{aligned} g(x \oplus s) &= \Delta F(x \oplus s \oplus G(\alpha_b, c), c) \\ &= \Delta F(x \oplus G(\alpha_b, c), c) \\ &= g(x). \end{aligned}$$

Through our construction, aside from $F(x \oplus G(\alpha_b, c), c)$, the other terms either depend only on x and thus get canceled out, or depend only on α_b, resulting in a fixed differential value $Q(\alpha_0, c) \oplus Q(\alpha_1, c)$. The period can always be set to $s = \Delta G(\alpha_b, c)$ to satisfy $g(x \oplus s) = g(x)$. □

Remark 1. As shown in Equation (2) and Theorem 3, we are only interested in the differences of x, c, and α_b, rather than their specific values. The following explains this in detail:

- Only when $x \oplus x' = s$, ensuring $g(x) \oplus g(x') = 0$. The function $F(x \oplus G(\alpha_b, c), c)$ is the essential characteristic determining periodicity. For any $\alpha_0 \neq \alpha_1$, the difference of x equals $\Delta G(\alpha_b, c)$, which is the period s.
- For α_0 and α_1, the only constraint is $\Delta\alpha_b = \alpha_0 \oplus \alpha_1 \neq 0$.
- The difference Δc is always zero, meaning c does not affect the construction of the periodicity in $g(x)$. Thus, we can omit c in our theory and set it as a constant 0 in the periodic distinguisher.

Example 1 (Improvement to Xiang et al.'s truncated differential theory in [30]). GFS-2F [24] is a type of GFS, proposed by Nyberg at ASIACRYPT 1996. Fig. 3 shows the improvement of our theory.

Using the truncated differential theory from [30], a 5-round distinguisher $(s, \delta, 0, 0) \xrightarrow{5r} (0, 0, ?, ?)$ is found. In the subsequent round, every branch has an unknown difference.

For the same input, a 6-round distinguisher $(s, \delta, 0, 0) \xrightarrow{6r} (0, ?, ?, ?)$ is identified based on Theorem 3. u_0^6 satisfies the differential separability property:

$$u_2^0 \oplus f_1^3(u_1^0 \oplus f_0^1(u_2^0)) \oplus f_0^4(u_3^0 \oplus f_1^2(u_2^0) \oplus f_0^3(u_0^0 \oplus f_1^1(u_3^0) \oplus f_0^2(u_1^0 \oplus f_0^1(u_2^0))))).$$

The corresponding periodic function is

$$\begin{aligned} g(x) &= E^6(x, \alpha_0, u_2^0, u_3^0)|_{u_0^6} \oplus E^6(x, \alpha_1, u_2^0, u_3^0)|_{u_0^6} \\ &= \Delta f_1^3(\alpha_b \oplus C_1) \oplus \Delta f_0^4(C_2 \oplus f_0^3(x \oplus f_0^2(\alpha_b \oplus C_3) \oplus C_4)), \end{aligned}$$

where u_2^0, u_3^0 are random constants, $C_1 = f_0^1(u_2^0)$, $C_2 = u_3^0 \oplus f_1^2(u_2^0)$, $C_3 = f_0^1(u_2^0)$, and $C_4 = f_1^1(u_3^0)$ are also constants. Then, set the period as $s = \Delta f_0^2(\alpha_b \oplus C_3)$. For any x, we have $g(x \oplus s) \oplus g(x) = 0$.

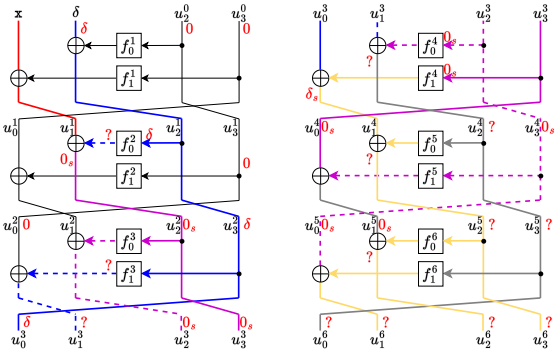

Fig. 3: The 6-round periodic distinguisher for GFS-2F.

3.3 Probabilistic Periodic Distinguisher

In this section, we utilize the collisions of $\Delta\alpha_b$ to enhance the periodic distinguisher. The difference of initial $\Delta\alpha_b$ affects the propagation of the periodic distinguisher, even though these values were chosen randomly in Kuwakado and Morii's work [20]. Thus, we propose a method to construct the probabilistic periodic distinguisher, which can be solved by the Grover-meet-Simon algorithm.

Definition 3 (Probabilistic periodic distinguisher problem). *For any* x, α_0, α_1, c, *let* $f : \{0,1\}^n \times \{0,1\}^{e_1} \times \{0,1\}^{e_1} \times \{0,1\}^{e_2} \to \{0,1\}^n$ *be a function such that there exists some* $(\alpha_0, \alpha_1) \in \{0,1\}^{2e_1}$ *for which* $f(\cdot, \alpha_0, \alpha_1, c)$ *hides a non-trivial period* s_u *with a probability of* 2^{-p}. *The goal is to find any tuple* $(\alpha_0, \alpha_1, s_u) \in U_s$, *where* $U_s := \{(\alpha_0, \alpha_1, s_u) : (\alpha_0, \alpha_1) \in \{0,1\}^{2e_1}, s_u$ *is the period of* $f(\cdot, \alpha_0, \alpha_1, c)\}$.

The simple example below is a demonstration of the probabilistic periodic distinguisher.

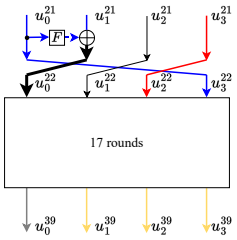

Fig. 4: The 18-round periodic distinguisher of CAST-256.

Example 2 (18-round Distinguisher for CAST-256 [1]). In [27], Sun et al. proposed a 17-round periodic distinguisher of CAST-256: $(0,0,s,\delta) \to (0,?,?,?)$ from $(u_0^{22}, u_1^{22}, u_2^{22}, u_3^{22})$ to $(u_0^{39}, u_1^{39}, u_2^{39}, u_3^{39})$.

With the new technique, we found a 18-round probabilistic periodic distinguisher $(\delta^0, \delta^1, 0, s) \to (0,?,?,?)$, as shown in Fig. 4, where δ is from two branches by $\delta = \delta^0 || \delta^1$. We have $\delta^0 = \alpha_0^0 \oplus \alpha_1^0$, $\delta^1 = \alpha_0^1 \oplus \alpha_1^1$, and $\alpha_b = \alpha_b^0 || \alpha_b^1$. Only the constraint $\Delta F(\alpha_b^0) = \Delta \alpha_b^1$ holds, u_0^{22} is 0 in the distinguisher because of $\Delta u_0^{22} = 0$. Then, u_0^{39} can satisfy the differential separability property. The probability is 2^{-n}, where n is the length of one branch. Using the Grover-meets-Simon algorithm, we can find the period with a complexity of $O(2^{n/2})$ (ignoring polynomial factors). The full distinguishers are shown in Appendix J.

We refer to this phenomenon, in which α_0, α_1 must satisfy specific conditions, as *a collision*. The collision resets the zero difference in the periodic function.

Accordingly, we propose the following theorem about the probabilistic periodic distinguisher using the same periodic function. The proof is similar to Theorem 3.

Theorem 4 (Probabilistic periodic distinguisher). *If after some collisions with probability* 2^{-p}, *a branch* u_i^r *satisfies the differential separability property,* u_i^r *can be used to construct an* r*-round periodic distinguisher with probability* 2^{-p}.

Discussion on the theoretical probability and the practical collision probability. Given the randomness of the key, for any two arbitrary functions f_1 and f_2, the values $f_1(\alpha_0)$, $f_1(\alpha_1)$, $f_2(\alpha_0)$, and $f_2(\alpha_1)$ are unpredictable. On average, the probability of a collision, where $\Delta f_1(\alpha_b) = \Delta f_2(\alpha_b)$ in two n-bit branches, is about 2^{-n}. We use this average probability as an estimate of

complexity. On the other hand, if multiple collisions need to be satisfied, it is essential to determine which constraints are necessary. Therefore, we provide two algorithms to evaluate the probability.

- Algorithm 4 describes the procedure for determining the theoretical probability. The probability can be computed based on the rank of all collisions. If Algorithm 4 detects incompatible constraints, the probability is 0. If the two constraints require only a single collision, Algorithm 4 allows us to reduce the theoretical complexity by computing the rank.
- Algorithm 5 uses data to validate the practical collision probability. This test is based on the instantiation of the structure, which is similar to [7], where we randomly generate permutations, constants, keys, etc., and then perform repeated testing. If the rank is ℓ, we conduct $2^{n\cdot\ell+4}$ tests. On average, a period occurs approximately 2^4 times.

Algorithm 4 The process of calculate the theoretical probability.

Input: The collisions $c_0, c_1, \ldots, c_{\ell-1}$, the number of bits n in the branch.
Output: The theoretical probability of the collisions.
1: Form a system of equations $S = \{c_0, c_1, \ldots, c_{\ell-1}\}$.
2: **if** the system of equations S has incompatible constraints **then**
3: **return** 0.
4: **end if**
5: Compute the rank of S, denoted by r_S.
6: **return** the theoretical probability $2^{-n\cdot r_s}$.

Here, we adopt an assumption from traditional distinguisher searches. The constraints of differences in different rounds are assumed to be independent after passing through the random round keys and S-boxes. In Appendix C, we present the experiments on distinguishers generated by the automated model between the theoretical distinguishers and the practical probability.

Next, we provide the constraints on the degrees of freedom, which are determined by the number of initial choices of α_0 and α_1. Note that an e_1-bit α_b can introduce 2^{2e_1} degrees of freedom. If the probability of the periodic distinguisher is 2^{-p}, then on average, 2^p choices are required to obtain a feasible solution, which costs 2^p degrees of freedom. Thus, we roughly set $p \leq 2e_1$, and the exact complexity can be computed by our algorithms.

We present an example of the probabilistic periodic distinguisher.

Example 3. Assume $u_i^r = F(x \oplus G(\alpha_b, c), c) \oplus P(x, c) \oplus Q(\alpha_b, c) \oplus R_0(x, \alpha_b, c)$, where $R_0(x, \alpha_b, c) = R_0(R_1(x) \oplus R_2(\alpha_b) \oplus R_3(\alpha_b) \oplus R_4(c))$. According to Definition 2, u_i^r does not satisfy the separability property. Define the periodic function:

$$\begin{aligned} g(x) &= E^r(x, \alpha_0, c)|_{u_i^r} \oplus E^r(x, \alpha_1, c)|_{u_i^r} \\ &= \Delta F(x \oplus G(\alpha_b, c), c) \oplus \Delta R_0(x, \alpha_b, c) \oplus \Delta Q(\alpha_b, c). \end{aligned} \tag{3}$$

Algorithm 5 The process of the practical probability verification.

Input: The collisions $c_0, c_1, \ldots, c_{\ell-1}$ for the r-round cipher E^r, where each branch is n bits.

Output: The actual probability of the collisions.

1: Let T denote the number of successful verifications.
2: **for** t from 0 to $2^{n\cdot\ell+4}$ **do**
3: Generate random values of $x \in \{0,1\}^n, \alpha_0 \in \{0,1\}^{e_1}, \alpha_1 \in \{0,1\}^{e_1}, c \in \{0,1\}^{e_2}$ and random keys.
4: Calculate $E^r(x, \alpha_0, c)$ and $E^r(x, \alpha_1, c)$ and verify whether the intermediate values satisfy the system of equations S or verify whether the period exists. If it is satisfied, let $T = T + 1$.
5: **end for**
6: **return** the practical probability $\frac{T}{2^{n\cdot\ell+4}}$.

If $\Delta R_2(\alpha_b) = \Delta R_3(\alpha_b)$ holds, we have $R_2(\alpha_0) \oplus R_3(\alpha_0) = R_2(\alpha_1) \oplus R_3(\alpha_1) = c'$, where c' is a constant. Equation (3) reduces to $g(x) = \Delta F(x \oplus G(\alpha_b, c), c) \oplus \Delta Q(\alpha_b, c)$, which has a period $\Delta G(\alpha_b, c)$. The probability is 2^{-n}. When α_0, α_1 do not satisfy the condition, the probability of $g(x \oplus s) = g(x)$ becomes negligible. For the degrees of freedom, 2^{2n} is sufficiently large for the attack.

4 Automated SMT-based Searching Model

Automating the search for periodic distinguishers remains an open problem for complex structures.

- As the number of rounds increases, the cipher function becomes more complex, making it increasingly difficult to determine the existence of a period by exhaustively searching all possible inputs.
- Manually deriving a probabilistic periodic distinguisher is also extremely challenging.

In this section, we achieve the automated search for periods for complex structures. We first introduce several symbols that represent all types of states in a distinguisher, simplifying the search process. With these new symbols, all states and their combinations can be easily represented. We then propose the first automated SMT-based model to address this open problem.

4.1 Symbols Used in Our Model

Before introducing the model, we define all relevant operations and specify the three operations used in it.

- `R`: the keyed round function R taking a round key.
- `XOR`: the XOR operation between two branches.
- `SPLIT`: the branching operation.

We omit the round key and constant in our model, which does not affect the periodicity of the function.

To identify the separability property, 7 types of states are defined based on Definition 2 (see Table 2). We use the symbol $\perp$ (the state that loses its periodic properties) and 5-bit variables (the leftmost bit is the least significant bit), corresponding to $(\mathtt{0_s}, \mathtt{R(x)}, \mathtt{x}, \mathtt{R}(\delta), \delta)$, where $\mathtt{0_s}$ indicates a function like "$F(x \oplus G(\alpha_b, c), c)$", to represent all the states.

Table 2: Notations for used symbols in our model.

Symbol	Encoding	Meaning
$\mathtt{0}$	$(0,0,0,0,0)$	The zero difference
δ	$(0,0,0,0,1)$	$\delta = \Delta\alpha_b$ where α_0, α_1 are chosen randomly
$\mathtt{x}$	$(0,0,1,0,0)$	The period difference s between x and $x \oplus s$
$\mathtt{R(x)}$	$(0,1,0,0,0)$	Indicating a function like "$P(x, c)$"
$\mathtt{R}(\delta)$	$(0,0,0,1,0)$	Indicating a function like "$Q(\alpha_b, c)$"
$\mathtt{0_s}$	$(1,0,0,0,0)$	Indicating a function like "$F(x \oplus G(\alpha_b, c), c)$"
?	$\perp$	The unknown difference

According to Definition 2 and Theorem 3, a branch u_j^i has the periodicity when it can be represented by $u_j^i = F(x \oplus G(\alpha_b, c), c) \oplus P(x, c) \oplus Q(\alpha_b, c)$. States $\mathtt{0_s}$, $\mathtt{R(x)}$, and $\mathtt{R}(\delta)$ represent $F(x \oplus G(\alpha_b, c), c)$, $P(x, c)$, and $Q(\alpha_b, c)$, respectively. u_j^i can be written by $\mathtt{0_s} \oplus \mathtt{R(x)} \oplus \mathtt{R}(\delta)$, which is encoded by $(1,1,0,1,0)$, where the meaning of the encoding is the XOR value of the terms corresponding to the bits being 1. In general, $(1,*,*,*,*)$ always satisfies the separability property, where $*$ represents either 0 or 1. We can always obtain a path that uses our symbols

$$(\mathtt{x}, \delta, \mathtt{0}) \rightarrow (?, \ldots, ?, \mathtt{0_s} \oplus *, ?, \ldots, ?),$$

where $*$ represents any state except from ?. The path corresponds to a periodic distinguisher $(s, \delta, 0) \rightarrow (?, \ldots, ?, 0, ?, \ldots, ?)$ (see Equation (2)).

Using our symbolic representation, the process of searching for periods becomes more efficient, eliminating the need to handle numerous algebraic equations. Even without the automated model, employing these symbols enables faster identification of periodic distinguishers compared to manual methods.

4.2 SMT-based Automated Model

In this section, we propose a unified automated model to identify branches that satisfy the separability property from complex polynomials derived from multiple inputs. Our approach mainly uses the SMT solver, STP, to solve these models. Further details on the SMT problem are provided in Appendix A.

The complete propagation rules of our model are provided below, including the initial constraints and the propagation rules of `SPLIT`, `R`, and `XOR`.

Modelling the initial constraints. Assume that there is an r-round path with n branches: $(u_0^0, u_1^0, \ldots, u_{n-1}^0) \xrightarrow{r} (u_0^r, u_1^r, \ldots, u_{n-1}^r)$. The model is subject to the following constraints:

1. Each u_i^0 $(0 \le i \le n-1)$ must be one of $\{\mathtt{x}, \delta, \mathtt{0}, \mathtt{x} \oplus \delta\}$.
2. At least one element in $(u_0^0, u_1^0, \ldots, u_{n-1}^0)$ must be δ, and $\mathtt{x}$ or $\mathtt{x} \oplus \delta$ can appear only once.
3. At least one element in $(u_0^r, u_1^r, \ldots, u_{n-1}^r)$ must be $(1, *, *, *, *)$ or $(0, 0, 1, *, *)$.

Constraints 1 and 2 ensure that the initial values are valid. Constraint 3 ensures that the tail satisfies the separability property or still maintains a direct relationship with the input. The encoding $(0, 0, 1, *, *)$ actually represents a weakened function "$F(x \oplus G(\alpha_b, c), c)$", where F is identify function. Since $(0, 0, 1, *, *)$ only occurs in the case of a small number of rounds, we ignore it in the tail of our model. Based on these constraints, we can always find a path from $\{\mathtt{x}, \delta, \mathtt{0}, \mathtt{x} \oplus \delta\}$ to $\mathtt{0_s} \oplus *$.

Modelling the propagation rules of `SPLIT`. Let `SPLIT` duplicate a state without modifying it. That is, for each state u, the output states v, w of `SPLIT` satisfy $u = v = w$.

Modelling the propagation rules of `R`. `R` indicates that the current state undergoes a round function. The following property shows the rules.

Property 1. The complete rules of `R` are:

$$\mathtt{x},\ \mathtt{R(x)},\ \mathtt{R(x)} \oplus \mathtt{x} \xrightarrow{\mathtt{R}} \mathtt{R(x)}, \tag{4}$$

$$\delta,\ \mathtt{R}(\delta),\ \mathtt{R}(\delta) \oplus \delta \xrightarrow{\mathtt{R}} \mathtt{R}(\delta), \tag{5}$$

$$\mathtt{0} \xrightarrow{\mathtt{R}} \mathtt{0},\ \mathtt{0_s} \xrightarrow{\mathtt{R}} \mathtt{0_s}, \tag{6}$$

$$\mathtt{x} \oplus \delta,\ \mathtt{x} \oplus \mathtt{R}(\delta),\ \mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta \xrightarrow{\mathtt{R}} \mathtt{0_s}\ \text{(only once) or}\ \perp\ \text{(others)}, \tag{7}$$

$$\text{others} \xrightarrow{\mathtt{R}} \perp. \tag{8}$$

Proof. After applying the round function `R`, the new states $\mathtt{R(x)}$ and $\mathtt{R}(\delta)$ represent the outputs for $\mathtt{x}$ and δ, respectively. Assume that $\mathtt{R(x)} \oplus \mathtt{x}$ represents $P(x, c) \oplus x$. $\mathtt{R}(\mathtt{R(x)} \oplus \mathtt{x})$ is also $\mathtt{R(x)}$, representing $P'(x, c) := R(P(x, c) \oplus x)$, where P' is a function. In general, we define any function that only contains x and c as $\mathtt{R(x)}$. Although this may result in $R(x)$ not being a permutation, it generally does not affect the propagation of the period. Our model is more concerned with which initial values are included in this branch. The same rule applies to δ. Thus, Equations (4) and (5) are proved.

Next, we prove Equation (6). The transformation $\mathtt{0} \xrightarrow{\mathtt{R}} \mathtt{0}$ is trivial. By definition, assume that $\mathtt{0_s}$ represents a term $F(x \oplus G(\alpha_b, c), c)$, where the period is $\Delta G(\alpha_b, c)$. The output of `R` is $F'(x \oplus G(\alpha_b, c), c) := R(F(x \oplus G(\alpha_b, c), c))$, where F' is a function. This still satisfies the definition of $\mathtt{0_s}$, and the period $\Delta G(\alpha_b, c)$ remains unchanged.

Next, we prove Equation (7). We assume that multiple occurrences can happen from $\mathtt{x} \oplus \delta$, $\mathtt{x} \oplus \mathtt{R}(\delta)$, $\mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta$ to $\mathtt{0_s}$. Let us consider a counterexample:

– A branch u with the state $\mathtt{x} \oplus \mathtt{R}(\delta)$ represents the term $x \oplus G_1(\alpha_b, c)$. After `R`, we have $u' = R(x \oplus G_1(\alpha_b, c))$, where the implied period is $\Delta G_1(\alpha_b, c)$.

– A branch v with the state $\mathtt{x} \oplus \mathtt{R}(\delta)$ represents $x \oplus G_2(\alpha_b, c)$. After $\mathtt{R}$, we have $v' = R(x \oplus G_2(\alpha_b, c))$, where the implied period is $\Delta G_2(\alpha_b, c)$.

If two branches u' and v' have the states $\mathtt{0_s}$, errors may arise in state propagation. Let us discuss this case by case.

– $\Delta G_1(\alpha_b, c) = \Delta G_2(\alpha_b, c)$. The two branches have the same period. $w := u' \oplus v' = R(x \oplus G_1(\alpha_b, c)) \oplus R(x \oplus G_2(\alpha_b, c))$. We construct the periodic function $g(x) = \Delta R(x \oplus G_1(\alpha_b, c)) \oplus \Delta R(x \oplus G_2(\alpha_b, c))$. Setting $s = \Delta G_1(\alpha_b, c) = \Delta G_2(\alpha_b, c)$, we have $g(x \oplus s) = R(x \oplus s \oplus G_1(\alpha_0, c)) \oplus R(x \oplus s \oplus G_1(\alpha_1, c)) \oplus R(x \oplus s \oplus G_2(\alpha_0, c)) \oplus R(x \oplus s \oplus G_2(\alpha_1, c)) = \Delta R(x \oplus G_2(\alpha_b, c)) \oplus \Delta R(x \oplus G_1(\alpha_b, c)) = g(x)$. $u' \oplus v' \to w$ implies the rule $\mathtt{0_s} \oplus \mathtt{0_s} \to \mathtt{0_s}$.
– $\Delta G_1(\alpha_b, c) \neq \Delta G_2(\alpha_b, c)$. The two branches have different periods. $w := u_1 \oplus u_2 = R(x \oplus G_1(\alpha_b, c)) \oplus R(x \oplus G_2(\alpha_b, c))$ loses its periodicity. This situation must be avoided, as $\mathtt{0_s}$ from different branches should not interact.

Thus, we allow the occurrence of $\mathtt{x} \oplus \delta$, $\mathtt{x} \oplus \mathtt{R}(\delta)$, $\mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta \xrightarrow{\mathtt{R}} \mathtt{0_s}$ through $\mathtt{R}$ to happen only once, which ensures that the case $\Delta G_1(\alpha_b, c) \neq \Delta G_2(\alpha_b, c)$ will not happen. Although we may lose some precision for certain special structures, this constraint ensures that the $\mathtt{0_s}$ in the model always implicitly share the same period. As a generalized model, we consider this to be necessary.

Now, we can ensure that the first $\mathtt{0_s}$ is always generated by Equation (7) and all other $\mathtt{0_s}$ are either copies or combinations of $\mathtt{0_s}$ by $\mathtt{0_s} \xrightarrow{\mathtt{R}} \mathtt{0_s}$ and $\mathtt{0_s} \oplus \mathtt{0_s} = \mathtt{0_s}$, containing the same periods.

Equation (8) shows that for all branches where the period cannot be identified, we denote them by $\perp$. □

In the proof, we need to add new constraints for Equation (7). We assign a Boolean variable s_i for each application of $\mathtt{R}$, let $\Sigma(s_i) = 1$, and set

$$\begin{cases} s_i = 1, & \text{if } \mathtt{x} \oplus \delta \xrightarrow{\mathtt{R}} \mathtt{0_s} \text{ or } \mathtt{x} \oplus \mathtt{R}(\delta) \xrightarrow{\mathtt{R}} \mathtt{0_s} \text{ or } \mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta \xrightarrow{\mathtt{R}} \mathtt{0_s}, \\ s_i = 0, & \text{if } \mathtt{x} \oplus \delta \xrightarrow{\mathtt{R}} \perp \text{ or } \mathtt{x} \oplus \mathtt{R}(\delta) \xrightarrow{\mathtt{R}} \perp \text{ or } \mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta \xrightarrow{\mathtt{R}} \perp. \end{cases}$$

Modelling the propagation rules of XOR. For XOR, we need to consider the possibility of collisions. Let the input states be $u = (u_0, u_1, u_2, u_3, u_4)$ and $v = (v_0, v_1, v_2, v_3, v_4)$, and the output state be $w = (w_0, w_1, w_2, w_3, w_4)$.

Note that only w_3 and w_4 may be affected by the collisions. We first consider the rules for w_0, w_1, w_2, which are established in Property 2.

Property 2. If $u = \perp$ or $v = \perp$, then $w = \perp$. Otherwise, the complete rules of (w_0, w_1, w_2) are:

$$w_0 = \max(u_0, v_0), \quad w_1 = \max(u_1, v_1), \quad w_2 = u_2 \oplus v_2.$$

Proof. Fig. 5 shows the propagation of states. In the proof of Property 1, we have proven $\mathtt{0_s} \oplus \mathtt{0_s} = \mathtt{0_s}$. As long as the two branches with states $\mathtt{0_s}$ imply the same period, the equation $\mathtt{0_s} \oplus \mathtt{0_s} = \mathtt{0_s}$ always holds. The constraint is satisfied by Equation (7). We prove a more generalized case for different round functions:

– u has the state $\mathtt{0_s}$, representing $F_1(x \oplus G(\alpha_b, c), c)$,
– v has the state $\mathtt{0_s}$, representing $F_2(x \oplus G(\alpha_b, c), c)$,

where F_1 and F_2 are different functions. Equation (7) that u and v have the same implied period $\Delta G(\alpha_b, c)$. Let $w := u \oplus v = F_1(x \oplus G(\alpha_b, c), c) \oplus F_2(x \oplus G(\alpha_b, c), c)$. We prove that the branch w also has the same period. The periodic function is $g(x) = \Delta F_1(x \oplus G(\alpha_b, c), c) \oplus \Delta F_2(x \oplus G(\alpha_b, c), c)$. Setting $s = \Delta G(\alpha_b, c)$, $g(x \oplus s) = F_1(x \oplus s \oplus G(\alpha_0, c), c) \oplus F_1(x \oplus s \oplus G(\alpha_1, c), c) \oplus F_2(x \oplus s \oplus G(\alpha_0, c), c) \oplus F_2(x \oplus s \oplus G(\alpha_1, c), c) = g(x)$. Thus, $w = u \oplus v$ can be written by $F'(x \oplus G(\alpha_b, c), c)$ and also has the state $\mathtt{0_s}$. On the other hand, if the periods of u and v are different, this rule does not hold.

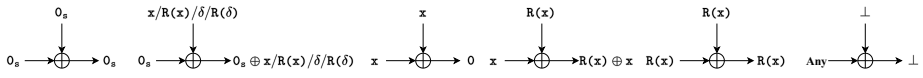

Fig. 5: The rules of (w_0, w_1, w_2).

The state $\mathtt{x}$ represents the term x. Thus, we have $\mathtt{x} \oplus \mathtt{x} \to \mathtt{0}$. According to the definition of $\mathtt{R(x)}$, $\mathtt{R(x)} \oplus \mathtt{R(x)} \to \mathtt{R(x)}$ is also trivial. □

Now, we consider the rules of w_3, w_4, i.e., $\mathtt{R}(\delta)$ and δ. In order to compute the number of collisions, we assign a Boolean variable x_i for each $\mathtt{XOR}$ operation. $x_i = 1$ indicates that the collision occurs. We first consider the rules without collisions. That is, x_i always be 0.

Property 3. The rules of $(u_3, u_4, v_3, v_4) \to (w_3, w_4, x_i)$ without collisions are:

$$\begin{cases} (0,0,v_3,v_4) => (v_3,v_4,0), \quad (u_3,u_4,0,0) => (u_3,u_4,0), \\ (0,1,0,1) => (0,0,0), \quad (0,1,1,1) => (1,0,0), \quad (1,1,0,1) => (1,0,0). \end{cases}$$

The proof is similar and therefore omitted. Fig. 6 shows the rules.

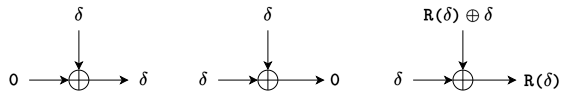

Fig. 6: The rules of (w_3, w_4, x_i) without collisions.

Next, we consider the propagation rules where collisions may occur.

Property 4. The rules of $(u_3, u_4, v_3, v_4) \to (w_3, w_4, x_i)$ with collisions are:

$$\begin{cases} ((1,0,0,1)\ \text{OR}\ (0,1,1,0)) => ((1,1,0)\ \text{OR}\ (0,0,1)), \\ ((1,1,1,1)\ \text{OR}\ (1,0,1,0)) => ((1,0,0)\ \text{OR}\ (0,0,1)), \\ ((1,1,1,0)\ \text{OR}\ (1,0,1,1)) => ((1,1,0)\ \text{OR}\ (0,0,1)\ \text{OR}\ (0,1,1)). \end{cases}$$

Proof. If there is no collision, the rule is the same as $\mathtt{R}(\mathtt{x})$ (see Fig. 7). When a collision occurs, we have the following properties. The output of $\delta \oplus \mathtt{R}(\delta)$ is $\mathtt{0}$. The output of $\mathtt{R}(\delta) \oplus \mathtt{R}(\delta)$ is $\mathtt{0}$. For the states $\mathtt{R}(\delta)$ and $\mathtt{R}(\delta) \oplus \delta$, there are two cases:

- $\mathtt{R}(\delta)$ and $\mathtt{R}(\delta)$ has a collision, the output is δ.
- $\mathtt{R}(\delta)$ and $\mathtt{R}(\delta) \oplus \delta$ has a collision, the output is $\mathtt{0}$.

□

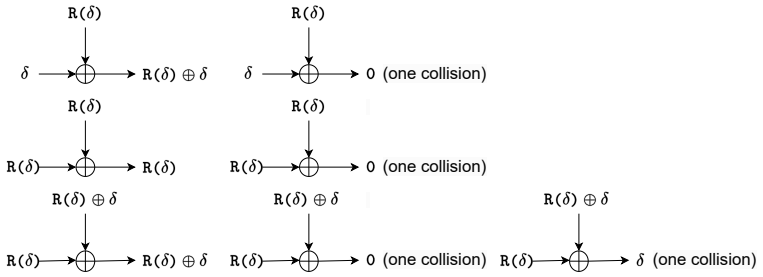

Fig. 7: The rules of (w_3, w_4, x_i) with collisions.

For Property 4, we define an integer variable X to count the number of collisions and set $\Sigma(x_i) = X$. On average, the probability of a collision occurring in each n-bit branch is 2^{-n}. Thus, the required probability of collisions is calculated by $2^{-n \cdot X}$. Additionally, we impose the constraint $X \leq t-1$, where t denotes the number of branches. If $X = 0$, a distinguisher with polynomial time complexity is returned. Once the distinguisher is generated, Algorithms 4 and 5 can provide the exact probability.

As an application, we reproduced various distinguishers for Type-1/2/3 GFS and achieved the highest number of rounds (see Appendix K).

Extending the tail of the periodic distinguisher. Notably, a linear combination of multiple non-periodic branches in the tail may sometimes produce a value that satisfies periodicity. While all possibilities can be manually verified after the model returns a periodic distinguisher, we introduce a mask encoding at the end of the current model to automate the search for potential linear combinations. This approach simplifies the tail verification process and enhances the model's completeness, although it does not yield improved results. This part of the content is placed in Appendix B. Skipping this encoding will not affect the understanding of the overall model.

5 Application on GFS-2F/4F, Skipjack-Type Structure, LBlcok, and TWINE

In this section, we apply the automated model to GFS-2F/4F, Skipjack-type structure, LBlcok, and TWINE, improving the distinguishers.

5.1 Application on GFS-2F/4F

GFS-2F and GFS-4F [24] are two types of GFSs proposed by Nyberg at ASIACRYPT 1996 that are resistant to both differential and linear attacks. In [31], the authors provided a 5-round periodic distinguisher for GFS-2F and a 8-round periodic distinguisher for GFS-4F. They also obtained a 6-round distinguisher for GFS-2F with complexity $2^{\frac{N}{4}}$.

Table 3: The periodic distinguishers of GFS-2F/4F.

Structure	#Rounds	Input	Output	Complexity	Reference
GFS-2F	5	-	-	$O(N)$	[31]
	6	-	-	$O(2^{\frac{N}{4}})$	[31]
	6	$(s, \delta, 0, 0)$	$(0, ?, ?, ?)$	$O(N)$	**Our model**
GFS-4F	8	-	-	$O(N)$	[31]
	10	$(s, \delta, 0, 0, 0, 0, 0, 0)$	$(0, ?, ?, ?, ?, ?, ?, ?)$	$O(N)$	**Our model**

Using our model, we find a new 6-round periodic distinguisher of GFS-2F without the need for probability, as well as the first 10-round periodic distinguisher of GFS-4F in complexity $O(N)$. Table 3 shows the input/output pattern of each distinguisher. The 6-round periodic distinguisher has been explained in Section 3.1. We introduce the 10-round distinguisher below, the other results are explained in Appendix D.

The 10-round periodic distinguisher of GFS-4F. We extensively use colors while depicting distinguishers. In order to have a better explanation, we take the 10-round distinguisher of GFS-4F (see Fig. 8) for illustration.

Solid red, blue, and black lines represent $\mathtt{x}$, δ and $\mathtt{0}$ in the model, respectively. Red and blue lines become dashed lines after passing through the round function $\mathtt{R}$. A solid purple line represents $\mathtt{x} \oplus \mathtt{R}(\delta)$, indicating the existence of an explicit period. After passing through $\mathtt{R}$, the explicit period will be hidden within the dashed purple line ($\mathtt{0_s}$). According to Property 7, the rule of (the solid purple line $\rightarrow$ the dashed purple line) occurs only once. Finally, A solid gray line ($\mathtt{0_s} \oplus *$) indicates which branch still retains the separability property. According to Theorem 3, we can construct the periodic function. The period s is $\Delta f_0^4(\alpha_b \oplus C_2)$, where $C_2 = f_2^1(u_6^0) \oplus f_1^2(u_4^0) \oplus f_0^3(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))$ is a constant. The complete periodic function is provided in Appendix D.

5.2 Application on Skipjack-Type Structure

Skipjack [23] is the encryption algorithm developed by the NSA for the Clipper chip and Fortezza PC card. The published description of Skipjack characterizes the rounds as either Rule A or Rule B (see Fig. 9). In [19], Knudsen and Wagner considered different combinations of the two rules. Furthermore, in [4], Blondeau et al. used the impossible differential attack and zero-correlation linear attack to show the 16-round distinguishers for two common variants:

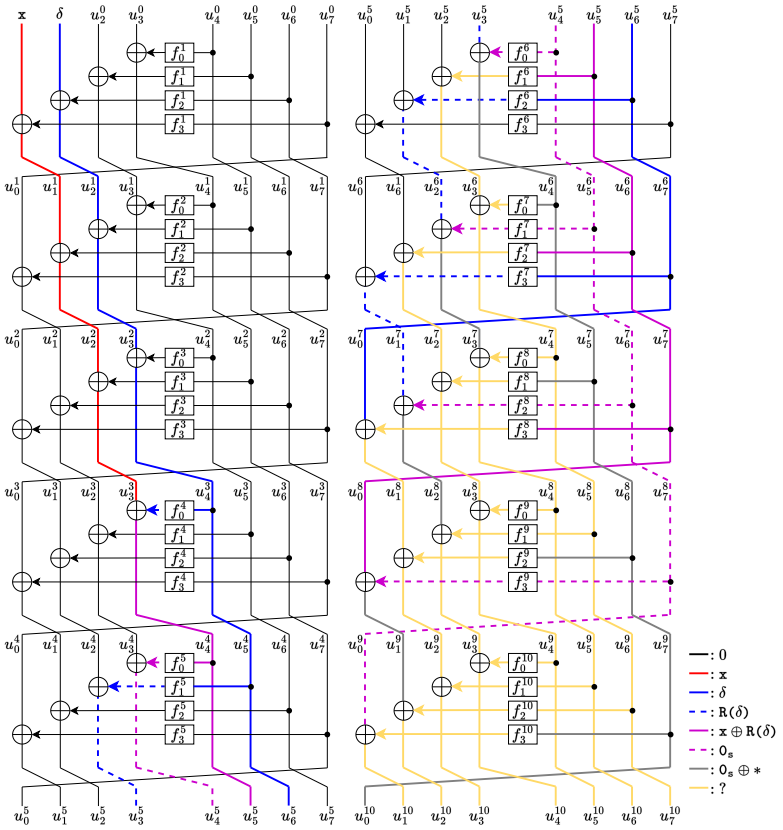

Fig. 8: The 10-round periodic distinguisher for GFS-4F.

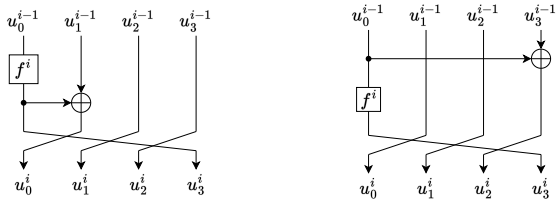

Fig. 9: The round function of Rule A (left) and Rule B (right).

– Skipjack-A: only use Rule A as the round function,
– Skipjack-B: only use Rule B as the round function.

Then, Cui et al. [8] provided the first 13-round periodic distinguishers for the two variants Skipjack-A and Skipjack-B.

In this section, we apply our model to Skipjack-A and Skipjack-B, and find the first 16-round polynomial-time periodic distinguisher of Skipjack-A/B. Appendix E depicts the periodic distinguishers. Based on the following observation, we present only the results for Skipjack-B, as the periodic distinguishers of both variants can be transformed into each other.

Observation 1 *As Rule B is basically the inverse of Rule A with minor positioning differences, any periodic distinguisher of Skipjack-B based on qCPA can be transformed into the periodic distinguisher of Skipjack-A based on qCCA.*

The first 16-round polynomial-time distinguisher of Skipjack-B with qCPA. Our model finds a 16-round periodic distinguisher of Skipjack-B:

$$(\delta, 0, 0, s \oplus \delta) \overset{16r}{\to} (0, ?, ?, ?).$$

In the input, the states in the model are $(\delta, \mathtt{0}, \mathtt{0}, \mathtt{x} \oplus \delta)$. After one round, the states are $(\mathtt{0}, \mathtt{0}, \mathtt{x}, \delta)$. Since the initial difference can be controlled by the adversary, the XOR of the two δ results in $\mathtt{0}$ based on Property 3. This distinguisher is three rounds longer than the previous quantum distinguisher and is the longest distinguisher of Skipjack-B in polynomial time. It can be transformed into a 16-round distinguisher of Skipjack-A based on qCCA. Even in the classical setting, the distinguisher still surpasses the impossible differential and zero-correlation linear distinguishers.

Applying the 16-round periodic distinguisher in the classical setting. Usually, for GFS, impossible differential and zero-correlation linear distinguishers are the most effective attacks. For Skipjack-B, in [4], Blondeau et al. showed a 16-round zero-correlation distinguisher $(u, u, 0, 0) \xrightarrow{\mathtt{16r}} (0, v, 0, 0)$ and a 16-round impossible differential distinguisher $(\delta, \delta, 0, 0) \xrightarrow{\mathtt{16r}} (\gamma, 0, 0, 0)$. The data complexity is not less than $O(2^{2n})$, where n is the size of one branch.

Our distinguisher can be achieved in polynomial time using Simon's algorithm. In the classical setting, based on the birthday problem, we prepare $2^{\frac{n}{2}}$ values of x and store the corresponding outputs in order to find a pair of x that satisfies the possible period with data complexity $O(2^{\frac{n}{2}})$. For each x, we perform encryption once for $b = 0$ and once for $b = 1$. The total complexity is about $O(2^{\frac{n}{2}})$, and the success probability exceeds 50%.

This result demonstrates that the proposed distinguisher is not only more effective in the quantum setting but also outperforms the traditional distinguishers in the classical setting. We hope this provides a good example of the application of periodic distinguishers in the cryptanalysis of GFSs.

5.3 Application on LBlock and TWINE

LBlock [29] is a variant of Feistel structures with the only difference that a left circular shift is performed on the right branch. TWINE [28] is a variant of GFS and is designed for multiple platforms.

In [16], Ito et al. proposed a 4-round periodic distinguisher for Feistel structure and can be applied to LBlock. If each S-box is treated as a branch, LBlock is similar to the Twine. The instantiated method in [7] is difficult to apply to such a large number of inputs. Considering the permutation of the S-box positions, Xiang et al. [30] propose the first 8-round distinguisher for LBlock. Using the equivalence of LBlock and TWINE [17], a 8-round periodic distinguisher for TWINE can be constructed easily.

Our model also provides a 8-round periodic distinguisher of LBlock. Furthermore, with 4 collisions, the first 10-round distinguisher is discovered for both LBlock and TWINE. Table 4 summarizes the results. The conditions of 10-round periodic distinguisher for LBlock are $\Delta u_0^1 = 0$, $\Delta u_3^1 = 0$, $\Delta u_0^2 = 0$, and $\Delta u_5^4 = 0$, and the conditions of 10-round periodic distinguisher for TWINE are $\Delta u_8^1 = 0$, $\Delta u_{14}^1 = 0$, $\Delta u_4^2 = 0$, and $\Delta u_0^4 = 0$. The other distinguishers are explained in Appendix F. We observed that the occurrence of collisions originates from a specific difference of $\Delta\alpha_b$. For the 10-round distinguisher for LBlock/TWINE, by choosing the input of the distinguisher, the probability of one collision is improved to 2^{-2}. Thus, the probability of the distinguisher can be reduced to 2^{-8}.

Table 4: The periodic distinguishers of LBlock and TWINE.

Cipher	#Rounds	Input	Output	Complexity	Reference
LBlock	3	-	-	$O(N)$	[20]
	4	-	-	$O(N)$	[16]
	8	-	-	$O(N)$	[30]
	8	$(0,0,0,0,0,0,s,0,0,0,0,0,0,0,0,\delta)$	$(?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?)$	$\boldsymbol{O(N)}$	**Our model**
	10	$(0,\delta,\delta,0,0,0,0,0,0,s,\delta,\delta,0,\delta,0,0)$	$(?,?,?,?,?,?,?,?,?,0,?,?,?,?,?,?)$	$\boldsymbol{O(2^{\frac{N}{16}})}$	**Our model**
TWINE	**8**	$(0,0,\delta,s,0,0,0,\delta,0,0,0,0,0,0,0,0)$	$(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,0)$	$\boldsymbol{O(N)}$	**Our model**
	10	$(0,0,0,0,0,0,\delta,\delta,0,s,0,\delta,0,0,\delta,\delta)$	$(?,0,?,?,?,?,?,?,?,?,?,?,?,?,?,?)$	$\boldsymbol{O(2^{\frac{N}{16}})}$	**Our model**

6 Extending Periodic Distinguishers to SPN Structure

SPN is another fundamental structure in cryptographic design. It is typically considered to have at least 4×4 state, making it increasingly difficult for the instantiated circuit search method [7] to determine the existence of a period by exhaustively searching through all possible inputs. On the other hand, the fast diffusion of the SPN structure makes it challenging for the truncated differential theory [30] to propagate through a large number of rounds. To date, no periodic distinguisher has been proposed for any SPN structure.

In this section, we successfully apply the automated model to the SPN structure, and propose the first periodic distinguishers for SKINNY and CRAFT. All the distinguishers are structural attacks that do not depend on the properties of internal components and block size N. Note that we adopt an assumption

from traditional distinguisher searches: The constraints of differences in different rounds are assumed to be independent after passing through the random round keys and S-boxes.

6.1 New Periodic Distinguishers of SKINNY

SKINNY [2] is a family of lightweight block ciphers which adopt the SPN structure. The internal states are represented as 4×4 arrays of cells with each cell being a nibble in case of 64-bit internal state or a byte in case of 128-bit internal state. The round function is described in Fig. 10.

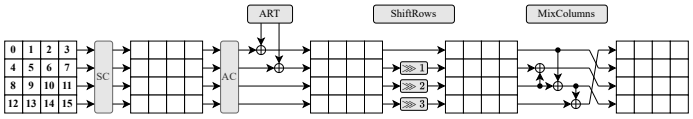

Fig. 10: The SKINNY round function: `SubCells`(SC), `AddConstants`(AC), `AddRoundTweakey`(ART), `ShiftRows`(SR), `MixColumns`(MC).

Notice that the transformation `MixColumns` maps a column $(a, b, c, d)^T$ into $(a \oplus c \oplus d, a, b \oplus c, a \oplus c)^T$, the word-based operations can be used to generate probabilistic distinguishers. For `SubCells`, let it be the operation `R` in our model. Since constants and keys do not affect the difference reviewing the trail of the periodic distinguisher, the two transformations `AddConstants` and `AddRoundTweakey` can be omitted. Our model requires that the input be `x` or δ and the period must be included in $\texttt{R}(\texttt{x} \oplus \delta)$ or $\texttt{R}(\texttt{x} \oplus \texttt{R}(\delta))$. However `SubCells` prevents the occurrence of a period before passing through `R`.

Fortunately, the values before `SubCells` in the first round are known, which allows us to construct an inverse operation $\texttt{SubCells}^{-1}$ to address this issue. Let E^r be the r-round function of SKINNY, E_1^r be the function obtained by removing the first `SubCells` operation from E^r, and E_2^r be the function that first applies the $\texttt{SubCells}^{-1}$ operation to the input, followed by E^r. We have $E_1^r \circ \texttt{SubCells} \circ \texttt{SubCells}^{-1}(x, \alpha_b, c) = E_1^r(x, \alpha_b, c)$.

Table 5 shows the results of periodic distinguishers for SKINNY. We use superscripts to indicate the positions in the cipher. For example, for the 7-round distinguisher of SKINNY, $\alpha_b = (\alpha_b^5, \alpha_b^6, \alpha_b^8, \alpha_b^9, \alpha_b^{15})$ and the difference is $\delta = (\delta^5, \delta^6, \delta^8, \delta^9, \delta^{15})$. Unlike previous distinguishers, the difference δ in this distinguisher needs to first satisfy constraints in the input difference: $\delta^5 = \delta^8 = \delta^{15}$ and $\delta^6 = \delta^9$. Also, the 9-round distinguisher must first satisfy constraints in the input difference: $\delta^1 = \delta^4 = \delta^{11}$, and $\delta^2 = \delta^8$. These constraints are used to generate zero differences after `MixColumns`. We can control the initial differences, ensuring that the probability does not increase.

Let the j-th branch before and after the i-th round of `MixColumns` be denoted as SR_j^i and MC_j^i, respectively. The 9-round distinguisher requires the following constraints with complexity $O(2^{\frac{3N}{16}})$. In round 2, we have the constraints

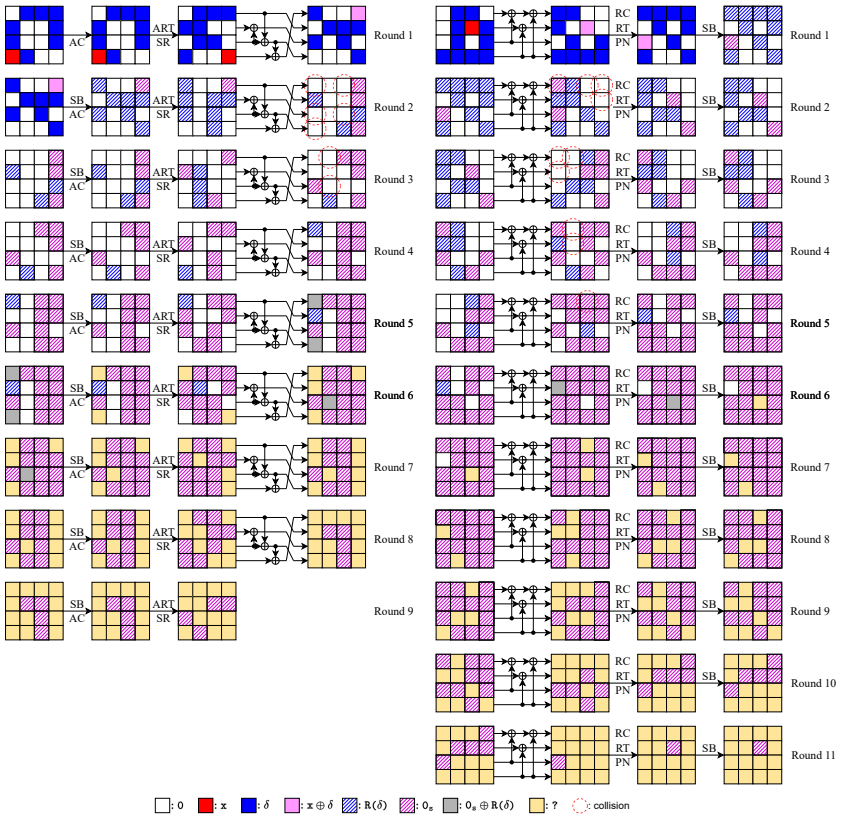

Fig. 11: The 9-round periodic distinguisher for SKINNY (left) and the 12-round periodic distinguisher for CRAFT (right).

$\Delta SR_0^2 = \Delta SR_8^2$, $\Delta SR_4^2 = \Delta SR_8^2$, $\Delta SR_{10}^2 = \Delta SR_{14}^2$, and $\Delta SR_6^2 = \Delta SR_{10}^2$, which affect ΔMC_0^2, ΔMC_{12}^2, ΔMC_8^2, ΔMC_2^2, and ΔMC_{10}^2. In round 3, we have the constraints $\Delta SR_9^3 = \Delta SR_{13}^3$, and $\Delta SR_5^3 = \Delta SR_9^3$, which affect ΔMC_1^3 and ΔMC_9^3. Fig. 11 shows the 9-round periodic distinguishers. Appendix G shows other periodic distinguishers.

Based on our assumption, the constraints of different rounds are considered independent after `AddRoundTweakey` and `SubCells` operations, which are used to calculate the theoretical probability. Experiments have verified that this probability is reasonable.

We apply the distinguishers in the classical setting and compare with the best current differential-linear distinguishers in [13]. If the bias is ϵ, we set the complexity to $(\frac{1}{\epsilon})^2$ for a simple comparison. The complexity of 7-/8-/9-round differential-linear distinguisher for SKINNY-64 is $2^{10}/2^{17.74}/2^{29.88}$. The complexity of our distinguisher is $2^3/2^7/2^{27}$, where we add 2^3 to search for a period.

Table 5: The periodic distinguishers of SKINNY and CRAFT.

Cipher	#Rounds	Input	Output	Complexity
SKINNY	**7**	$(0,0,0,0,0,\delta^5,\delta^6,0,\delta^8,\delta^9,0,0,s,0,0,\delta^{15})$	$(?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,?)$	$\boldsymbol{O(N)}$
	9	$(0,\delta^1,\delta^2,\delta^3,\delta^4,0,0,\delta^7,\delta^8,0,0,\delta^{11},s,\delta^{13},0,0)$	$(?,?,?,?,?,?,0,0,0,?,?,?,?,0,?,?)$	$\boldsymbol{O(2^{\frac{3N}{16}})}$
CRAFT	**7+1**	$(s,0,0,0,0,0,0,0,\delta^8,0,0,0,0,0,0,0)$	$(?,?,?,?,0,?,?,?,?,?,?,?,?,?,?,?)$	$\boldsymbol{O(N)}$
	11+1	$(0,\delta^1,\delta^2,0,0,\delta^5,s,\delta^7,0,\delta^9,0,\delta^{11},\delta^{12},\delta^{13},\delta^{14},\delta^{15})$	$(?,?,?,?,?,?,?,0,?,?,?,?,?,?,?,?)$	$\boldsymbol{O(2^{\frac{5N}{16}})}$

It shows that our distinguishers achieve better performance for the same number of rounds.

6.2 New Periodic Distinguishers of CRAFT

CRAFT [3] is a lightweight tweakable block cipher, the internal states are represented as 4×4 arrays of cells with each cell being a nibble. The round function is described in Fig. 12. For stronger diffusion, `PermuteNibbles` uses different methods, such as `RShift`, `Shuffle`, and `LShift`. `S-box` is not included in the last round.

`MixColumns` maps a column $(a,b,c,d)^T$ into $(a \oplus c \oplus d, b \oplus d, c, d)^T$. Then, `AddRoundConstants` and `AddTweakey` can be omitted, and we let `S-box` be the operation `R` in our model. However, if the difference with unknown values passes `R`, the periodicity will vanish. Since our model requires that the input is `x` or δ and the period is included in $\mathtt{R}(\mathtt{x} \oplus \delta)$ or $\mathtt{R}(\mathtt{x} \oplus \mathtt{R}(\delta))$, `x` must occur in the first two lines. The periodic point $\mathtt{0_s}$ always occurs in the first round.

As a result, we find a 8-round distinguisher with probability 1 and a 12-round probabilistic periodic distinguisher. The latter is shown in Fig. 11. The 8-round distinguisher is also the first polynomial-time distinguisher of CRAFT. The previous distinguisher in the same round requires the probability 2^{-4} in [22]. Since the last round has no S-box, the difference in the head of the last round is known. Therefore, we can omit the final round. For example, If the output of the 12-th round is known, the difference of the output of the 11-th round will also be known. Table 5 shows the 7-/11-round distinguishers.

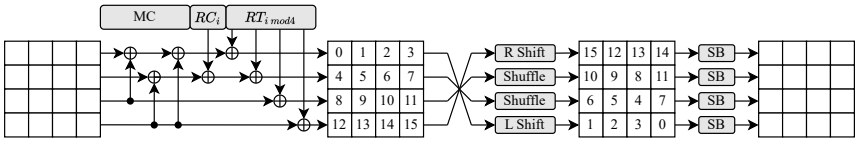

Fig. 12: The CRAFT round function, including `MixColumns`(MC), `AddRoundConstants`(RC), `AddTweakey`(RT), `PermuteNibbles`(PN), and `S-box`(SB).

Let the j-th branch before and after the i-th round of `MixColumns` be denoted as S_j^{i-1} and MC_j^i. In the input of the 12-round periodic distinguisher, we need to set $\delta^5 = \delta^{13}$, $\delta^2 = \delta^{14}$, and $\delta^7 = \delta^{11} = \delta^{15}$, but the probability does not increase with these constraints. The 12-round distinguisher requires the following

constraints with complexity $O(2^{\frac{5N}{16}})$. In round 2, we have the constraints $\Delta S_0^1 = \Delta S_{12}^1$, $\Delta S_2^1 = \Delta S_{10}^1$, $\Delta S_3^1 = \Delta S_{15}^1$, and $\Delta S_7^1 = \Delta S_{15}^1$, which affect ΔMC_0^2, ΔMC_2^2, ΔMC_3^2, and ΔMC_7^2. In round 3, we have $\Delta S_0^2 = \Delta S_{12}^2$, $\Delta S_4^2 = \Delta S_{12}^2$, and $\Delta S_1^2 = \Delta S_9^2$, which affect ΔMC_0^3, ΔMC_4^3, and ΔMC_1^3. In round 4, we have $\Delta S_1^3 = \Delta S_{13}^3$ and $\Delta S_5^3 = \Delta S_{13}^3$, which affect ΔMC_1^4 and ΔMC_5^4. In round 5, we have $\Delta S_2^4 = \Delta S_{10}^4$, which affects ΔMC_2^5. Appendix H shows other distinguishers. Note that we also adopt the assumption that the constraints of differences in different rounds are independent.

7 Further Discussion on Modelling MDS Matrices

We have proven that the model is highly applicable to various structures and ciphers. However, for ciphers that include MDS matrices, there is currently no effective method to describe periodic attacks. In this section, we take Piccolo [25] as an example to illustrate how our model can be applied to this type of cipher.

Piccolo is a lightweight block cipher using the Feistel-SP structure, proposed by Shibutani et al. for extremely constrained devices. The round function for Piccolo is shown in Fig. 13, including `S-box` and `MixColumns`. For Feistel-SP structure, there exists a 3-round periodic distinguisher based on qCPA and a 4-round distinguisher based on qCCA [16]. The more complex permutation used in Piccolo makes it difficult for these attacks to be directly applied. It remains unknown whether a better distinguisher can be found by decomposing the round function.

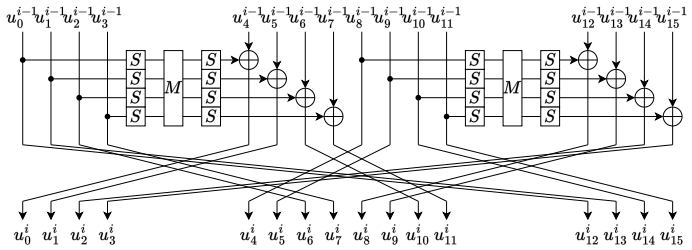

Fig. 13: The round function of Piccolo.

Adapting the model for Piccolo. In Piccolo, each branch is 16 bits. To apply the model to Piccolo, we split each 16-bit branch into four 4-bit branches (see Fig. 13) and we made the following modifications:

1. We propose an approximate strategy for modeling MDS matrices. As long as the difference of $\mathtt{x}$ or δ changes, we treat it as $\mathtt{R}(\mathtt{x})$ or $\mathtt{R}(\delta)$. This constraint is necessary because once the difference changes, the cancellation property,

such as $\delta \oplus \delta = 0$, is lost. For example, for a column of inputs (x_0, x_1, x_2, x_3), the output y_0 can be expressed as $y_0 = \mathtt{R}(x_0) \oplus \mathtt{R}(x_1) \oplus x_2 \oplus x_3$.

2. We relax the initial constraints. We hope all the state $\mathtt{x}$ come from an unsplit 16-bit branch of Piccolo. That is, at the head, x appears at most four times. However, in the actual model, we find that even without restricting the number of occurrences of $\mathtt{x}$, the model still produces the same result. Therefore, we remove this restriction on the number of x in the head of Piccolo. This modifies the initial constraints of the previous model.
3. We relax the rule of $\mathtt{R}$. In Equation (7), only one branch with state $\mathtt{x} \oplus \delta$, $\mathtt{x} \oplus \mathtt{R}(\delta)$, or $\mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta$ can generate the state $\mathtt{0_s}$, and the others are set to $\perp$. Because of the modification of $\mathtt{x}$, we limit the rule $\mathtt{x} \oplus \delta$, $\mathtt{x} \oplus \mathtt{R}(\delta)$, $\mathtt{x} \oplus \mathtt{R}(\delta) \oplus \delta \overset{\mathtt{R}}{\rightarrow} \mathtt{0_s}$ to a maximum of 4 times. This modification is reasonable. For each x, the implicit period is the same, and the final distinguisher has at most four implicit periods from four different $\mathtt{x}$. In the previous proof, the output after XORing $\mathtt{0_s}$ with different periods was $\perp$. However, if the $\mathtt{0_s}$ are generated by different branches with x, it is also possible for the output to be $\mathtt{0_s}$.

Relaxing the model's constraints invalidates our original proof of its correctness. To address this, we introduce an additional testing phase to verify whether the implicit period remains consistent for each x and to search for the actual period of each result returned by the model. Finally, we successfully identify the first 4-round polynomial-time periodic distinguisher:

$$(0,0,0,0,0,0,0,0,\delta,\delta,0,0,s,s,s,s) -> (?,?,?,?,0,0,0,0,?,?,?,?,?,?,?,?).$$

In the first round, we set $(u_8^0, u_9^0) = (\alpha_b, \beta_b)$, where $\alpha_0 \neq \alpha_1, \beta_0 \neq \beta_1$. The input of the periodic function is $x = x_{12}||x_{13}||x_{14}||x_{15} \in \{0,1\}^{4\times 4}$. Then, let

$$E_b := E^4(u_0^0, u_1^0, u_2^0, u_3^0, u_4^0, u_5^0, u_6^0, u_7^0, \alpha_b, \beta_b, u_{10}^0, u_{11}^0, x_{12}, x_{13}, x_{14}, x_{15}),$$

where $u_0^0, u_1^0, u_2^0, u_3^0, u_4^0, u_5^0, u_6^0, u_7^0, u_{10}^0, u_{11}^0$ are random constants, $||$ represents bit concatenation, and $b \in \{0,1\}$. The modified periodic function is:

$$\begin{aligned} g : \{0,1\}^{16} &\rightarrow \{0,1\}^{16}, \\ x &\mapsto \Delta(u_4^4||u_5^4||u_6^4||u_7^4), \\ g(x) &= (E_0|_{u_4^4})||(E_0|_{u_5^4})||(E_0|_{u_6^4})||(E_0|_{u_7^4}) \oplus (E_1|_{u_4^4})||(E_1|_{u_5^4})||(E_1|_{u_6^4})||(E_1|_{u_7^4}). \end{aligned}$$

Actually, we recombine the outputs of these four branches into a single 16-bit value. For the period value $s \in \{0,1\}^{16}$, $g(x \oplus s) = g(x)$ for any $x \in \{0,1\}^{16}$. Our experiment also verifies the periodic distinguisher. Appendix I provides a detailed explanation.

Discussion on modelling MDS matrices. Using our approximate strategy, we can also automate the search for the number of rounds in periodic distinguishers of ciphers with MDS matrices. There are two open problems. First, we aim to develop a more precise model for MDS matrices instead of relying on the

approximate strategy. Second, when the number of initial x increases, we want to explore whether there is a more precise model that ensures the correctness of the output distinguisher without requiring an additional testing phase. As the first proposed symbolic model, we leave these two problems as directions for future work.

8 Conclusion

This paper presents a significant advancement in quantum cryptanalysis of symmetric-key schemes by introducing an automated approach for discovering periodic distinguishers. The proposed methodology for Simon's algorithm offers a more refined and systematic framework compared to existing techniques. A key contribution is the development of probabilistic periodic distinguishers using the Grover-meet-Simon algorithm, which enhances the identification of effective periodic distinguishers through optimized strategies for selecting initial differences. This theoretical framework significantly broadens the scope of detectable periodic distinguishers. The SMT-based model demonstrates extensive applicability, achieved through significant advancements in the analysis of both generalized Feistel structures and substitution-permutation network (SPN) ciphers. This framework enables the discovery of optimized periodic distinguishers for numerous cryptographic constructions, including generalized Feistel structures (e.g., GFS-2F/4F, Skipjack-type structures, CAST-256, LBlock, and TWINE) as well as SPN-based designs (e.g., SKINNY and CRAFT). Our findings reveal that periodic distinguishers, as identified through our automated model, demonstrate effects that differ from traditional methods. Notably, these distinguishers not only prove effective in quantum cryptanalysis but also exhibit new insight in classical cryptanalysis settings. We anticipate that future research will further explore these distinguishers and potentially extend their application to a wider range of cryptographic primitives by providing improved periodic analysis methodologies. The results from our automated model suggest that periodic analysis may establish a novel cryptanalytic technology. However, some challenges remain, particularly in developing more precise analytical models for ciphers with MDS matrices, which represents a critical direction for future investigation.

References

1. Adams, C., Gilchrist, J.: The CAST-256 encryption algorithm. RFC **2612**, 1–19 (1999). https://doi.org/10.17487/RFC2612
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5

3. Beierle, C., Leander, G., Moradi, A., Rasoolzadeh, S.: CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. IACR Trans. Symmetric Cryptol. **2019**(1), 5–45 (2019). https://doi.org/10.13154/TOSC.V2019.I1.5-45
4. Blondeau, C., Bogdanov, A., Wang, M.: On the (in)equivalence of impossible differential and zero-correlation distinguishers for feistel- and skipjack-type ciphers. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8479, pp. 271–288. Springer (2014). https://doi.org/10.1007/978-3-319-07536-5_17
5. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline Simon's algorithm. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 552–583. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_20
6. Bonnetain, X., Leurent, G., Naya-Plasencia, M., Schrottenloher, A.: Quantum linearization attacks. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 422–452. Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_15
7. Canale, F., Leander, G., Stennes, L.: Simon's algorithm and symmetric crypto: Generalizations and automatized applications. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13509, pp. 779–808. Springer (2022). https://doi.org/10.1007/978-3-031-15982-4_26
8. Cui, J., Guo, J., Ding, S.: Applications of Simon's algorithm in quantum attacks on feistel variants. Quantum Inf. Process. **20**(3), 117 (2021). https://doi.org/10.1007/S11128-021-03027-X
9. Daemen, J., Rijmen, V.: The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition. Information Security and Cryptography, Springer (2020). https://doi.org/10.1007/978-3-662-60769-5
10. Dong, X., Dong, B., Wang, X.: Quantum attacks on some feistel block ciphers. Des. Codes Cryptogr. **88**(6), 1179–1203 (2020). https://doi.org/10.1007/S10623-020-00741-Y
11. Dong, X., Li, Z., Wang, X.: Quantum cryptanalysis on some generalized feistel schemes. Sci. China Inf. Sci. **62**(2), 22501:1–22501:12 (2019). https://doi.org/10.1007/S11432-017-9436-7
12. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996)
13. Hadipour, H., Derbez, P., Eichlseder, M.: Revisiting differential-linear attacks via a boomerang perspective with application to aes, ascon, clefia, skinny, present, knot, twine, warp, lblock, simeck, and SERPENT. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV.

Lecture Notes in Computer Science, vol. 14923, pp. 38–72. Springer (2024). https://doi.org/10.1007/978-3-031-68385-5_2

14. Hodzic, S., Knudsen, L.R., Kidmose, A.B.: On quantum distinguishers for type-3 generalized feistel network based on separability. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12100, pp. 461–480. Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_25
15. Hosoyamada, A., Sasaki, Y.: Quantum demiric-selçuk meet-in-the-middle attacks: Applications to 6-round generic feistel constructions. In: Catalano, D., Prisco, R.D. (eds.) Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11035, pp. 386–403. Springer (2018). https://doi.org/10.1007/978-3-319-98113-0_21
16. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Y., Iwata, T.: Quantum chosen-ciphertext attacks against feistel ciphers. In: Matsui, M. (ed.) Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11405, pp. 391–411. Springer (2019). https://doi.org/10.1007/978-3-030-12612-4_20
17. Ju, Z., Liu, P., Xue, W., Mu, D., Lai, X.: On the equivalence of lblock and TWINE in structure. In: 10th International Conference on Communications and Networking in China, ChinaCom 2015, Shanghai, China, August 15-17, 2015. pp. 289–294. IEEE Computer Society (2015). https://doi.org/10.1109/CHINACOM.2015.7497953
18. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 207–237. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_8
19. Knudsen, L.R., Wagner, D.A.: On the structure of skipjack. Discret. Appl. Math. **111**(1-2), 103–116 (2001). https://doi.org/10.1016/S0166-218X(00)00347-4
20. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings. pp. 2682–2685. IEEE (2010). https://doi.org/10.1109/ISIT.2010.5513654
21. Leander, G., May, A.: Grover meets Simon - quantumly attacking the fx-construction. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10625, pp. 161–178. Springer (2017). https://doi.org/10.1007/978-3-319-70697-9_6
22. Moghaddam, A.E., Ahmadian, Z.: New automatic search method for truncated-differential characteristics application to midori, SKINNY and CRAFT. Comput. J. **63**(12), 1813–1825 (2020). https://doi.org/10.1093/COMJNL/BXAA004
23. NIST: Skipjack and kea algorithm specifications (1998), https://csrc.nist.gov/CSRC/media//Projects/Cryptographic-Algorithm-Validation-Program/documents/skipjack/skipjack.pdf

24. Nyberg, K.: Generalized feistel networks. In: Kim, K., Matsumoto, T. (eds.) Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1163, pp. 91–104. Springer (1996). https://doi.org/10.1007/BFB0034838
25. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 342–357. Springer (2011). https://doi.org/10.1007/978-3-642-23951-9_23
26. Simon, D.R.: On the power of quantum computation. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 116–123. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365701
27. Sun, H.W., Cai, B.B., Qin, S.J., Wen, Q.Y., Gao, F.: Quantum attacks on type-1 generalized feistel schemes. Advanced Quantum Technologies **6**(10), 2300155 (2023)
28. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7707, pp. 339–354. Springer (2012). https://doi.org/10.1007/978-3-642-35999-6_22
29. Wu, W., Zhang, L.: Lblock: A lightweight block cipher. In: López, J., Tsudik, G. (eds.) Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6715, pp. 327–344 (2011). https://doi.org/10.1007/978-3-642-21554-4_19
30. Xiang, Z., Wang, X., Yu, B., Sun, B., Zhang, S., Zeng, X., Shen, X., Li, N.: Links between quantum distinguishers based on Simon's algorithm and truncated differentials. IACR Trans. Symmetric Cryptol. **2024**(2), 296–321 (2024). https://doi.org/10.46586/TOSC.V2024.I2.296-321
31. Xu, Y., Du, X., Jia, M., Wang, X., Zou, J.: Quantum attacks on generalized feistel networks based on the strong-weak separability. Quantum Inf. Process. **22**(10), 375 (2023). https://doi.org/10.1007/S11128-023-04135-6

Appendix

A SMT Problem

In recent years, the application of automated search tools in cryptography has become more and more extensive. The SAT problem belongs to the deterministic problem, and it is also the first problem to be proved to be NP-complete. To solve it, boolean expressions are usually encoded in Conjunctive Normal Form (CNF) as the inputs of a SAT solver.

Extending SAT to include the theory of modulus (satisfiability modulo theories, abbreviated as SMT) enriches the forms of CNF expressions, which can include linear constraints, arrays, and more. Compared to methods based on SAT problems, those based on SMT are more flexible and applicable to a broader range of scenarios, making them particularly suitable for use in the field of cryptography.

This paper primarily uses the solver of the SMT problem, STP, to automatically solve our new proposed models. CVC formats are among the commonly used file-based input languages in STP. We list some CVC language references and two examples as follows.

Table 6: Usage of the STP solver.

Name	Symbol	Example
Concatenation	@	$t_1@t_2@\ldots@t_n$
Extraction	$i:j$	$x[31:26]$
Bitwise XOR	BVXOR	$\mathrm{BVXOR}\,(t_1,t_2)$
Bitvector AND	BVPLUS	$\mathrm{BVPLUS}\,(n,t_1,t_2,\ldots,t_n)$
Less Than Or Equal To	BVLE	$\mathrm{BVLE}\,(t_1,t_2)$
Greater Than or Equal To	BVGE	$\mathrm{BVGE}\,(t_1,t_2)$
Not Equal to	NOT	$\mathrm{NOT}(t_1=t_2)$
If Then	=>	$t_1 => t_2$

B Extending the Tail of the Periodic Distinguisher

We observe that for certain structures, a linear combination of multiple non-periodic branches in the tail can generate a value that satisfies the periodicity. Therefore, in this section, we propose a method that combines our automation model to simultaneously search for distinguishers and tail extensions. Since only special structures can be further extended, we have placed this part of the content in the appendix. Skipping this part will not affect the reader's understanding of the model.

Here, we discuss how to further extend the number of rounds after obtaining the tail u_i^r of a constructible periodic function.

Definition 4 (Linear trail with known values). *Set a r'-round linear trail of E from tail to head. Let Γ_i^j represents the mask of u_i^j in the linear trail $(r \le j \le r + r')$ and $(\Gamma_0^j, \ldots, \Gamma_{n-1}^j) \neq (0, \ldots, 0)$. Towards to the round function* `R`*, the XOR operation* `XOR`*, and the branching operation* `SPLIT`*, the trail satisfies the following rules:*

- $0 \xrightarrow{\texttt{R}} 0, \quad 1 \xrightarrow{\texttt{R}} 0,$
- $(u_0, u_1) \xrightarrow{\texttt{XOR}} u_2$, *where* $u_2 = u_0 \,\&\, u_1$,
- $u_0 \xrightarrow{\texttt{SPLIT}} (u_1, u_2)$, *where* $u_1 = u_0,\ u_2 = u_0$.

Lemma 1. *Assume the r'-round linear trail with known values satisfies Definition 4. Let there be t $(1 \le t \le n-1)$ masks of 1 at the head and t' $(1 \le t' \le n-1)$ masks of 1 at the tail. If the values corresponding to these t' masks are known, then the values at the head of the linear trail with masks of 1 are also known.*

- *For* `R`*, the output values are always indeterminate due to the unknown key.*
- *For* `XOR`*, $(u_0, u_1) \xrightarrow{\texttt{XOR}} u_2$ requires values of u_0, u_1 to calculate u_2.*
- *For* `SPLIT`*, $u_0 \xrightarrow{\texttt{SPLIT}} (u_1, u_2)$ indicates that if u_0 is known, then the other two values are also known.*

Fig. 14 shows a linear path with known values. In these two structures, a is unknown ("unknown" means a is a combination of all the inputs and cannot be separated) and b satisfies the separability property. We set the masks $(\Gamma_a, \Gamma_b) = (0, 1)$. In the left, the masks $\Gamma_c = \Gamma_f = 1$ and $\Gamma_d = \Gamma_e = 1$, and $\Gamma_f = 1$. We have a 1-round linear path with known values from (Γ_e, Γ_f) to (Γ_a, Γ_b) with $(1, 1) \to (0, 1)$ and $b = e \oplus f$. In the right, Γ_c is the output of F and must be 0. Then, $\Gamma_d = 0$. We cannot calculate b.

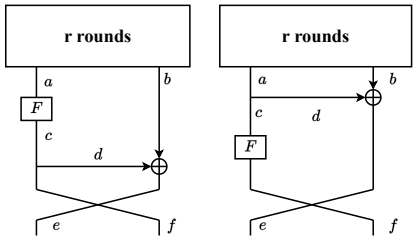

Fig. 14: An example illustrating a linear path.

As long as an r-round trail exists, the distinguisher can be extended by r additional rounds.

Based on the above discussion, if there exists an output u_i^r of E^r satisfying the (probability) separability property and a r'-round linear trail with known values connected by u_i^r, we can construct an $(r+r')$-round periodic distinguisher.

Modelling the propagation rules of linear trail. Assume that there are an r-round differential trail and an r'-round linear trail. We set masks $(\Gamma_0^r, \Gamma_1^r, \ldots, \Gamma_{n-1}^r)$ for the tail $(u_0^r, u_1^r, \ldots, u_{n-1}^r)$ of the differential trail. We define a new Boolean variable $mask_i$ for each branch. If u_0^r does not satisfy the separability property, then set $mask_i = 0$. Then, let $\Sigma(mask_i) = 1$, as at least one branch satisfying the separability property needs to be calculated The other rules are consistent with Lemma 1.

C Validations of the Model

To validate the correctness of our model, we successfully replicated all the currently optimal periodic distinguishers, Type-1/2/3 GFS, GFS-2F/4F, Skipjack-B, LBlock, TWINE, Piccolo, SKINNY, and CRAFT, and constructed small circuit structures. We apply random keys and constants for each structure to search for periods and can consistently find the actual period by constructing a periodic function. Table 7 shows the results.

Table 7: The experiments of periodic distinguishers.

Structure	#Rounds	Theoretical probability	Experimental probability
Type-I GFS	9	1	1
Type-II GFS	5	1	1
Type-III GFS	5	1	1
GFS-2F	6	1	1
GFS-4F	10	1	1
Skipjack-B	16	1	1
LBlock	8	1	1
LBlock	10	2^{-8}	2^{-8}
Twine	8	1	1
Twine	10	2^{-8}	2^{-8}
Piccolo	4	1	1
SKINNY	7	1	1
SKINNY	8	2^{-4}	$2^{-3.5}$
SKINNY	9	2^{-24}	2^{-23}
CRAFT	7+1	1	1
CRAFT	8+1	2^{-4}	$2^{-3.5}$
CRAFT	9+1	2^{-12}	2^{-11}
CRAFT	10+1	2^{-24}	2^{-22}

D More Explanations for GFS-4F

For the 10-round distinguisher, we have

$$\begin{aligned}u_0^{10} = & u_6^0 \oplus f_3^3(u_5^0) \oplus f_2^4(u_3^0 \oplus f_0^1(u_4^0)) \\ & \oplus f_1^5(u_1^0 \oplus f_2^1(u_6^0) \oplus f_1^2(u_4^0) \oplus f_0^3(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))) \\ & \oplus f_0^6(u_7^0 \oplus f_3^2(u_6^0) \oplus f_2^3(u_4^0) \oplus f_1^4(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0))) \\ & \oplus f_0^5(u_0^0 \oplus f_3^1(u_7^0) \oplus f_2^2(u_5^0) \oplus f_1^3(u_3^0 \oplus f_0^1(u_4^0)) \\ & \oplus f_0^4(u_1^0 \oplus f_2^1(u_6^0) \oplus f_1^2(u_4^0) \oplus f_0^3(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))))).\end{aligned}$$

The periodic function is constructed by

$$\begin{aligned}g(x) &= E^{10}(x, \alpha_0, u_2^0, u_3^0, u_4^0, u_5^0, u_6^0, u_7^0)|_{u_0^{10}} \oplus E^{10}(x, \alpha_1, u_2^0, u_3^0, u_4^0, u_5^0, u_6^0, u_7^0)|_{u_0^{10}} \\ &= \Delta f_1^5(\alpha_b \oplus C_1) \oplus \Delta f_0^6(f_0^5(x \oplus f_0^4(\alpha_b \oplus C_2) \oplus C_3) \oplus C_4),\end{aligned}$$

where $u_2^0, u_3^0, u_4^0, u_5^0, u_6^0, u_7^0$ are constants, $C_1 = f_2^1(u_6^0) \oplus f_1^2(u_4^0) \oplus f_0^3(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))$, $C_2 = f_2^1(u_6^0) \oplus f_1^2(u_4^0) \oplus f_0^3(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))$, $C_3 = f_3^1(u_7^0) \oplus f_2^2(u_5^0) \oplus f_1^3(u_3^0 \oplus f_0^1(u_4^0))$, $C_4 = u_7^0 \oplus f_3^2(u_6^0) \oplus f_2^3(u_4^0) \oplus f_1^4(u_2^0 \oplus f_1^1(u_5^0) \oplus f_0^2(u_3^0 \oplus f_0^1(u_4^0)))$ are also constants.

E More Explanations for Skipjack and Its Variant

In this section, we show the periodic distinguishers for Skipjack and its variant Skipjack-B.

- 15-round periodic distinguisher for Skipjack is shown in Fig. 15, where u_1^{14} can be calculated by $u_0^{15} \oplus u_3^{15}$.
- 16-round periodic distinguisher for Skipjack-B is shown in Fig. 16.

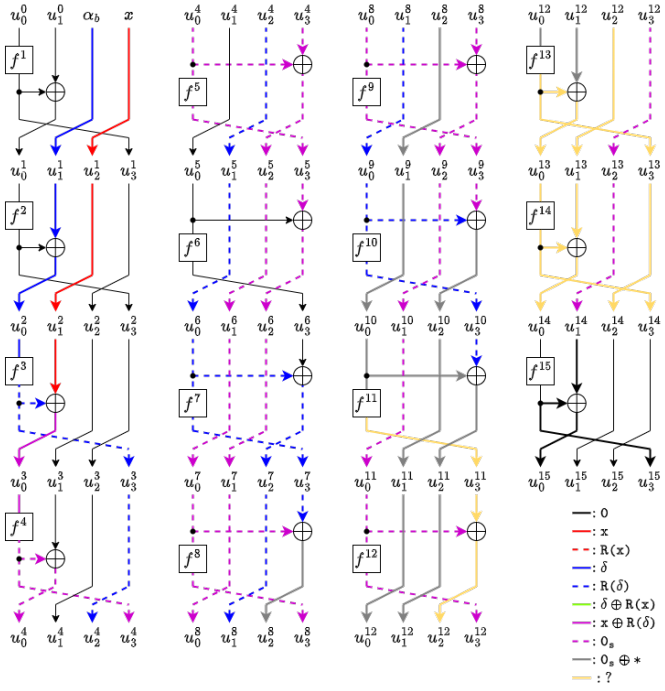

Fig. 15: 15-round distinguisher of Skipjack.

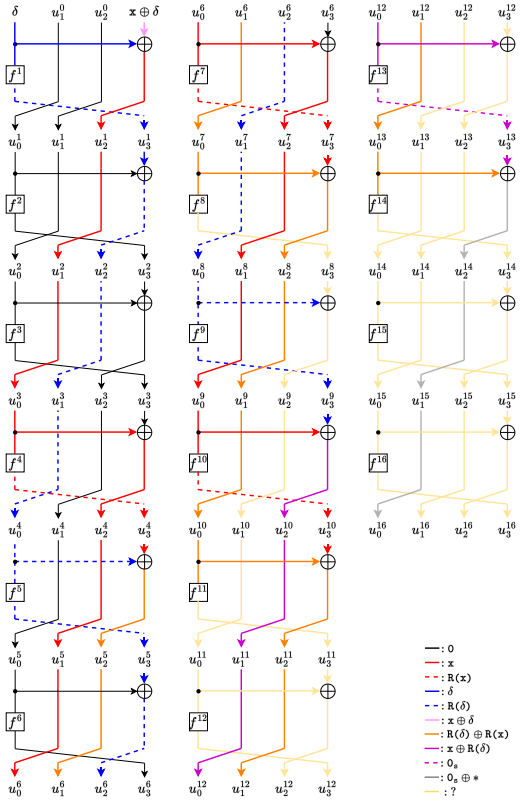

Fig. 16: 16-round distinguisher of Skipjack-B.

F 8/10-round Periodic Distinguishers for LBlock/TWINE

In this section, we present the periodic distinguishers for LBlock/TWINE.

- 8-round periodic distinguisher for LBlock is shown in Fig. 17.
- 8-round periodic distinguisher for TWINE is shown in Fig. 18.
- 10-round periodic distinguisher for LBlock is shown in Fig. 19, requiring 4 collisions.
- 10-round periodic distinguisher for TWINE is shown in Fig. 20, requiring 4 collisions.

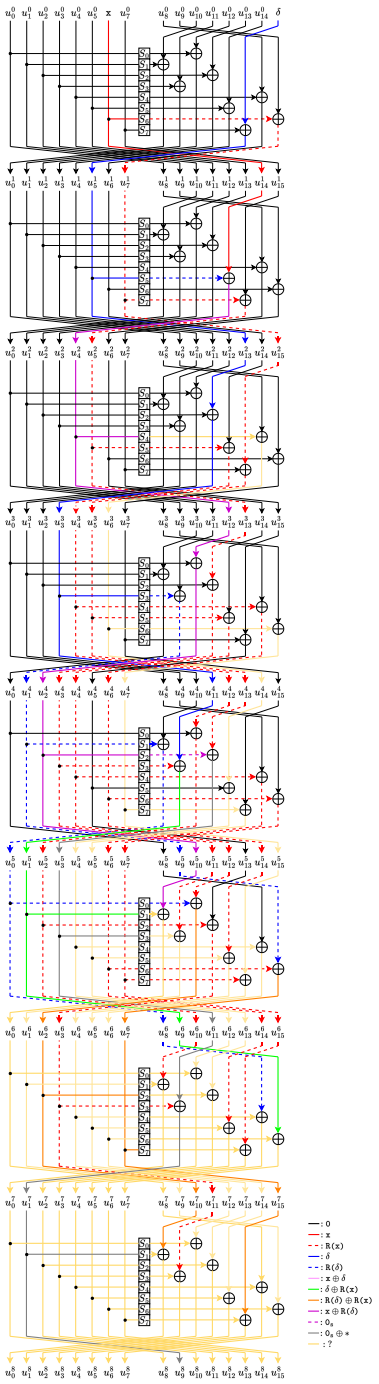

Fig. 17: 8-round distinguisher of LBlock.

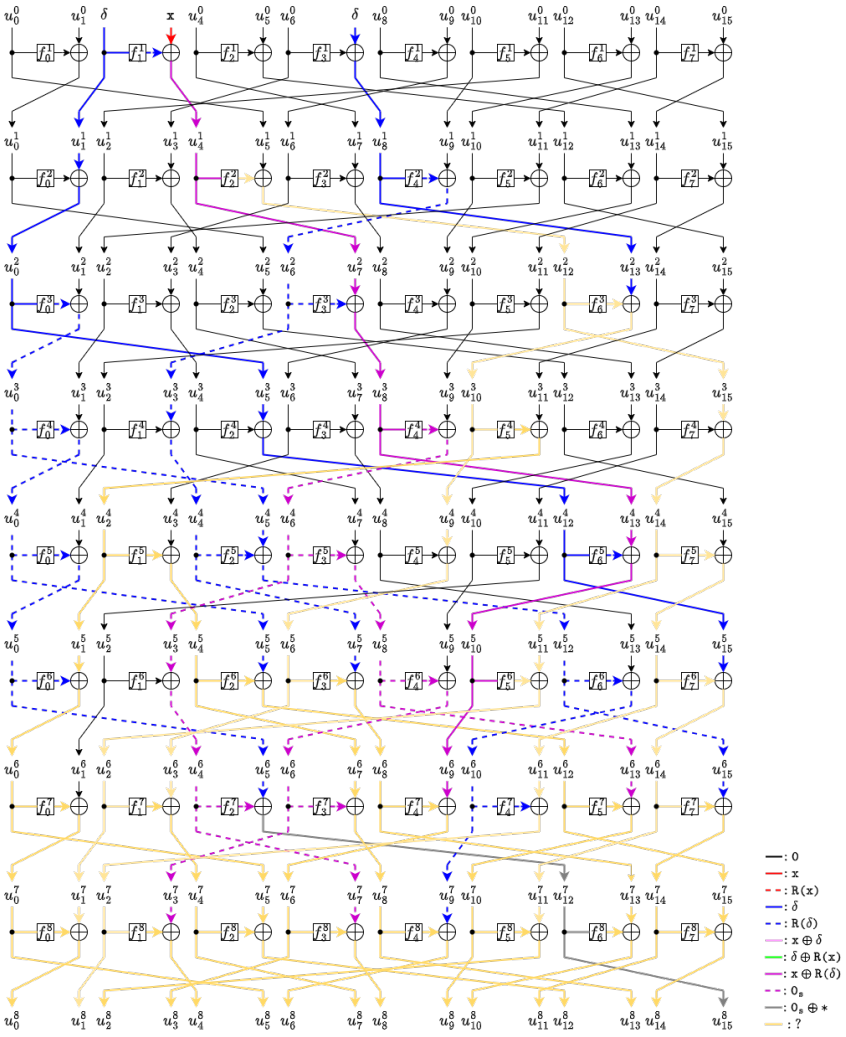

Fig. 18: 8-round distinguisher of TWINE.

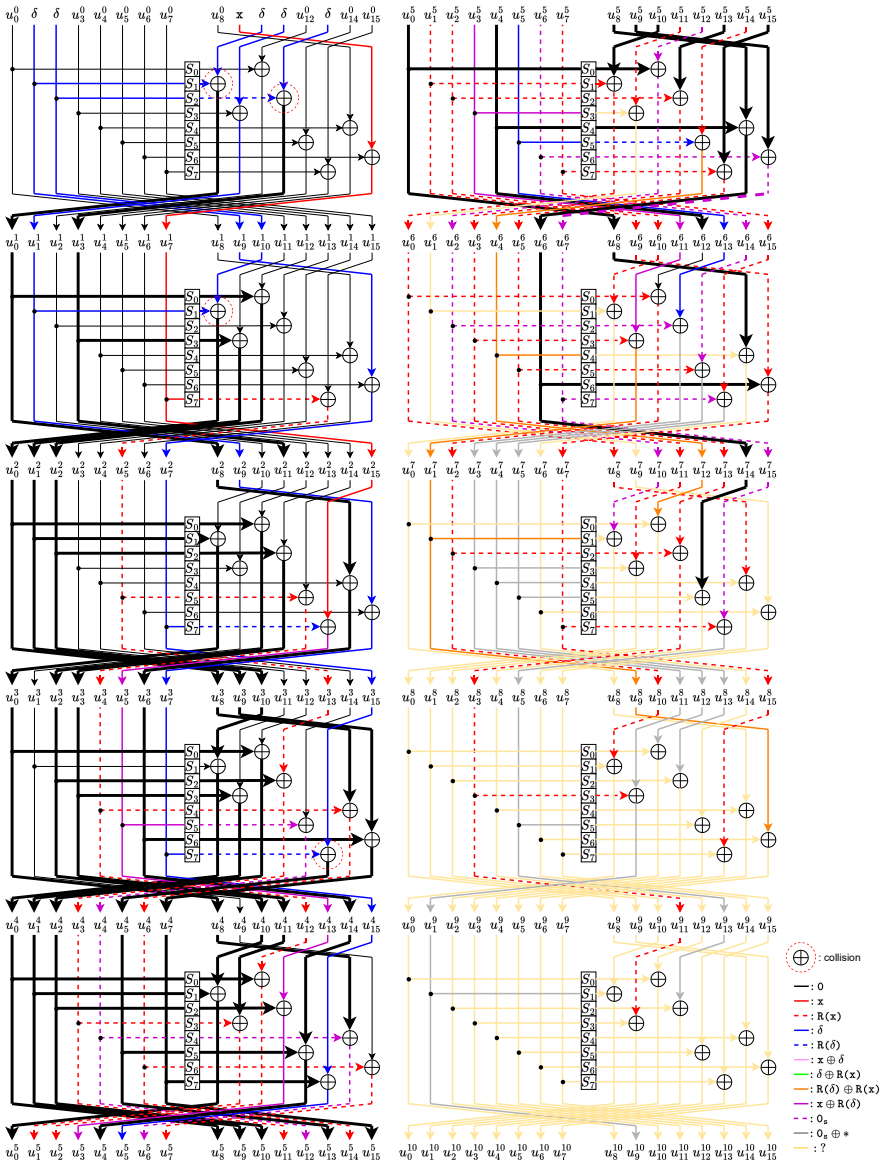

Fig. 19: 10-round distinguisher of LBlock.

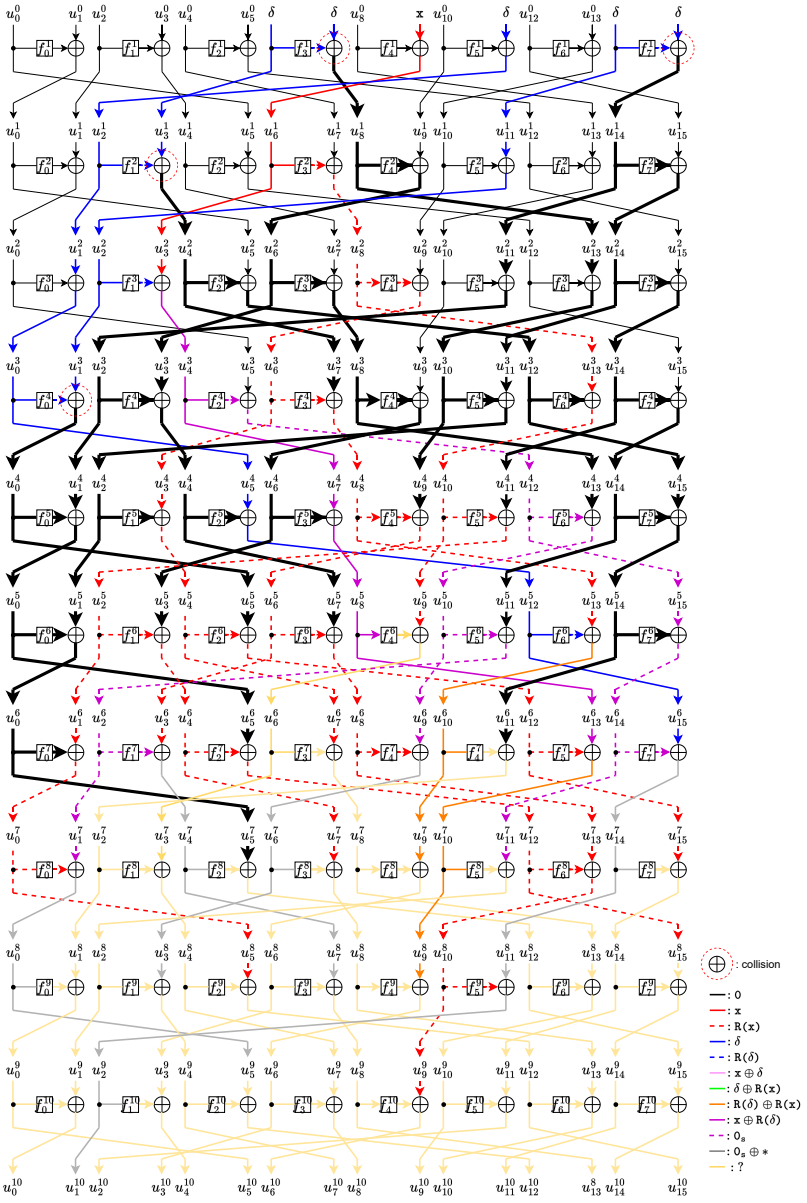

Fig. 20: 10-round distinguisher of TWINE.

G Periodic Distinguishers for SKINNY

In this section, we present the periodic distinguishers for SKINNY. The advantages of our distinguishers lie in its independence from specific components, while maintaining low complexity.

- 7-round periodic distinguisher for SKINNY is shown in Fig. 21.
- 9-round periodic distinguisher for SKINNY is shown in Fig. 22, requiring 6 collisions.

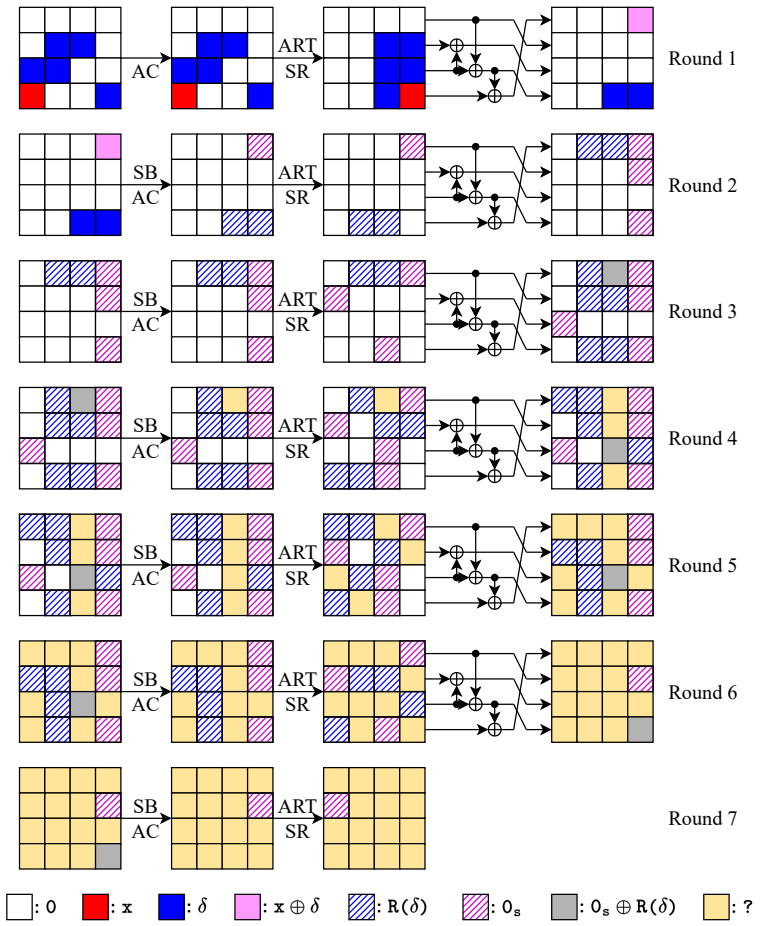

Fig. 21: 7-round distinguisher of SKINNY.

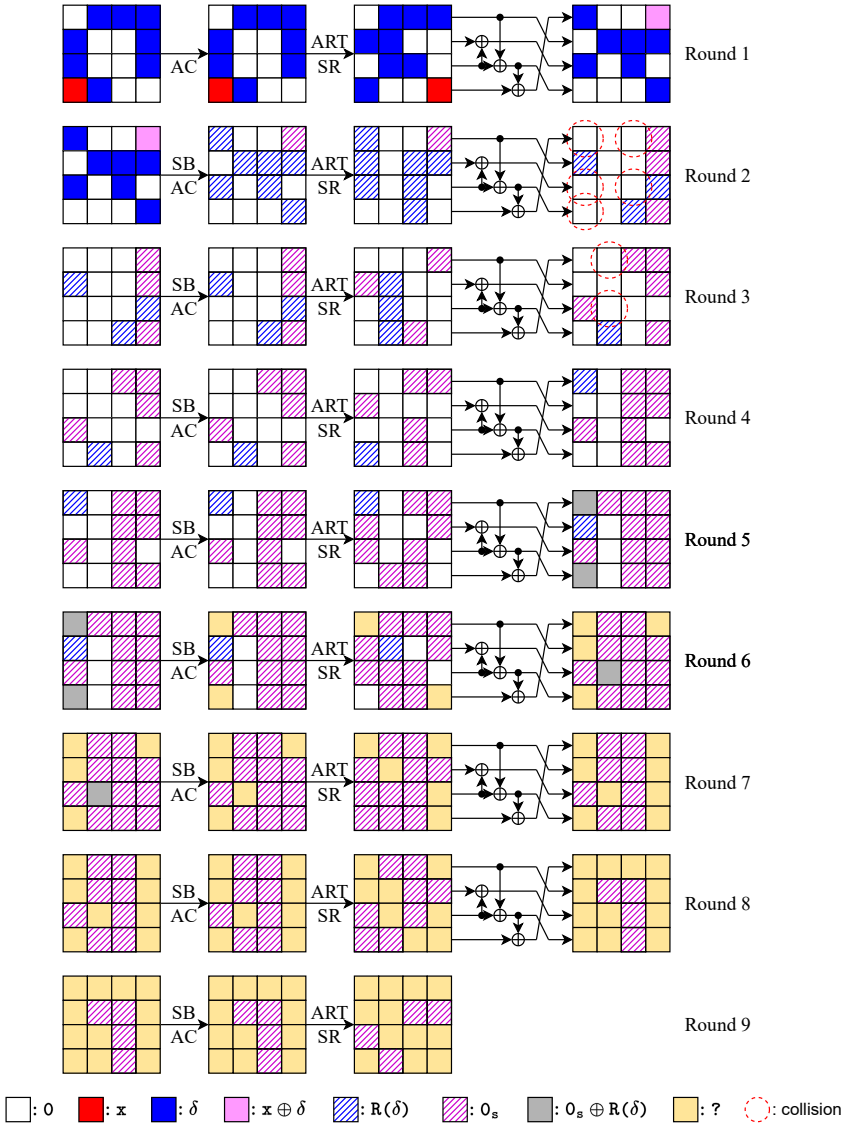

Fig. 22: 9-round distinguisher of SKINNY.

H Periodic Distinguishers for CRAFT

In this section, we present the periodic distinguishers for CRAFT. The final round is omitted.

- 8-round periodic distinguisher for CRAFT is shown in Fig. 23.
- 12-round periodic distinguisher for CRAFT is shown in Fig. 24, requiring 10 collisions.

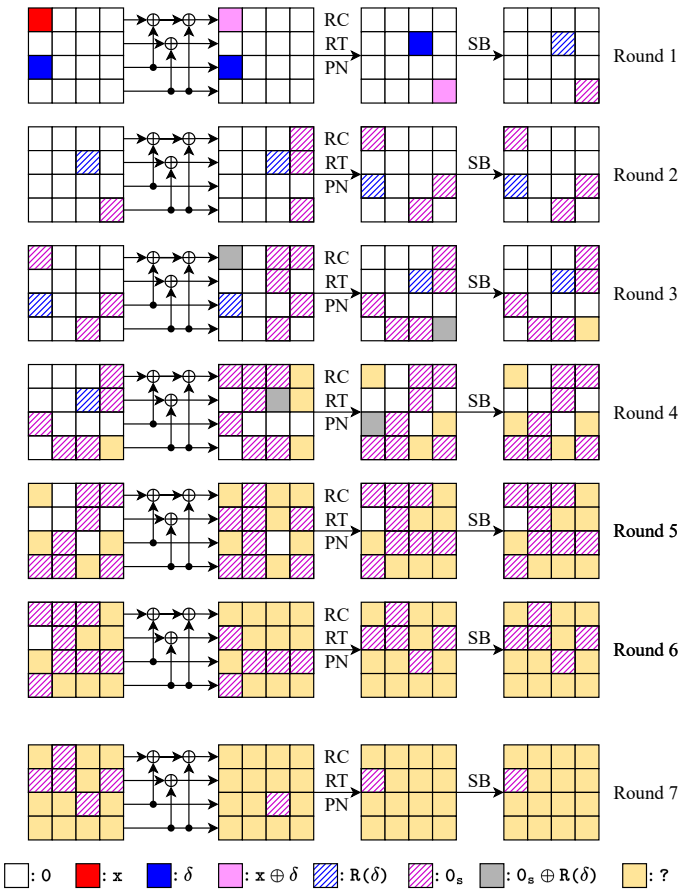

Fig. 23: 8-round distinguisher of CRAFT.

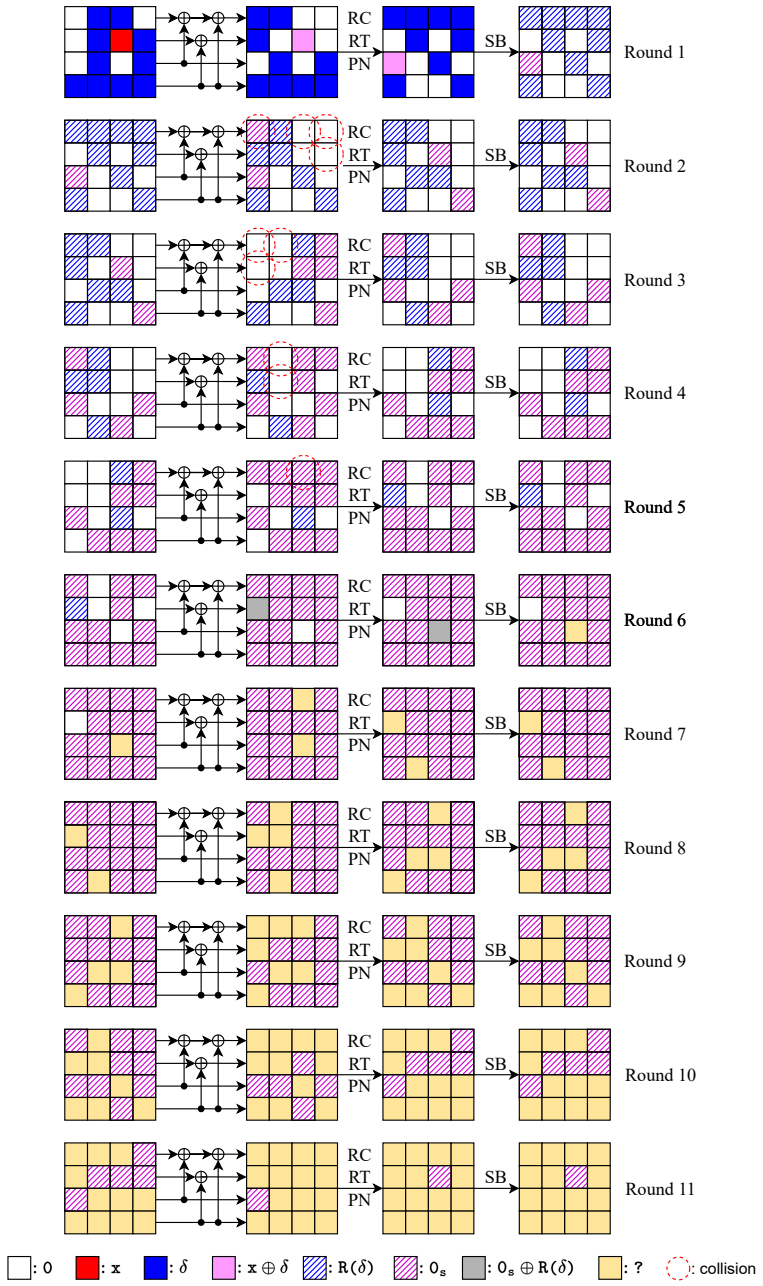

Fig. 24: 12-round distinguisher of CRAFT.

I 4-round Periodic Distinguisher for Piccolo

We present the 4-round periodic distinguishers for Piccolo, which is shown in Fig. 25.

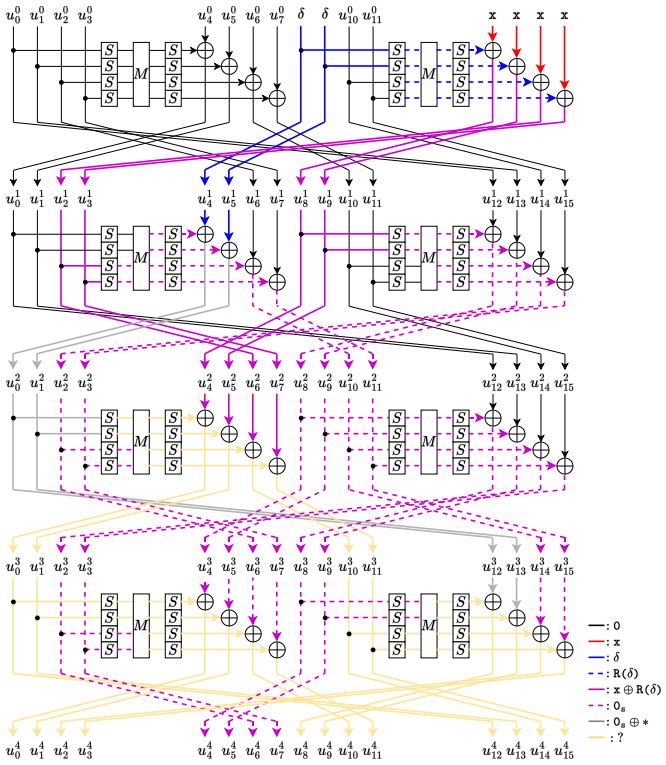

Fig. 25: 4-round distinguisher of Piccolo.

J 17/18-round Periodic Distinguishers for CAST-256

In this section, we present the periodic distinguishers for CAST-256.

- 17-round periodic distinguisher for CAST-256 is shown in Fig. 26. The first two rounds are encryption, and the following 15 rounds are decryption.
- 18-round periodic distinguisher for CAST-256 is shown in Fig. 27, requiring 1 collision. The first three rounds are encryption, and the following 15 rounds are decryption.

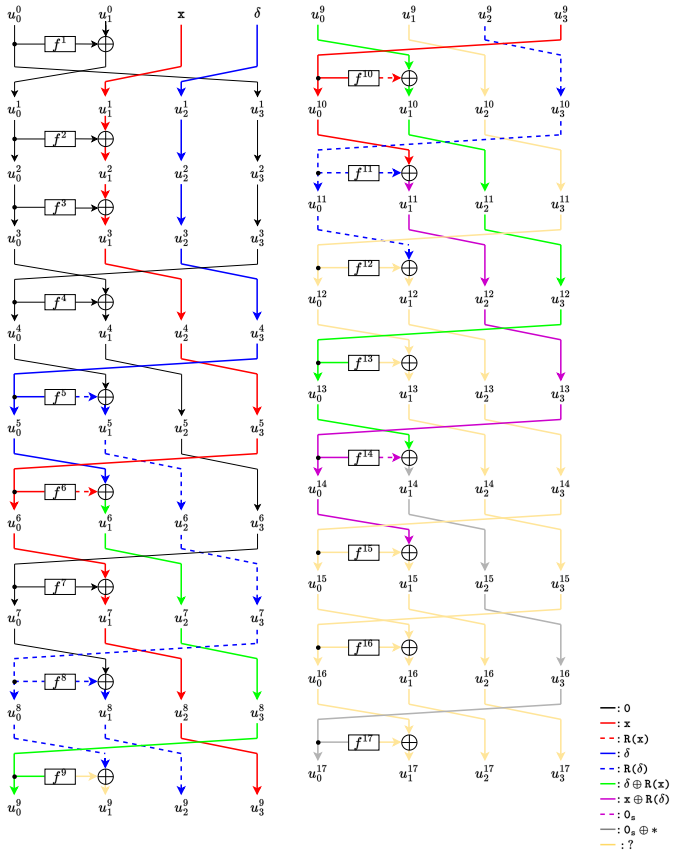

Fig. 26: 17-round distinguisher of CAST-256.

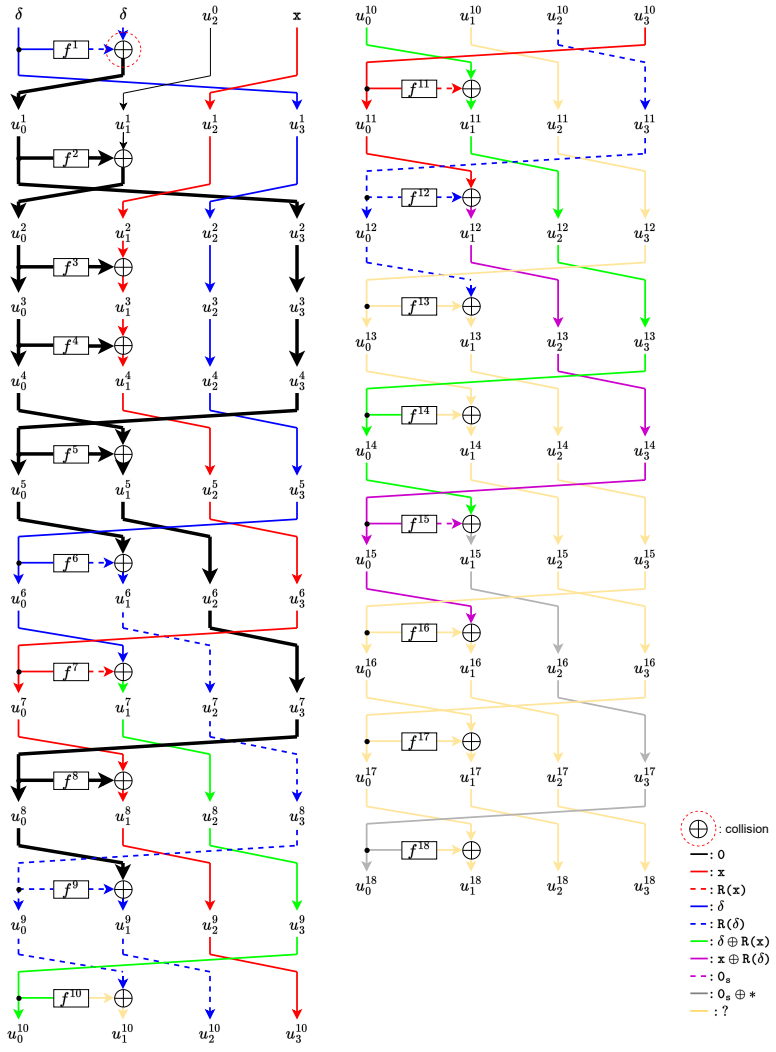

Fig. 27: 18-round distinguisher of CAST-256.

K A Reinterpretation of Original Periodic Distinguishers for Type-1/2/3 GFS

The round function for Type-1 GFS is shown in Fig. 28.

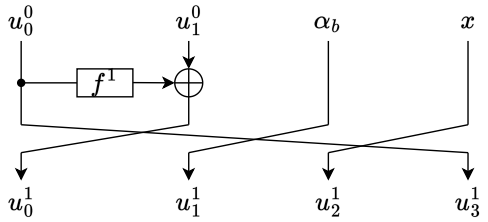

Fig. 28: The round function of Type-1 GFS.

The decryption round function for Type-1 GFS is shown in Fig. 29.

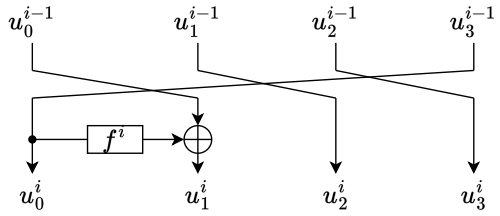

Fig. 29: The decryption round function of Type-1 GFS.

The round function for Type-2 GFS is shown in Fig. 30.

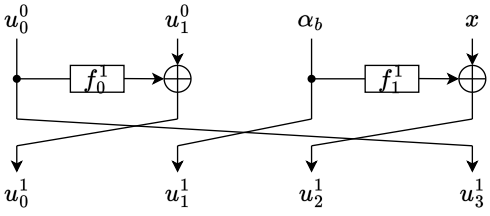

Fig. 30: The round function of Type-2 GFS.

The round function for Type-3 GFS is shown in Fig. 31.

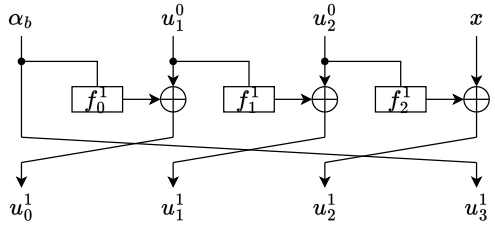

Fig. 31: The round function of Type-3 GFS.

In this section, we present the periodic distinguishers for Type-1/2/3 GFS.

- 9-round periodic distinguisher for Type-1 GFS is shown in Fig. 32.
- 15-round periodic distinguisher (CCA) for Type-1 GFS is shown in Fig. 33.
- 5-round periodic distinguisher for Type-2 GFS is shown in Fig. 34.
- 5-round periodic distinguisher for Type-3 GFS is shown in Fig. 35.

All distinguishers are derived from our model and are consistent with the current best results. This demonstrates the powerful effectiveness of our model.

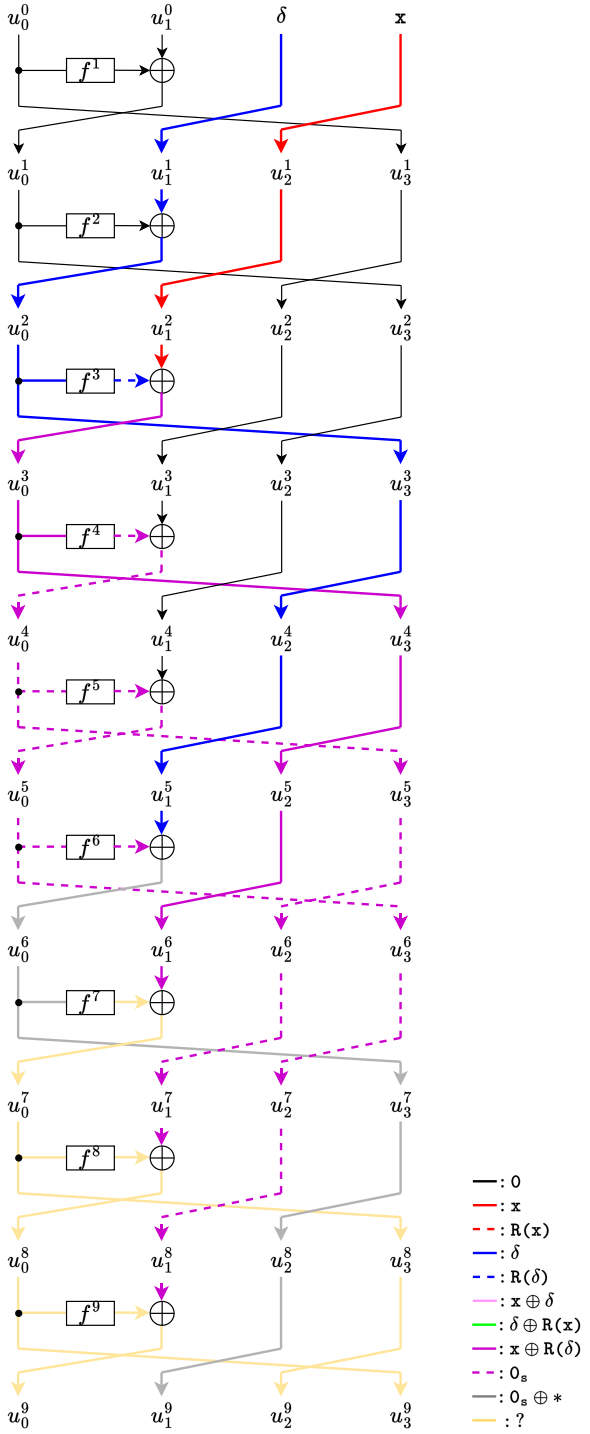

Fig. 32: 9-round distinguisher of Type-1 GFS.

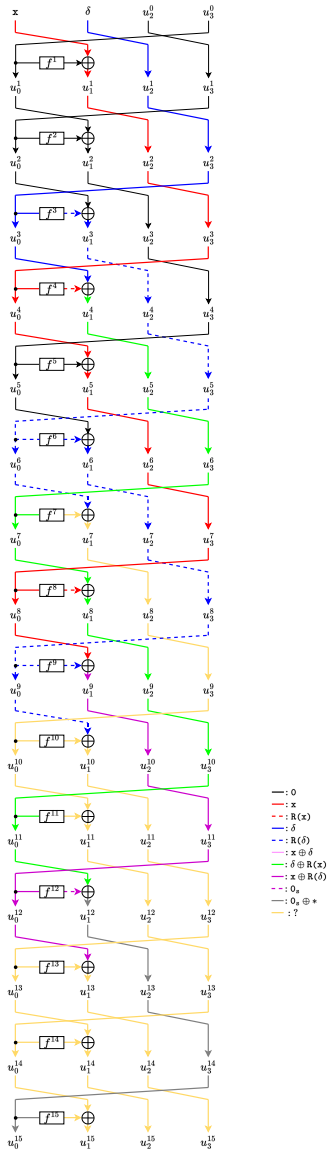

Fig. 33: 15-round distinguisher (CCA) of Type-1 GFS.

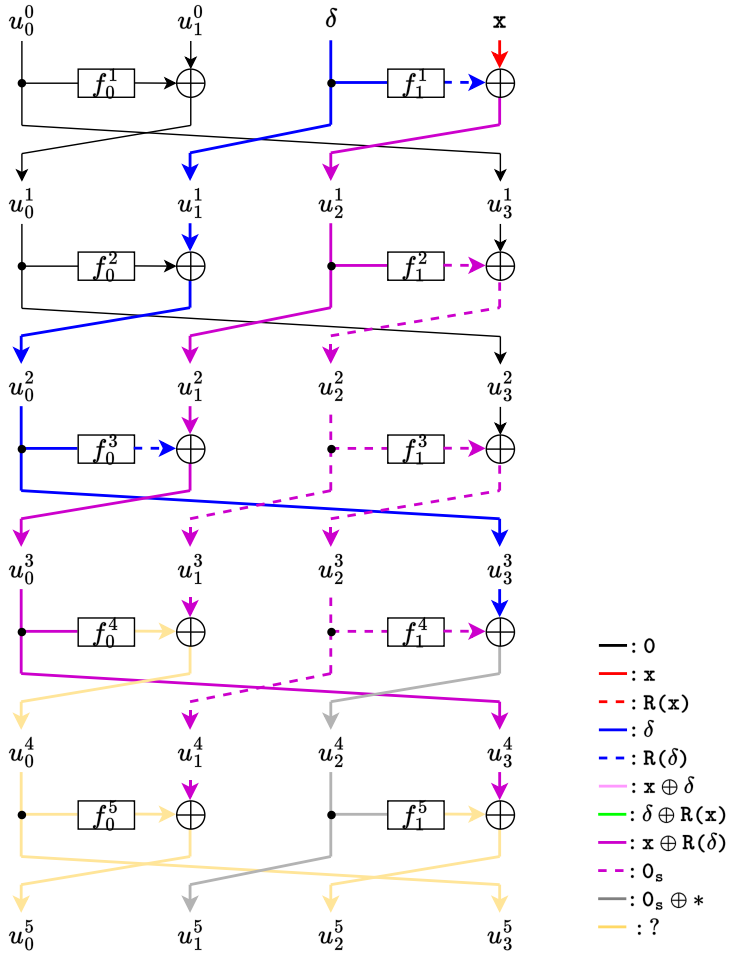

Fig. 34: 5-round distinguisher of Type-2 GFS.

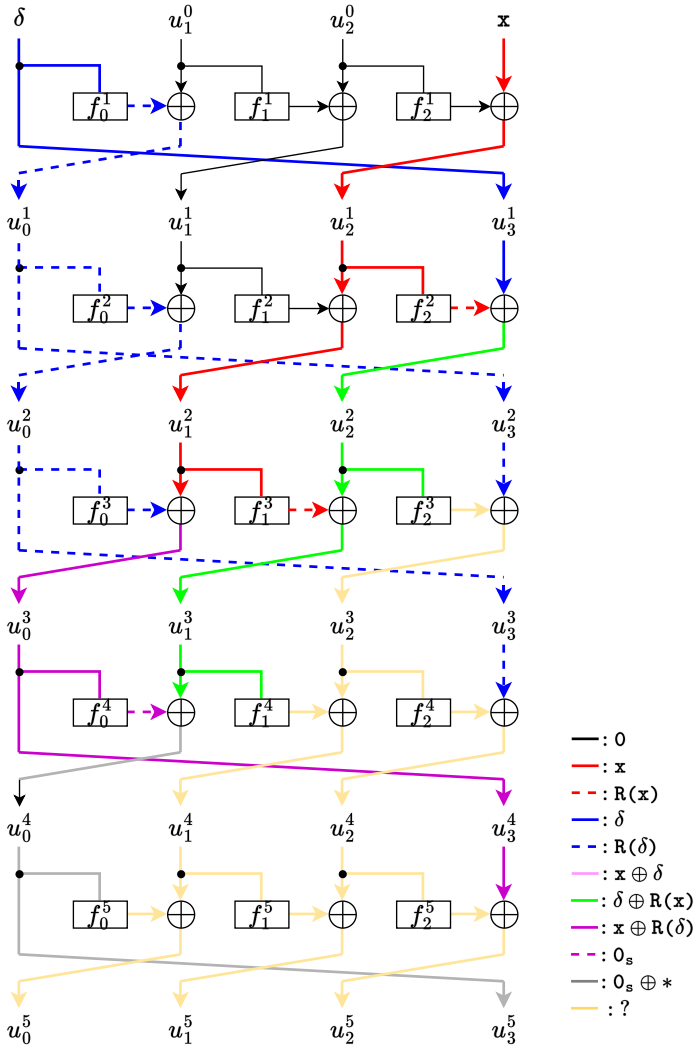

Fig. 35: 5-round distinguisher of Type-3 GFS.