A Unified Key Recovery Framework for Impossible Boomerang Attacks: Applications to Full-Round-ARADI and SKINNYe v2

Xichao Hu, Lin Jiao, Dengguo Feng, Yongqiang Li, Senpeng Wang, Yonglin Hao, Xinxin Gong

Abstract

The impossible boomerang attack is a powerful cryptanalytic technique, but existing key recovery methods face several limitations that restrict its applicability. Specifically, the key pre-guessing is coarse-grained, S-box details are ignored in the differential propagation, the complexity estimation and the key guessing order determination remain rudimentary. To overcome these issues, we introduce three key improvement measures. First, we propose a flexible partial key and difference pre-guessing technique based on directed graphs, enabling selective identification of required keys and differences for generating partial pairs and quartets. Second, we propose a pre-sieving technique to early eliminate invalid quartets by exploiting cipher-specific details. Third, we introduce an automatic key-guessing strategy based on the same directed graphs to efficiently determine valid guessing orders. We integrate these techniques to develop a unified key recovery framework for impossible boomerang attacks, accompanied by a formal and precise characterization of the overall complexity. This is the first framework to support flexible key and difference pre-guessing while incorporating block cipher details during key recovery for impossible boomerang attacks. Crucially, it enables the automatic generation of detailed recovery steps, a capability missing in prior work. As applications, under the four related-key/tweakey setting, we apply the framework to ARADI, a low-latency cipher proposed by the National Security Agency (NSA), and SKINNYe v2, a threshold-implementation-friendly cipher proposed at EUROCRYPT 2020. For ARADI, we achieve the first full-round attack with 2^{130} data, $2^{253.78}$ time, and $2^{235.75}$ memory complexity. For SKINNYe v2, we present the first 34-round impossible boomerang attack with 2^{66} data, $2^{253.75}$ time, and $2^{239.75}$ memory complexity. These results demonstrate the framework's significance and its substantial improvement in advancing the impossible boomerang attack.

Index Terms

ARADI, SKINNYe v2, Impossible boomerang attack, Key recovery, Block cipher

I. Introduction

THE impossible boomerang attack (IB attack) is a universal key recovery cryptanalysis method for block ciphers, initially introduced and extended to related-key scenarios by Lu in [22], [23]. This method has successfully targeted 6-round AES-128, 7-round AES-192/AES-256 [11] in single-key setting, as well as 8-round AES-192 and 9-round AES-256 in related-key setting. The term RK-IB attack refers to IB attack in the related-key setting, while (RK-)IB attack denotes IB attack in either setting, depending on context.

The core of IB attacks lies in the impossible boomerang distinguisher (IB distinguisher), whose fundamental concept can be best explained through a boomerang distinguisher with zero probability. Specifically, for a block cipher E_d, given two input differences α, α' and two output differences β, β', if no pair of plaintexts (x_1, x_2) satisfies

$$E_d(x_1) \oplus E_d(x_2) = \beta, E_d(x_1 \oplus \alpha) \oplus E_d(x_2 \oplus \alpha') = \beta',$$

then $(\alpha, \alpha') \nrightarrow (\beta, \beta')$ forms an IB distinguisher of E_d. The term RK-IB distinguisher refers to IB distinguisher in the related-key setting, while (RK-)IB distinguisher denotes IB distinguisher in either single-key or related-key setting, depending on context.

For constructing (RK-)IB distinguishers, Lu's method [22] decomposes the block cipher E_d into two sub-ciphers E_0 and E_1 (i.e., $E_d = E_1 \circ E_0$). Exactly, $(\alpha, \alpha') \nrightarrow (\beta, \beta')$ holds if for $\forall \gamma, \gamma', \delta, \delta'$ such that $\alpha \xrightarrow{E_0} \gamma$, $\alpha' \xrightarrow{E_0} \gamma'$, $\beta \xrightarrow{E_1^{-1}} \delta$ and $\beta' \xrightarrow{E_1^{-1}} \delta'$, it follows that $\gamma \oplus \gamma' \oplus \delta \oplus \delta' \neq 0$. However, this method ignores dependencies between the sub-ciphers, as noted by Murphy [25], potentially hindering the discovery of longer (RK-)IB distinguishers. With the advancement of boomerang attacks, Dunkelman et al. [14], [15], introduced the sandwich framework, dividing E_d into three parts: $E_1 \circ E_m \circ E_0$, as depicted in Figure 1. To assess the boomerang behavior on E_m, tools like the Boomerang Connectivity Table (BCT) [10], Double BCT (DBCT) [12], [29], and others [6] were developed. Building on BCT and

Xichao Hu, Lin Jiao, Dengguo Feng, Yonglin Hao, Xinxin Gong are with State Key Laboratory of Cryptology, Beijing, China

Yongqiang Li is with State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and are also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Senpeng Wang is with Information Engineering University, Zhengzhou, China

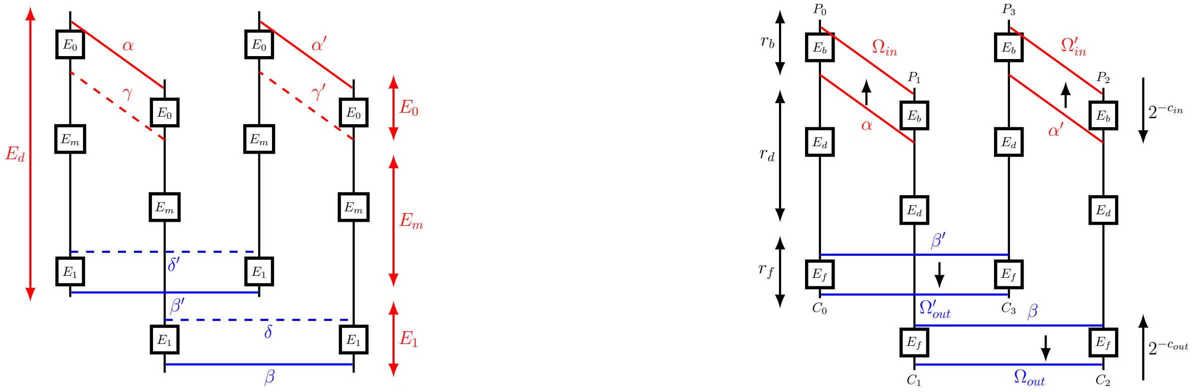


Fig. 1: The `IB distinguisher` and its extended `IB attack`.

DBCT, two recent studies [5], [32] proposed MILP-based and CP-based methods for constructing. More recently, Hu et al. introduced the $\mathcal{HJF}$-method based on state propagation and the Generalized Extended BCT (GEBCT) [19], enabling more universal searches for longer-round `(RK-)IB distinguishers`.

To launch the `(RK-)IB attack`, given an `(RK-)IB distinguisher`, an attacker extends the attack by r_b rounds before and r_f rounds after the distinguisher, as shown in Figure 1. Two primary key recovery methods exist: impossible differential style (IDS) and boomerang style (BS) [5], [23], [32]. In IDS, the attacker builds quartets that may satisfy the input and output differences of the `(RK-)IB distinguisher`, then applies the early abort technique [24] and discard incorrect key guesses. In BS, the attacker first guesses all necessary keys in the first r_b rounds (resp. last r_f rounds) to construct quartets that satisfy the input differences and potentially align with the output differences of the `(RK-)IB distinguisher`, followed by the early abort technique. Early `(RK-)IB attacks` were applied manually to `AES` [23], but recent work [5], [32] uses automatic methods, achieving new results on ciphers like `SKINNY` [1] and `SKINNYee` [27].

Compared with other differential attacks on block ciphers, the research and application of `(RK-)IB attack` remain relatively underdeveloped, reflected in the following two areas:

- Distinguisher construction methods. Current key recovery frameworks [5], [32] rely on distinguishers constructed by BCT- and DBCT-based methods, which limits the applicability and power of `(RK-)IB attack` when compared to the use of longer and more abundant distinguishers generated by the more general and efficient $\mathcal{HJF}$-method.
- Key recovery techniques. Several limitations persist.
  1) Currently, only the above two key recovery methods, IDS and BS, are employed in `(RK-)IB attack`. Both use coarse-grained key pre-guessing strategies: one guesses no keys early, the other guesses all added-round keys at one end at once[1] The lack of a fine-grained key pre-guessing strategy significantly increases complexity and often leads to failed attacks. Furthermore, existing techniques do not account for the impact of pre-guessing partial differences, which further diminishes the attack's effectiveness.
  2) The differential propagation analysis for extending distinguishers focuses only on the positions of active bits, neglecting the possible differential patterns corresponding to the details of S-boxes. This oversight undoubtedly increases the number of invalid quartets and raises overall complexity.
  3) The early abort technique [24] plays a pivotal role in the key recovery process. Unlike brute-force methods that guess all keys at once, it uses a stepwise key-guessing strategy, thereby reducing the attack complexity. Current key recovery methods use approximate formulas to estimate the complexity of the early abort technique but fail to determine the optimal key guessing order. As a result, attackers must manually design the key recovery process, which is both complex and time-consuming.

Our contributions. This paper aims to comprehensively enhance the effectiveness of `(RK-)IB attack`. To systematically address the aforementioned limitations, we employ the $\mathcal{HJF}$-method to search for `(RK-)IB distinguishers`

[1] Two additional concurrent works [9], [31], though published later than ours (see the ePrint release record), warrant mention. Our work is the first to introduce the partial key pre-guessing technique in `(RK-)IB attack`, together with the difference pre-guessing technique, pre-sieving technique, and automatic key-guessing strategy. By integrating these new techniques into a cohesive whole, we propose a unified key recovery framework tailored for `(RK-)IB attacks`, enabling precise formal evaluation of attack complexity. Additionally, we build upon the general $\mathcal{HJF}$-method for searching longer `(RK-)IB distinguishers`. In contrast,

- In [9], Chen et al. propose a key recovery framework using the partial key pre-guessing technique, building on `(RK-)IB distinguishers`from prior work [32] and constructing an MILP model to automatically find the optimal key recovery attack. They achieve the first 33-round `(RK-)IB attack` on `SKINNYe v2`, one round fewer than our result.
- In [31], Zhang et al. also develop a key recovery framework with partial key pre-guessing and introduce an MILP-based tool that integrates distinguisher search and key recovery, proposing a new method for finding `(RK-)IB distinguishers`.

TABLE I: Summary of the cryptanalytic results by related-keys impossible boomerang attacks.

Cipher	Round	Complexity			Reference	Remarks
	(attacked/full)	Time	Data	Memory		
ARADI	16/16	$2^{253.78}$	2^{130}	$2^{235.75}$	Section V-A	First full-round attack
SKINNYe v2	33/44	2^{232}	2^{70}	2^{232}	[9]	
	33/44	$2^{249.46}$	$2^{68.25}$	2^{160}	[9]	
	34/44	$2^{253.75}$	2^{66}	$2^{239.75}$	Section V-B	Best RK-IB attack; 3 more rounds than the best RK-ID attack [18]

† The optimal attack on SKINNYe v2 to our knowledge is the 38-round rectangle attack [28].

and, more significantly, propose a series of advanced key recovery techniques.

1) **Pre-guessing Technique.** We propose a flexible partial key and difference pre-guessing technique based on directed graphs. Two directed graphs are built for the adding rounds at both ends of the (RK-)IB distinguishers using a well-designed method, capturing definite forward and backward differential propagation with associated keys. By analyzing subgraphs within these directed graphs, we determine which key and difference guesses are needed to generate partial plaintext or ciphertext pairs. This allows full planning of pre-guessing for keys and differences in both added rounds, reducing large-scale guessing complexity and filtering out invalid quartets early.
2) **Pre-sieving technique.** We use details from both linear and nonlinear layers to precisely identify possible difference patterns, enabling early elimination of invalid quartets.
3) **Automatic key-guessing strategy.** When using pre-sieving or early abort technique, determining the optimal key guessing order is crucial. Our automatic key-guessing strategy uses directed graphs to select, at each step, the filter block requiring the fewest key bit guesses. Unlike traditional methods that rely on a two-step process involving complexity estimation followed by manual derivation of guessing steps and complexity verification, the proposed approach eliminates the need for estimation and directly generates the key recovery order.

By combining these new techniques, we propose a unified key recovery framework tailored for (RK-)IB attacks with exact formal attack complexity. This is the first framework to support flexible key and difference pre-guessing and integrate block cipher specifics during key recovery. More importantly, it automatically generates detailed key recovery steps—a capability not available in prior work. As applications, under the four related-keys/related-tweakeys setting, we apply the IB attack to ARADI and SKINNYe v2, with results shown in Table I and detailed below.

ARADI is a block cipher with a 128-bit block size and a 256-bit key size, designed by the National Security Agency (NSA) [16]. It targets memory encryption, where low latency is critical to avoid delaying RAM access. As with many NSA-developed cryptographic standards, ARADI has received significant security scrutiny [2], [3], [20]. However, prior analyses do not pose serious threats. In this work, we achieve the results of RK-IB distinguishers and RK-IB attacks on ARADI as follows.

- 11-round Distinguishers. By analyzing the linear key schedule of ARADI, we find 3-round related-key differentials with probability 1. Using two such differentials and the $\mathcal{HJF}$-method, we obtain 97 11-round RK-IB distinguishers.
- First full-round attack. We extend the attack by adding 2 rounds before and 3 after these 11-round distinguishers, and apply our new key recovery framework to mount a full-round attack on ARADI, with data complexity 2^{130}, time complexity $2^{253.78}$, and memory complexity $2^{235.75}$.

This result indicates that ARADI is completely broken. Although the designers stated that “related-key security is not a significant concern for memory encryption”, they do not negate the feasibility of related-key attacks. Furthermore, we demonstrate that ARADI’s lightweight S-box is vulnerable to pre-sieving technique, and its linear layer is weak against partial key and difference pre-guessing technique, enabling effective RK-IB attacks.

SKINNYe v2 is a threshold implementation friendly block cipher for the authenticated encryption PFB_Plus proposed at EUROCRYPT 2020 [26]. It has a 256-bit tweakey, 64-bit block, and 4-bit cell size. In this work, we achieve the results of RK-IB distinguishers and RK-IB attacks on SKINNYe v2 as follows.

- 23-round Distinguishers. By analyzing the linear tweakey schedule of SKINNYe v2, we derive 8-round related-key differentials with probability 1. Using two such differentials and the $\mathcal{HJF}$-method, we obtain 381 23-round RK-IB distinguishers.
- First 34-round Attack. We extend the attack by adding 6 rounds before and 5 after these 23-round distinguishers, and apply our new key recovery framework to mount a 34-round attack on SKINNYe v2, with data complexity 2^{66}, time complexity $2^{253.75}$, and memory complexity $2^{239.75}$.

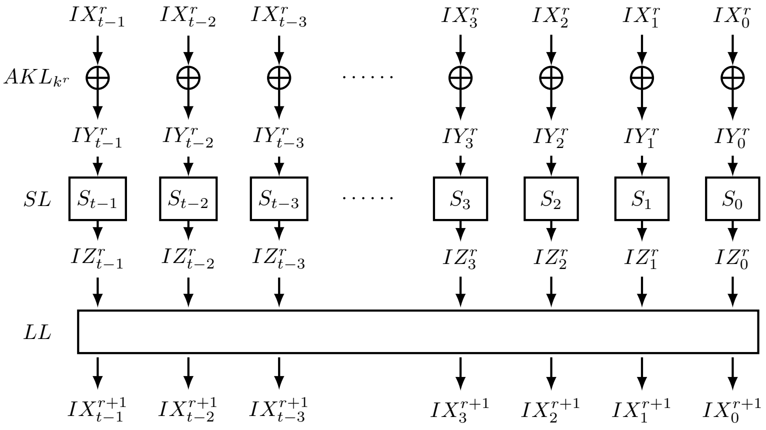


Fig. 2: One round of SPN block cipher.

The impossible differential attack is a highly effective and widely studied attack method [4], [8], [17], [21], [30], [33], while (RK-)IB attack is also an impossible attacks relying on zero differential probability distinguishers. Compared to the best known related-key impossible differential attack (RK-ID) that reaches 31 rounds [18], our method covers 3 additional rounds, demonstrating superior effectiveness within the impossible attacks. Compared to the best known RK-IB attacksthat reaches 33 rounds, which was a concurrent work but published later than ours [9], our method covers 1 additional rounds, highlighting the strength of our framework.

Notably, without our proposed techniques, a full-round RK-IB attack on ARADI and a 34-round RK-IB attack on SKINNYe v2 are unachievable, regardless of whether impossible differential or boomerang style is used. Particularly, the pre-sieving technique is crucial for the attack on ARADI—without it, the full-round attack fails. This highlights the effectiveness of our key recovery framework for (RK-)IB attacks.

Outline. Section 2 introduces the notations and related work. Section 3 presents the new techniques that facilitate (RK-)IB attacks. Section 4 provides a unified key recovery framework for (RK-)IB attacks based on these techniques. Section 5 provides a detailed description of the full-round attack on ARADI and 34-round attack on SKINNYe v2. Finally, we summarize this paper in Section 6.

II. Preliminaries

Our key recovery method applies to S-box-based block ciphers. For clarity, we use the SPN cipher as an example and introduce the corresponding notations.

A. Notations

Let E be an n-bit SPN block cipher with an m-bit key. One round of E is shown in Figure 2 and consists of three fundamental operations:

- SL: the S-box layer, applies t parallel q-bit S-boxes to introduce non-linearity;
- LL: the linear layer, applies a global linear transformation to enhance diffusion;
- $\mathtt{AKL}_{k^r}$: the key addition layer, XORs the round key k^r with the state.

The state, difference, and key are divided into cells according to S-box size. The following notations are used throughout:

- $\mathbb{Z}_n$: the set $\{0, 1, \ldots, n-1\}$.
- $\alpha \xrightarrow{F} \beta$: input difference α propagates to output difference β through function F with nonzero probability.
- $K_i, i = 0, 1, 2, 3$: the keys of E in the related-key setting.
- $T_i, i = 0, 1, 2, 3$: the plaintext-ciphertext sets encrypted under K_i.
- $IX^r_i, IY^r_i, IZ^r_i, i = 0, 1, 2, 3$: the states before and after each component in round r, as shown in Figure 2.
- $IK^r_i, i = 0, 1, 2, 3$: the round key in round r under K_i; $IK^r_{i,j}$: the j-th cell of IK^r_i, analogous for IX, IY, IZ.
- $\Delta X^r_{01}, \Delta X^r_{23}$: the upper trail differences in the IB distinguisher, where $\Delta X^r_{01} = IX^r_0 \oplus IX^r_1$ and $\Delta X^r_{23} = IX^r_2 \oplus IX^r_3$; analogous for IY, IZ, IK.
- $\nabla X^r_{12}, \nabla X^r_{03}$: the lower trail differences in the IB distinguisher, where $\nabla X^r_{12} = X^r_1 \oplus X^r_2$ and $\nabla X^r_{03} = X^r_0 \oplus X^r_3$; analogous for IY, IZ, IK.
- $\mathcal{N}(\beta)$: the number of input differences that can propagate to the output difference β through the S-box.

The notations of an IB attack according to Figure 1 are as follows:

- α, α' (resp. β, β'): input differences (resp. output differences) of the IB distinguisher.

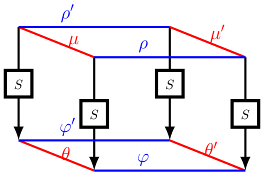


Fig. 3: The illustrations of GEBCT.

- Ω_{in} (resp. Ω_{out}): the set of plaintext (resp. ciphertext) differences that may lead to the input difference α (resp. output difference β) of the `IB distinguisher` under the key difference, with $|\Omega_{in}| = 2^{d_{in}}$ (resp. $|\Omega_{out}| = 2^{d_{out}}$).
- $2^{-c_{in}}$ (resp. $2^{-c_{out}}$): the probability of reaching input difference α (resp. output difference β) from a plaintext (resp. ciphertext) difference in Ω_{in} (resp. Ω_{out}).
- K_{in} (resp. K_{out}): the key bits involved in the `IB attack` in E_b (resp. E_f).

B. The definitions of (related-key) impossible boomerang distinguishers

The `(RK-)IB distinguisher` is defined as follows.

Definition 1 ([22], [23]). *Given a block cipher $E : \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ under four keys $K_i \in \mathbb{F}_2^m$, $i = 0, 1, 2, 3$, and state differences $\alpha, \alpha', \beta, \beta'$, along with key differences $\kappa_{01} = K_0 \oplus K_1$, $\kappa_{12} = K_1 \oplus K_2$, $\kappa_{23} = K_2 \oplus K_3$, $\kappa_{03} = K_0 \oplus K_3$ (where $\kappa_{03} = \kappa_{01} \oplus \kappa_{12} \oplus \kappa_{23}$), if no plaintext pair (x_1, x_2) satisfies*

$$E_{K_1}(x_1) \oplus E_{K_2}(x_2) = \beta, \quad E_{K_0}(x_1 \oplus \alpha) \oplus E_{K_3}(x_2 \oplus \alpha') = \beta', \tag{1}$$

then $(\alpha, \alpha', \beta, \beta')$ is an `RK-IB distinguisher` of E under the key differences $(\kappa_{01}, \kappa_{23}, \kappa_{12}, \kappa_{03})$. In the case where $K = K_0 = K_1 = K_2 = K_3$, it is an `IB distinguisher` of E under K[2]*.*

Next, we present the construction method of `(RK-)IB distinguisher` employed in this work. Among existing approaches, we select the GEBCT-based approach and state-propagation-based approach within the $\mathcal{HJF}$-method [19], as they enable searching contradictions over more rounds and exploit finer cipher details compared to BCT/DBCT-based approaches [5], [32], thus yielding longer `(RK-)IB distinguishers`.

- The GEBCT-based approach constructs `(RK-)IB distinguishers` using differential propagation rules, with non-linear operations following the GEBCT. The GEBCT is defined in Definition 2 and illustrated in Figure 3.
- The state-propagation-based approach constructs `(RK-)IB distinguishers` using state propagation rules [19]. Hu et al. proved that distinguishers built with this approach are equivalent to the essential definition of `(RK-)IB distinguishers`.

These two methods complement each other in terms of search efficiency and modeling accuracy, and can be used for automatic cross-validation of search results.

Definition 2 ([19]). *Let S be a permutation of $\mathbb{F}_2^n$, and $\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi' \in \mathbb{F}_2^n$. The GEBCT of S is the table:*

$$\mathrm{GEBCT}(\mu, \mu', \rho, \rho', \theta, \theta', \varphi, \varphi') = \left\{ (x_0, x_1, x_2, x_3) \in \mathbb{F}_2^{4n} \;\middle|\; \begin{array}{c} x_0 \oplus x_1 = \mu, \\ x_2 \oplus x_3 = \mu', \\ x_1 \oplus x_2 = \rho, \\ x_0 \oplus x_3 = \rho', \\ S(x_0) \oplus S(x_1) = \theta, \\ S(x_2) \oplus S(x_3) = \theta', \\ S(x_1) \oplus S(x_2) = \varphi, \\ S(x_0) \oplus S(x_3) = \varphi' \end{array} \right\}.$$

C. Early abort technique

We review the common technique of key recovery attacks, the early abort technique [24]. Instead of guessing all of the required round key bits $K_{in} \cup K_{out}$ at once, attackers can guess parts of $K_{in} \cup K_{out}$ step by step depending on the round function, partially verifying whether a plaintext or ciphertext pair yields the expected difference of the distinguisher and discarding invalid pairs early, which reduces the whole computational workload. We use `AES` as an example to illustrate the early abort technique.

[2] Single-key impossible boomerang attacks have limited value, as a single-key `IB distinguisher` typically allows deriving an impossible differential of the same round count [23], [5]. Although new construction methods like the $\mathcal{HJF}$ have found exceptions such as `PRINTcipher48` with key-dependent permutations, the practical utility of single-key `IB attacks` is generally modest. This paper uses `IB attacks` as examples for simplicity and clarity.

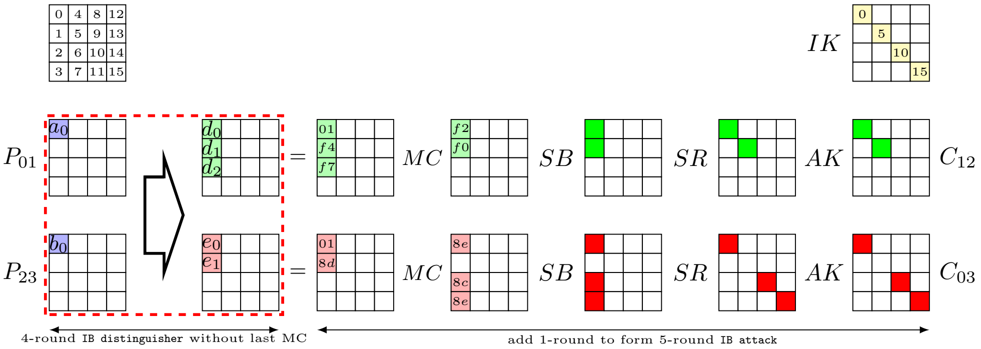


Fig. 4: The 5-round `IB attack` of `AES`.

Example 1. *Consider the 5-round `IB attack` of `AES`. As demonstrated in [23], for* $\forall a_0, b_0, d_0, d_1, d_2, e_0, e_1 \in (\mathbb{F}_2^8)^*$,

$$((a_0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),(b_0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)) \not\twoheadrightarrow$$
$$((d_0,d_1,d_2,0,0,0,0,0,0,0,0,0,0,0,0,0),(e_0,e_1,0,0,0,0,0,0,0,0,0,0,0,0,0,0))$$

is a 4-round `IB distinguisher` of `AES` without the last MC operation. The byte ordering is shown in Figure 4. To simplify the attack, we set $d_0 = 0x01$, $d_1 = 0xf4$, $d_2 = 0xf7$, $e_0 = 0x01$, $e_1 = 0x8d$, *and extend one round after the distinguisher to form a 5-round `IB attack`, as shown in Figure 4.*

Let P_i, C_i, $0 \le i \le 3$ *denote plaintexts and ciphertexts. Suppose we have* $\mathcal{Q}$ *quartets* $((P_0, C_0), \dots, (P_3, C_3))$ *satisfying:*

$$\begin{cases} P_{0,i_0} \oplus P_{1,i_0} = 0, & i_0 \in \mathbb{Z}_{16}/\{0\}, \\ P_{2,i_1} \oplus P_{3,i_1} = 0, & i_1 \in \mathbb{Z}_{16}/\{0\}, \\ C_{1,j_0} \oplus C_{2,j_0} = 0, & j_0 \in \mathbb{Z}_{16}/\{0,5\}, \\ C_{0,j_1} \oplus C_{3,j_1} = 0, & j_1 \in \mathbb{Z}_{16}/\{0,10,15\}, \end{cases}$$

where $P_{i,j}$ *and* $C_{i,j}$ *denote the* j*-th byte of* P_i *and* C_i*, respectively. Let* $IK_0, IK_5, IK_{10}, IK_{15}$ *be the round key bytes to guess (highlighted in yellow in Figure 4), corresponding to the AK operation. Then, the quartets should satisfy:*

$$\begin{cases} S^{-1}(C_{1,0} \oplus IK_0) \oplus S^{-1}(C_{2,0} \oplus IK_0) = 0xf2, \\ S^{-1}(C_{1,1} \oplus IK_5) \oplus S^{-1}(C_{2,1} \oplus IK_5) = 0xf0, \\ S^{-1}(C_{0,0} \oplus IK_0) \oplus S^{-1}(C_{3,0} \oplus IK_0) = 0x8e, \\ S^{-1}(C_{0,10} \oplus IK_{10}) \oplus S^{-1}(C_{3,10} \oplus IK_{10}) = 0x8c, \\ S^{-1}(C_{0,15} \oplus IK_{15}) \oplus S^{-1}(C_{3,15} \oplus IK_{15}) = 0x8e, \end{cases}$$

and the candidate keys satisfying all equations are discarded. Without early abort technique, guessing all four IK *bytes yields a time complexity of* $\mathcal{Q} \cdot 2^{32}$*. With early abort technique, the process proceeds stepwise:*

1) *Guess* IK_0*, partially decrypt, and discard quartets failing* $S^{-1}(C_{1,0} \oplus IK_0) \oplus S^{-1}(C_{2,0} \oplus IK_0) = 0xf2$ *or* $S^{-1}(C_{0,0} \oplus IK_0) \oplus S^{-1}(C_{3,0} \oplus IK_0) = 0x8e$*. The time complexity for this step is* $\mathcal{Q} \cdot 2^8$*, and the quartets remain* $\mathcal{Q} \cdot 2^{-16}$ *for each key guess.*
2) *Guess* IK_5*, partially decrypt, and discard quartets failing* $S^{-1}(C_{1,1} \oplus IK_5) \oplus S^{-1}(C_{2,1} \oplus IK_5) = 0xf0$*. The time complexity for this step is* $\mathcal{Q} \cdot 2^{-16} \cdot 2^{16} = \mathcal{Q}$*, and the quartets remain* $\mathcal{Q} \cdot 2^{-24}$ *for each key guess.*
3) *Guess* IK_{10}*, partially decrypt, and discard quartets failing* $S^{-1}(C_{0,10} \oplus IK_{10}) \oplus S^{-1}(C_{3,10} \oplus IK_{10}) = 0x8c$*. The time complexity for this step is* $\mathcal{Q} \cdot 2^{-24} \cdot 2^{24} = \mathcal{Q}$*, and the quartets remain* $\mathcal{Q} \cdot 2^{-32}$ *for each key guess.*
4) *Guess* IK_{15}*, partially decrypt, and discard quartets failing* $S^{-1}(C_{0,15} \oplus IK_{15}) \oplus S^{-1}(C_{3,15} \oplus IK_{15}) = 0x8e$*. The time complexity for this step is* $\mathcal{Q} \cdot 2^{-32} \cdot 2^{32} = \mathcal{Q}$*, and the quartets remain* $\mathcal{Q} \cdot 2^{-40}$ *for each key guess.*

Overall time complexity is approximately $\mathcal{Q} \cdot 2^8$*, much lower than* $\mathcal{Q} \cdot 2^{32}$*, demonstrating the effectiveness of the early abort technique.*

D. Existing key recovery process of IB attacks

As shown in Figure 1, given an r_d-round `(RK-)IB distinguisher` of E_d, attackers add r_b rounds before and r_f rounds after to launch an $(r_b + r_d + r_f)$-round `(RK-)IB attack`. Following [5], [32], the input differences (α, α') and output differences (β, β') are set equal respectively hereafter, i.e., $\alpha = \alpha'$ and $\beta = \beta'$. We focus on the related-key setting with

$$(K_0, K_1, K_2, K_3) = (K_0, K_0 \oplus \Delta K, K_0 \oplus \nabla K \oplus \Delta K, K_0 \oplus \nabla K). \tag{2}$$

Thus, the sets of plaintext and ciphertext differences that may lead to α and β are identical: $\Omega_{in} = \Omega'_{in}$ and $\Omega_{out} = \Omega'_{out}$. Queries are assumed to access the encryption oracle (resp. decryption oracle) without loss of generality.

We now outline the impossible differential style (IDS) and boomerang style (BS) for `IB attack` from [5], [32], and summarize key parameters common to both IDS and BS key recovery:

- The positions of the d_{in} (resp. d_{out}) activated bits in the plaintext (resp. ciphertext) are determined by the truncated differential back-propagation (resp. propagation) from the input difference α (resp. output difference β) and the round key differences of the `(RK-)IB distinguisher`.
- The probabilities $2^{-c_{in}}$ and $2^{-c_{out}}$ are usually derived from bit conditions. For fixed α and β, they equal $1/|\Omega_{in}|$ and $1/|\Omega_{out}|$, so $c_{in} = d_{in}$ and $c_{out} = d_{out}$.
- C_E is the ratio of the cost for partial encryption to full encryption, estimated as the number of nonlinear operations (e.g., S-boxes) in the partial encryption divided by that in the full cipher, for early abort technique.

1) Impossible Differential Style:

-IDS.1: Get plaintext-ciphertext pairs. Construct 2^s plaintext structures, each with $2^{d_{in}}$ plaintexts differing in d_{in} fixed bits. Query ciphertexts for all $2^{s+d_{in}}$ plaintexts under the four related keys as specified in Equation (2). In total, $\mathcal{D} = 2^{2+s+d_{in}}$ plaintext-ciphertext pairs are obtained.

-IDS.2: Get quartets.

-IDS.2a: Build plaintext pairs within each plaintext structure, and obtain $\mathcal{P}$ pairs of $((P_0, C_0), (P_1, C_1))$ under (K_0, K_1) and $\mathcal{P}$ pairs of $((P_3, C_3), (P_2, C_2))$ under (K_2, K_3), where $\mathcal{P} = 2^{s+2d_{in}}$.

-IDS.2b: Build a hash table H_0 indexing $((P_0, C_0), (P_1, C_1))$ by the $(n - d_{out})$ bits of C_0 and C_1 outside Ω_{out}. For each $((P_3, C_3), (P_2, C_2))$, look up matching entries by the $(n - d_{out})$ bits of C_3 and C_2 outside Ω_{out}. This yields $\mathcal{Q} = 2^{2(s+2d_{in})-2(n-d_{out})}$ quartets of $((P_0, C_0), (P_1, C_1), (P_2, C_2), (P_3, C_3))$ satisfying: (P_0, P_1) and (P_2, P_3) have differences in Ω_{in}, (C_0, C_3) and (C_1, C_2) have differences in Ω_{out}.

-IDS.3: Guess K_{in} and K_{out} sequentially.

-IDS.3a: Apply the early abort technique to filter the $\mathcal{Q}$ quartets by guessing K_{in}.

-IDS.3b: Apply the early abort technique to filter the remaining quartets by guessing K_{out}.

-IDS.3c: Discard the key candidates that are consistent with the final remaining quartets.

-IDS.4: Exhaustively search the remaining key candidates.

Complexity. The data complexity is $\mathcal{DC}_{IDS} = 2^{2+s+d_{in}}$. The time complexity $\mathcal{TC}_{IDS}$ comprises five components: $\mathcal{TC}_{IDS} = \mathcal{D} + 2\mathcal{P} + \mathcal{Q} + \mathcal{A} + \mathcal{S}$, where:

- $\mathcal{D} = 2^{2+s+d_{in}}$: cost of data generation.
- $2\mathcal{P} = 2^{s+2d_{in}+1}$: cost of building pairs.
- $\mathcal{Q} = 2^{2(s+2d_{in})-2(n-d_{out})}$: cost of producing quartets.
- $\mathcal{A} = \mathcal{Q} \cdot 2^{|K_{in} \cup K_{out}| - 2(c_{in}+c_{out})} \cdot C_E$: cost of the early abort technique.
- $\mathcal{S} = 2^{|K|}(1 - 2^{-2(c_{in}+c_{out})})^{\mathcal{Q}}$: cost of final exhaustive search (If a quartet causes the input and output differences of the `IB distinguisher`—with probability $2^{-2(c_{in}+c_{out})}$—it can discard a key candidate. Thus, the probability that a key candidate remains is $p = (1 - 2^{-2(c_{in}+c_{out})})^{\mathcal{Q}}$.).

The memory complexity is dominated by storage of data, pairs, quartets, and remaining key candidates: $\mathcal{MC}_{IDS} = \mathcal{D} + \mathcal{P} + \mathcal{Q} + p \cdot \mathcal{K}$, where $\mathcal{K} = 2^{|K_{in} \cup K_{out}|}$. The early abort technique does not require storing keys, so its memory use is bounded by $\mathcal{Q}$.

2) Boomerang Style:

-BS.1: This step is identical to Step **IDS.1**.

-BS.2: For each guess of K_{in}:

-BS.2a: For each plaintext structure, partially encrypt P_0 to the beginning of `IB distinguisher` under K_0, XOR the resulting state with α, then decrypt to obtain the plaintext P_1 under K_1. Retrieve the corresponding ciphertexts (C_0, C_1) from table T_0, T_1. This yields $2^{s+d_{in}}$ pairs $((P_0, C_0), (P_1, C_1))$. Similarly, construct $2^{s+d_{in}}$ pairs $((P_3, C_3), (P_2, C_2))$.

-BS.2b: This step is identical to Step **IDS.2b**. In total, $\mathcal{Q} = 2^{2(s+d_{in})-2(n-d_{out})}$ quartets of $((P_0, C_0), (P_1, C_1), (P_2, C_2), (P_3, C_3))$ are obtained, where (P_0, P_1) and (P_2, P_3) have differences in Ω_{in}, and (C_0, C_3) and (C_1, C_2) have differences in Ω_{out}.

-BS.2c: Apply the early abort technique to filter the remaining quartets by guessing K_{out}.

-BS.2d: Discard the key candidates that are consistent with the final remaining quartets.

-BS.3: Exhaustively search the remaining key candidates.

Complexity. The date complexity is $\mathcal{DC}_{BS} = 2^{2+s+d_{in}}$. The time complexity $\mathcal{TC}_{IDS}$ comprises five components: $\mathcal{TC}_{IDS} = \mathcal{D} + \mathcal{P}' + \mathcal{Q}' + \mathcal{A} + \mathcal{S}$, where:

- $\mathcal{D} = 2^{2+s+d_{in}}$: cost of data generation:.

- $\mathcal{P}' = 2^{|K_{in}|} \times 2\mathcal{P} \times 2|E_b|/|E|$, where $\mathcal{P} = 2^{s+d_{in}}$: cost of building pairs:
- $\mathcal{Q}' = 2^{|K_{in}|} \times \mathcal{Q} = 2^{|K_{in}|+2(s+d_{in})-2(n-d_{out})}$: cost of producing quartets.
- $\mathcal{A} = 2^{|K_{in}|} \times (\mathcal{Q} \times 2^{|K_{out}/K_{in}|-2c_{out}})C_E$: cost of the early abort technique.
- $\mathcal{S} = p \cdot 2^{|K|} = 2^{|K|}(1-2^{-2c_{out}})^{\mathcal{Q}}$: cost of final exhaustive search (If a quartet causes the input and output differences of the `IB distinguisher`—with probability $2^{-2c_{out}}$—it can discard a key candidate. Thus, the probability that a key candidate remains is $p = (1-2^{-2c_{out}})^{\mathcal{Q}}$.).

Similar to the impossible differential style, the memory complexity is $\mathcal{MC}_{BS} = \mathcal{D}+\mathcal{P}+\mathcal{Q}+p\cdot\mathcal{K}$, where $\mathcal{K} = 2^{|K_{in} \cup K_{out}|}$.

III. New techniques for IB Attacks

This section introduces the pre-guessing technique, pre-sieving technique, and automatic key-guessing strategy to optimize the key recovery in `(RK-)IB attack`. As shown in Figure 1, for an SPN block cipher E under the related keys specified in Equation (2), given an r_d-round `(RK-)IB distinguisher` $(\alpha, \alpha, \beta, \beta)$, adding r_b-round E_b before and r_f-round E_f after the distinguisher enables an $(r_b + r_d + r_f)$-round `(RK-)IB attack`.

A. Pre-guessing technique

In `(RK-)IB attack`, the time complexity of constructing quartets significantly impacts the overall attack complexity. When filtering quartets for `RK-IB distinguishers`, valid quartets may suggest key candidates that satisfy specific bit conditions. Inspired by existing key recovery styles, we find that pre-guessing certain key bits before generating quartets may reduce the number of invalid quartets—similar to the full pre-guesses of K_{in} or K_{out} in the Boomerang Style. However, guessing too many key bits at once may lose the benefit from the early abort technique, which may lead to a higher overall complexity. To get better tradeoff and enable more flexible selections, such as guessing certain key bits from both K_{in} and K_{out}, or guessing some difference in the middle to introduce additional bit conditions, we propose the partial key/difference pre-guessing technique. To choose the partial key/difference guessing bits, we constructed a directed graph whose subgraphs correspond to the effective partitions of the guessed bits.

For the first r_b-round, a directed graph $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$ models differential propagation with fixed differences under required key guesses in E_b, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. The vertices take the form $\mathbb{X}_j^r = (IX_{0,j}^r, IX_{3,j}^r, \Delta X_{01,j}^r, \Delta X_{23,j}^r)$, $\mathbb{Y}_j^r = (IY_{0,j}^r, IY_{3,j}^r, \Delta Y_{01,j}^r, \Delta Y_{23,j}^r)$ and $\mathbb{Z}_j^r = (IZ_{0,j}^r, IZ_{3,j}^r, \Delta Z_{01,j}^r, \Delta Z_{23,j}^r)$, with $0 \le r \le r_b - 1, 0 \le j \le t-1$. $\mathcal{E}$ contains edge types: anonymous edges, and key-named edges labeled with associated keys $\mathbb{K}_j^r = IK_{0,j}^r$. To back-propagate the input difference α of the `(RK-)IB distinguisher` along with the round key difference, we define the flag of $\mathbb{X}_j^r$ as follows:

$$f\mathbb{X}_j^r = \begin{cases} 0, & \text{if } \Delta X_{01,j}^r \text{ and } \Delta X_{23,j}^r \text{ are inactive,} \\ 1, & \text{if } \Delta X_{01,j}^r \text{ and } \Delta X_{23,j}^r \text{ are active and known,} \\ 2, & \text{if } \Delta X_{01,j}^r \text{ or } \Delta X_{23,j}^r \text{ is unknown.} \end{cases} \tag{3}$$

Since the input and output differences of the `(RK-)IB distinguisher` and the corresponding key difference are set equal in pairs, $\Delta X_{01,j}^r$ and $\Delta X_{23,j}^r$ are both inactive, both active, both known, or both unknown. Similar flags $f\mathbb{Y}_j^r$ and $f\mathbb{Z}_j^r$ apply to $\mathbb{Y}_j^r$ and $\mathbb{Z}_j^r$. In the `LL` layer, if $\mathbb{Z}_i^r$ influences $\mathbb{X}_j^{r+1}$ via the linear transformation, it is denoted as $\mathbb{Z}_i^r \rightarrowtail \mathbb{X}_j^{r+1}$; if $\mathbb{Z}_i^r$ is influenced by $\mathbb{X}_j^{r+1}$ via the inverse transformation, it is denoted as $\mathbb{Z}_i^r \leftarrowtail \mathbb{X}_j^{r+1}$.

Definition 3. *For r from $r_b - 1$ down to 0, the vertices and edges in $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$ are defined as follows:*

- *`LL` layer: For $0 \le j \le t-1$, if $\mathbb{X}_j^{r+1} \in \mathcal{V}$, add all $\mathbb{Z}_i^r$ with $\mathbb{Z}_i^r \rightarrowtail \mathbb{X}_j^{r+1}$ to $\mathcal{V}$; Otherwise, if there exists $\{\mathbb{Z}_i^r | \mathbb{Z}_i^r \rightarrowtail \mathbb{X}_j^{r+1}, f\mathbb{Z}_i^r = 2\}$, add all such $\mathbb{Z}_i^r$ and $\mathbb{X}_j^{r+1}$ to $\mathcal{V}$. Add edges directed from these $\mathbb{Z}_i^r$ to $\mathbb{X}_j^{r+1}$ to $\mathcal{E}$.*
- *`SL` layer: For $0 \le j \le t-1$, if $\mathbb{Z}_j^r \in \mathcal{V}$, add $\mathbb{Y}_j^r$ to $\mathcal{V}$; Otherwise, if $f\mathbb{Z}_j^r = 1$, add $\mathbb{Z}_j^r$ and $\mathbb{Y}_j^r$ to $\mathcal{V}$. Add an edge directed from $\mathbb{Y}_j^r$ to $\mathbb{Z}_j^r$ to $\mathcal{E}$.*
- *For `AKL`$_{k_r}$ layer: For $0 \le j \le t-1$, if $\mathbb{Y}_j^r \in \mathcal{V}$, add $\mathbb{X}_j^r$ to $\mathcal{V}$. Add an edge directed from $\mathbb{X}_j^r$ to $\mathbb{Y}_j^r$ to $\mathcal{E}$, labeled as $\mathbb{K}_j^r$.*

We present an example to illustrate the construction of the directed graph. First, we define a toy block cipher.

Toy block cipher. As shown in Figure 5, the cipher operates on a state of four 4-bit cells. Each round XORs an independent round key with the state, followed by a S-box layer with four identical 4-bit S-boxes defined by $S = [0, 1, 2, 3, 4, 13, 15, 6, 8, 11, 5, 14, 12, 7, 10, 9]$ (the same as used in `ARADI`), and then a linear layer with a multiplication of matrix $M = [[1, 0, 1, 1], [1, 0, 0, 0], [0, 1, 1, 0], [1, 0, 1, 0]]$ (the MixColumn matrix from `SKINNYe v2`). Assume $(\alpha, \alpha, \beta, \beta)$ is an r_d-round `IB distinguisher` with $\alpha = (1, 0, 0, 0)$ and $\beta = (1, 0, 0, 0)$. We add two rounds before the distinguisher to mount an $(r_d + 2)$-round `IB attack`.

Example 2. *As shown in Figure 5, for the toy block cipher and its `IB distinguisher`, we propagate the input difference two rounds backward. White cells represent inactive differences, green cells represent known active differences, and gray cells represent unknown differences. The directed graph $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$ is built as follows:*

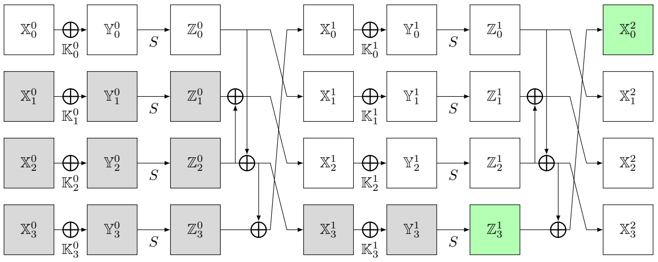


Fig. 5: The differential propagation for the toy block cipher.

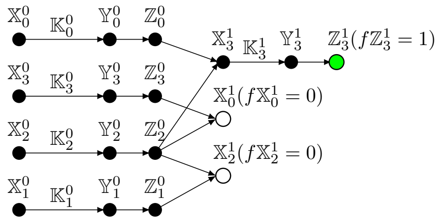


Fig. 6: The directed graph $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$ for the toy block cipher.

1) *Since $f\mathbb{Z}_3^1 = 1$, add $\mathbb{Y}_3^1$ and $\mathbb{Z}_3^1$ to $\mathcal{V}$, and add edge $\mathbb{Y}_3^1 \rightarrowtail \mathbb{Z}_3^1$ to $\mathcal{E}$;*
2) *Since $\mathbb{Y}_3^1$ is in $\mathcal{V}$, add $\mathbb{X}_3^1$ to $\mathcal{V}$, and add edge $\mathbb{X}_3^1 \rightarrowtail \mathbb{Y}_3^1$ to $\mathcal{E}$;*
3) *Since $\mathbb{X}_3^1$ is in $\mathcal{V}$, and $\mathbb{Z}_0^0 \rightarrowtail \mathbb{X}_3^1$, $\mathbb{Z}_2^0 \rightarrowtail \mathbb{X}_3^1$, add $\mathbb{Z}_0^0$, $\mathbb{Z}_2^0$ to $\mathcal{V}$; since $f\mathbb{Z}_1^0 = 2$, $f\mathbb{Z}_3^0 = 2$, and $\mathbb{Z}_1^0 \rightarrowtail \mathbb{X}_2^1$, $\mathbb{Z}_3^0 \rightarrowtail \mathbb{X}_0^1$, add $\mathbb{Z}_1^0, \mathbb{Z}_3^0$ and $\mathbb{X}_2^1, \mathbb{X}_0^1$ to $\mathcal{V}$. Add edges $\mathbb{Z}_0^0 \rightarrowtail \mathbb{X}_3^1$, $\mathbb{Z}_2^0 \rightarrowtail \mathbb{X}_3^1$, $\mathbb{Z}_1^0 \rightarrowtail \mathbb{X}_2^1$, $\mathbb{Z}_3^0 \rightarrowtail \mathbb{X}_0^1$, $\mathbb{Z}_2^0 \rightarrowtail \mathbb{X}_0^1$, $\mathbb{Z}_2^0 \rightarrowtail \mathbb{X}_2^1$ to $\mathcal{E}$.*
4) *Repeat steps 1)-2) and obtain the final directed graph, as shown in Example 4.*

We provide the following definitions for the vertices.

Definition 4. *In $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$, a vertex $\mathbb{V}$ with $f\mathbb{V} = 0$ or $f\mathbb{V} = 1$ is called a sink vertex if there exists a vertex $\mathbb{V}'$ with $f\mathbb{V}' = 2$ such that $\mathbb{V}' \rightarrowtail \mathbb{V}$. A vertex with no incoming edges is called a source vertex. An upper vertex of a given vertex is any vertex that directly or indirectly points to it.*

Let $H = \{j | \mathbb{X}_j^0$ is a source vertex in $\mathcal{G}_b\}$. Then the source vertices in $\mathcal{G}_b$ are exactly $\mathbb{X}_j^0$ for $j \in H$. Additionally, the difference of each sink vertex in $\mathcal{G}_b$ is known. For example, in Example 4, $\mathbb{Z}_3^1$, $\mathbb{X}_0^1$, and $\mathbb{X}_2^1$ are sink vertices, and $\mathbb{X}_i^0$ $(0 \le i \le 3)$ are source vertices. Next, we introduce conditional subgraphs, which enables bit condition derivation in key recovery.

Definition 5. *In $\mathcal{G}_b(\mathcal{V}, \mathcal{E})$, let $\mathbb{V}$ be a sink vertex and $\mathbb{V}'_0, \ldots, \mathbb{V}'_{\tau-1}$ be τ vertices such that $\mathbb{V}'_i \rightarrowtail \mathbb{V}$ and $f\mathbb{V}'_i = 2$. Then, $\mathbb{V}$, $\mathbb{V}'_0, \ldots, \mathbb{V}'_{\tau-1}$ and all their upper vertices, as well as the corresponding directed edges form a conditional subgraph of $\mathcal{G}_b$, denoted as $\mathcal{G}_b^S$.*

For a given $\mathcal{G}_b^S$, denote its sink vertex as $(V_0, V_3, \Delta V_{01}, \Delta V_{23})$. Through partial encryption, we derive relations among all associated keys of $\mathcal{G}_b^S$, the values $IX_{0,j}^0, IX_{1,j}^0, IX_{2,j}^0, IX_{3,j}^0$ (for $j \in H = \{j \mid \mathbb{X}_j^0$ is a source vertex in $\mathcal{G}_b^S\}$), and the differences $\Delta V_{01}, \Delta V_{23}$. Specifically, one relation links the keys, $IX_{0,j}^0$, $IX_{1,j}^0$, and ΔV_{23}; another links the same or related keys, $IX_{2,j}^0$, $IX_{3,j}^0$, and ΔV_{23}. If ΔV_{01} and ΔV_{23} represent ω_{01}-bit and ω_{23}-bit differences respectively, then $\mathcal{G}_b^S$ imposes $(\omega_{01} + \omega_{23})$-bit conditions. A detailed example follows.

Example 3. *In Example 4, the vertices $\mathbb{X}_1^0, \mathbb{X}_2^0, \mathbb{Y}_1^0, \mathbb{Y}_2^0, \mathbb{Z}_1^0, \mathbb{Z}_2^0$, and $\mathbb{X}_2^1$, along with their corresponding edges, form a conditional subgraph. This conditional subgraph enables the derivation of two 4-bit conditions:*

$$\begin{cases} (S(IX_{0,1}^0 \oplus IK_{0,1}^0) \oplus S(IX_{0,2}^0 \oplus IK_{0,2}^0)) \oplus (S(IX_{1,1}^1 \oplus IK_{1,1}^0) \oplus S(IX_{1,2}^1 \oplus IK_{1,2}^0)) = 0, \\ (S(IX_{2,1}^0 \oplus IK_{2,1}^0) \oplus S(IX_{2,2}^0 \oplus IK_{2,2}^0)) \oplus (S(IX_{3,1}^1 \oplus IK_{3,1}^0) \oplus S(IX_{3,2}^1 \oplus IK_{3,2}^0)) = 0. \end{cases}$$

Similarly, for the last r_f rounds, a directed graph $\mathcal{G}_f(\mathcal{V}, \mathcal{E})$ is constructed with the vertices in the form of $\overline{\mathbb{X}}_j^r = (IX_{1,j}^r, IX_{0,j}^r, \Delta X_{12,j}^r, \Delta X_{03,j}^r)$, $\overline{\mathbb{Y}}_j^r = (IY_{1,j}^r, IY_{0,j}^r, \Delta Y_{12,j}^r, \Delta Y_{03,j}^r)$ and $\overline{\mathbb{Z}}_j^r = (IZ_{1,j}^r, IZ_{0,j}^r, \Delta Z_{12,j}^r, \Delta Z_{03,j}^r)$, and the edges classified as anonymous or key-named with $\mathbb{K}_j^r = IK_{0,j}^r$, where $r_b + r_d \le r \le (r_b + r_d + r_f - 1), 0 \le j \le t - 1,$. To propagate the output difference β of the `(RK-)IB distinguisher` along with the round key difference, the flag of $\overline{\mathbb{X}}_j^r$ is defined as:

$$f\overline{\mathbb{X}}_j^r = \begin{cases} 0, \text{ if } \Delta X_{12,j}^r \text{ and } \Delta X_{03,j}^r \text{ are inactive,} \\ 1, \text{ if } \Delta X_{12,j}^r \text{ and } \Delta X_{03,j}^r \text{ are active and known,} \\ 2, \text{ if } \Delta X_{12,j}^r \text{ or } \Delta X_{03,j}^r \text{ is unknown.} \end{cases} \tag{4}$$

Flags $f\overline{\mathbb{Y}}_j^r$, $f\overline{\mathbb{Z}}_j^r$, and $f\overline{\mathbb{K}}_j^r$ are defined analogously for $\overline{\mathbb{Y}}_j^r$, $\overline{\mathbb{Z}}_j^r$, and $\overline{\mathbb{K}}_j^r$. The notations of $\overline{\mathbb{Z}}_i^r \rightarrowtail \overline{\mathbb{X}}_j^{r+1}$ and $\overline{\mathbb{Z}}_i^r \leftarrowtail \overline{\mathbb{X}}_j^{r+1}$ represent forward and backward influences in the `LL` layer, respectively.

Definition 6. *For r from $r_b + r_d$ up to $r_b + r_d + r_f - 1$, the vertices and edges in $\mathcal{G}_f(\mathcal{V}, \mathcal{E})$ are defined as follows:*

- *`AKL`$_{k_r}$ layer: For $0 \le j \le t - 1$, if $\overline{\mathbb{X}}_j^r \in \mathcal{V}$, add $\overline{\mathbb{Y}}_j^r$ to $\mathcal{V}$. Add an edge directed from $\overline{\mathbb{Y}}_j^r$ to $\overline{\mathbb{X}}_j^r$ to $\mathcal{E}$, labeled as $\mathbb{K}_j^r$.*
- *`SL` layer: For $0 \le j \le t - 1$, if $\overline{\mathbb{Y}}_j^r \in \mathcal{V}$, add $\overline{\mathbb{Z}}_j^r$ to $\mathcal{V}$; Otherwise, if $f\overline{\mathbb{Y}}_j^r = 1$, add $\overline{\mathbb{Y}}_j^r$ and $\overline{\mathbb{Z}}_j^r$ to $\mathcal{V}$. Add an edge directed from $\overline{\mathbb{Z}}_j^r$ to $\overline{\mathbb{Y}}_j^r$ to $\mathcal{E}$.*

- `LL` *layer: For* $0 \leq j \leq t-1$, *if* $\overline{\mathbb{Z}}_j^r \in \mathcal{V}$, *add all* $\overline{\mathbb{X}}_i^{r+1}$ *with* $\overline{\mathbb{X}}_i^{r+1} \rightarrowtail \overline{\mathbb{Z}}_j^r$ *to* $\mathcal{V}$*; Otherwise, if there exists* $\{\overline{\mathbb{X}}_i^{r+1} | \overline{\mathbb{Z}}_j^r \rightarrowtail \overline{\mathbb{X}}_i^{r+1}, f\overline{\mathbb{X}}_i^{r+1} = 2\}$, *add all such* $\overline{\mathbb{X}}_i^{r+1}$ *and* $\overline{\mathbb{Z}}_j^r$ *to* $\mathcal{V}$*. Add edges directed from these* $\overline{\mathbb{X}}_i^{r+1}$ *to* $\overline{\mathbb{Z}}_j^r$ *to* $\mathcal{E}$*.*

The definitions of sink vertex, source vertex, upper vertex, and conditional subgraph for $\mathcal{G}_b$ also apply to $\mathcal{G}_f$. The conditional subgraph of $\mathcal{G}_f$ is denoted as $\mathcal{G}_f^S$. Like $\mathcal{G}_b^S$, each $\mathcal{G}_f^S$ also imposes some bit conditions on the ciphertexts.

Partial key pre-guessing technique. During the key recovery process, let l_b conditional subgraphs $\mathcal{G}_{b,i_b}^S (0 \leq i_b \leq l_b - 1)$ and l_f conditional subgraphs $\mathcal{G}_{f,i_f}^S (0 \leq i_f \leq l_f - 1)$ be available for selection. The associated keys of the selected subgraphs can be pre-guessed to perform an `IB attack`. The partial key pre-guessing technique provides a flexible approach: if no subgraphs are selected, it corresponds to **IDS** key recovery; if all $\mathcal{G}_{b,i_b}^S$ of $\mathcal{G}_{f,i_f}^S$ are selected, it corresponds to **BS** key recovery; otherwise, selecting a subset enables identifying attacks that outperform existing methods. In Section V, we apply the partial key pre-guessing technique to `ARADI` and `SKINNYe v2`, demonstrating its effectiveness in improving attack performance.

Partial difference pre-guessing technique. As defined in Equation (3) and Equation (4), the difference of a vertex has three flags: 0, 1, and 2. When a vertex flag is 2, we can infer its difference value, potentially enabling more vertices to have known differences. Combined with partial key pre-guessing, this may introduce more bit conditions and reduce the complexity of constructing quartets. The following example illustrates the benefits of the partial difference pre-guessing technique.

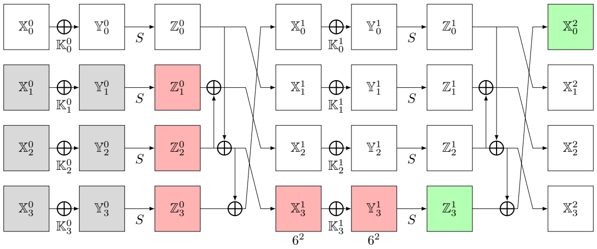


Fig. 7: The partial differences pre-guessing technique for the toy block cipher.

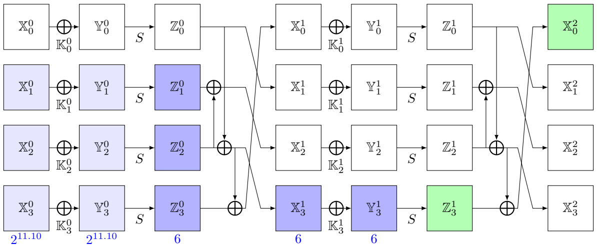


Fig. 8: The pre-sieving technique for the toy block cipher.

Example 4. *Consider the toy block cipher and the* (r_d+2)*-round* `IB attack` *with* 2^s *plaintext structures, where only the leftmost 4 bits are fixed. Thus,* $d_{in} = 12$ *and* $d_{out} = 0, n = 16$*. According to ,* $K_{in} = \{IK_{0,0}^0, IK_{0,1}^0, IK_{0,2}^0, IK_{0,3}^0, IK_{0,3}^1\}$*.*

- *In the* ***IDS*** *key recovery, the quartet generation complexity is* $Q_{IDS}^{toy} = 2^{2(s+2d_{in})-2(n-d_{out})} = 2^{2s+16}$*.*
- *In* ***BS*** *key recovery, the quartet generation complexity is* $Q_{BS}^{toy} = 2^{|K_{in}|+2(s+d_{in})-2(n-d_{out})} = 2^{2s+12}$ *by pre-guessing* $|K_{in}| = 20$ *bits keys.*
- *With the partial key and difference pre-guessing technique: As shown in Figure 7, the S-box properties imply* 6^2 *possible values for* $(\Delta Y_{01,3}^1, \Delta Y_{23,3}^1)$ *that propagate to* $(\Delta Z_{01,3}^1, \Delta Z_{23,3}^1) = (1,1)$*. Guessing all such* $(\Delta Y_{01,3}^1, \Delta Y_{23,3}^1)$ *and* $IK_{0,i}^0$ $(1 \leq i \leq 3)$*, denoted as* K'*, determines the differences of* $\mathbb{X}_3^1$ *and* $\mathbb{Z}_i^0$ $(1 \leq i \leq 3)$ *(highlighted in red). Thus, for any value of* $IX_{0,i}^0$ $(0 \leq i \leq 3)$*, the value of* $IX_{1,i}^0$ $(0 \leq i \leq 3)$ *is uniquely determined, which also holds for* $IX_{2,i}^0$ *and* $IX_{3,i}^0$ $(0 \leq i \leq 3)$*. Therefore, the quartet generation complexity is* $Q_{PD}^{toy} = 6^2 \cdot 2^{|K'|+2(s+d_{in})-2(n-d_{out})} = \cdot 2^{2s+9.17}$*.*

Hence, $Q_{PD}^{toy} < Q_{BS}^{toy} < Q_{IDS}^{toy}$*.*

B. Pre-sieving technique

Current key recovery methods assume that a nonzero output difference of a q-bit S-box can originate from any of the $2^q - 1$ nonzero input differences, leading to $|\Omega_{in}| = 2^{N_a^0 \cdot q}$. However, this ignores specific S-box properties that could further reduce $|\Omega_{in}|$. The core of our pre-sieving technique is to determine the possible propagation difference set using S-box difference patterns as precisely as possible, enabling early elimination of invalid quartets. This technique applies to the first r_b rounds before the `(RK-)IB distinguisher` for encryption oracle queries as detailed subsequently (also applicable to the last r_f rounds after the `(RK-)IB distinguisher` for decryption queries).

Let $\varphi^r = (\varphi_0^r, \ldots, \varphi_{t-1}^r)$ and $\eta^r = (\eta_0^r, \ldots, \eta_{t-1}^r)$ denote the input and output differences of the S-box layer in round r. Using truncated differential propagation, the indexes J^r and the number N_a^r of active S-boxes in round r within E_b are derived from α and the round key differences.

Example 5. *As shown in Figure 5, consider a plaintext structure where only the leftmost 4 bits are fixed and the rest vary freely. For the toy block cipher,* 2^{12} *plaintext differences can propagate to the* `IB distinguisher`*'s input difference* $\alpha = (1,0,0,0)$ *as previous analysis. However, accounting for S-box's difference distribution table reduces this to* $2^{11.10}$*, as shown in Figure 8 (the blue numbers indicate the difference counts).*

Consequently, we propose the pre-sieving technique. For example, when $r_b = 1$, the optimized plaintext difference set Ω_{in}^0 is derived from the S-box's DDT, with $|\Omega_{in}^0| = \prod_{j\in J^0}\mathcal{N}(\eta_j^0)$, where η^0 is determined by α. Since $\mathcal{N}(\eta_j^0) \leq 2^q$, it follows that $|\Omega_{in}^0| \leq |\Omega_{in}|$. Thus, the differences in $\Omega_{in}/\Omega_{in}^0$ cannot propagate to α and can be discarded early to avoid adding invalid quartets. When $r_b > 1$, the set of differences Ω_{in}^r that may lead to α at round r is obtained by back-propagating each difference in Ω_{in}^{r+1} through one round encryption under the corresponding round key difference, for $0 \leq r \leq r_b - 2$. Similarly, $\Omega_{in}^{r_b-1}$ is derived as in the $r_b = 1$ case.

To evaluate the attack's feasibility, we estimate its complexity by determining $|\Omega_{in}^0|$. As described, Ω_{in}^r is iteratively computed to reach a computable intermediate set $\Omega_{in}^{r_{bm}}$, where $0 \leq r_{bm} \leq r_b - 1$. If $r_{bm} = 0$, $|\Omega_{in}^0|$ is obtained directly; otherwise, we need further estimate $|\Omega_{in}^0|$. For S-boxes over $\mathbb{F}_2$, at most 2^{q-1} input differences can propagate to a given output difference. Thus, we amplify $\mathcal{N}$ to 2^{q-1} for all $j \in J^r$ and $0 \leq r \leq r_{bm} - 1$. Consequently, $|\Omega_{in}^0|$ is estimated as $|\Omega_{in}^{r_{bm}}|\prod_{r=0}^{r_{bm}-1}\left(\prod_{j\in J^r} 2^{q-1}\right)$, which is upper bounded by $2^{N_a^0\cdot q}$.

Next, we prove that p_{in}^0, the probability of reaching the differences α from a plaintext difference within Ω_{in}^0, is $1/|\Omega_{in}^0|$.

Theorem 1. *Let $E_k^{r_b}$ denote the r_b-round encryption under the round keys $k = (k^0, \ldots, k^{r_b-1})$, and let Ω_{in}^0 be the set of plaintext differences that back-propagate α through r_b rounds under the round key difference $\Delta k = (\Delta k^0, \ldots, \Delta k^{r_b-1})$ using the pre-sieving technique. For a plaintext pair $(x_0, x_1) \in \{(x, x\oplus\mu)|\mu \in \Omega_{in}^0\}$, the probability that $E_k^{r_b}(x_0) \oplus E_{k\oplus\Delta k}^{r_b}(x_1) = \alpha$ is $1/|\Omega_{in}^0|$.*

Proof. We prove by induction.

Case $r_b = 1$: For a q-bit bijective S-box S and given output difference ν, let $\mathcal{U}$ be the set of input differences that can propagate to ν with $\mathcal{M} = |\mathcal{U}|$. Since $|\{(x, x\oplus\mu)|S(x\oplus k)\oplus S(x\oplus\mu\oplus k) = \nu, \mu\in\mathcal{U}\}| = 2^q$ and $|\{(x, x\oplus\mu)|x\in\mathbb{F}_2^q, \mu\in\mathcal{U}\}| = 2^q\mathcal{M}$, it follows that for $\forall(x_0, x_1) \in \{(x, x\oplus\mu)|x\in\mathbb{F}_2^q, \mu\in\mathcal{U}\}$ and fixed $k\in\mathbb{F}_2^q$, the probability that $S(x_0\oplus k)\oplus S(x_1\oplus k) = \nu$ is $2^q/(2^q\mathcal{M}) = 1/\mathcal{M}$. When $r_b = 1$, similarly for the S-box layer, $p_{in}^0 = 1/(\prod_{j\in J^0}\mathcal{N}(\eta_j^0)) = 1/|\Omega_{in}^0|$.

Case $r_b \geq 2$: Assume for state pair $(x_0, x_1) \in \{(x, x\oplus\mu)|\mu\in\Omega_{in}^1\}$ in round 1, the probability that $E_k^{r_b-1}(x_0) \oplus E_{k\oplus\Delta k}^{r_b-1}(x_1) = \alpha$ is $p_{in}^1 = 1/|\Omega_{in}^1|$. For each $\eta^0 \in \Omega_{in}^1$, there are $\prod_{j\in J^0}\mathcal{N}(\eta_j^0)$ possible plaintext pairs in Ω_{in}^0 that may propagate to η^0, but only one such pair actually reaches η^0 according to the analysis in $r_b = 1$ case. Since no pair can satisfy two different values of η^0 simultaneously, exactly $|\Omega_{in}^1|$ plaintext pairs in Ω_{in}^0 lead to differences in Ω_{in}^1. Based on the conditional probability formula, $p_{in}^0 = |\Omega_{in}^1|/|\Omega_{in}^0| \times 1/|\Omega_{in}^1| = 1/|\Omega_{in}^0|$.

□

For a block cipher, the pre-sieving technique can be applied before the early abort technique to filter quartets in the first r_b rounds. Specifically, we compute Ω_{in}^r from $r_b - 1$ to 0 by back-propagating α under the round key differences, and generate quartets from plaintext pairs based on Ω_{in}^0. Filtering proceeds round-by-round starting from round 0. In each round r ($0 \leq r \leq r_b - 1$), for a given output difference pair (η^r, η'^r) of the S-box layer, we guess all 2^q possible values of $IK_{0,j}^r$ for $j \in J^r$, along with any required unguessed key bits from rounds 0 to $r - 1$ (if $r \geq 1$). Then the corresponding output differences of the S-box layer are then computed, and quartets are filtered according to (η^r, η'^r). The following example illustrates this process. We only outline the workflow here; complexity analysis is discussed in detail later.

Example 6. *For the toy block cipher and its $(r_d + 2)$-round `IB attack`, applying the pre-sieving technique yields $\Omega_{in}^0 = 2^{11.1}$ and $\Omega_{in}^1 = 6$. The key recovery proceeds as follows:*

-1: *For each of $(\Omega_{in}^1)^2 = 36$ possible difference pairs $(\Delta Z_{01,1}^0, \Delta Z_{01,2}^0, \Delta Z_{01,3}^0)$ and $(\Delta Z_{23,1}^0, \Delta Z_{23,2}^0, \Delta Z_{23,3}^0)$, the number of possible plaintext quartets that can propagate to them is $(\mathcal{N}(\Delta Z_{01,1}^0)\mathcal{N}(\Delta Z_{01,2}^0)\mathcal{N}(\Delta Z_{01,3}^0))\cdot(\mathcal{N}(\Delta Z_{23,1}^0)\mathcal{N}(\Delta Z_{23,2}^0)$ $\mathcal{N}(\Delta Z_{23,3}^0))$. Note that $(\Omega_{in}^0)^2$ equals the sum of these products over all such pairs, i.e.,*

$$\sum\nolimits_{(\Delta Z_{01,1}^0,\Delta Z_{01,2}^0,\Delta Z_{01,3}^0),(\Delta Z_{23,1}^0,\Delta Z_{23,2}^0,\Delta Z_{23,3}^0)}[\mathcal{N}(\Delta Z_{01,1}^0)\mathcal{N}(\Delta Z_{01,2}^0)\mathcal{N}(\Delta Z_{01,3}^0)\cdot\mathcal{N}(\Delta Z_{23,1}^0)\mathcal{N}(\Delta Z_{23,2}^0)\mathcal{N}(\Delta Z_{23,3}^0)].$$

Select all possible values of IX_0^0 and IX_3^0, and yield $2^{2\cdot 16}\cdot(\Omega_{in}^0)^2$ plaintext quartets. After verifying that the corresponding ciphertexts satisfy the `IB distinguisher`'s output differences, only $(\Omega_{in}^0)^2$ quartets remain.

-2: *Guess first-round keys:*

-1.1: *Guess $IK_{0,1}^0$ and discard the quartets failing $S(IX_{0,1}^0 \oplus IK_{0,1}^0) \oplus S(IX_{1,1}^0 \oplus IK_{1,1}^0) = \Delta Z_{01,1}^0$ and $S(IX_{2,1}^0 \oplus IK_{2,1}^0) \oplus S(IX_{3,1}^0 \oplus IK_{3,1}^0) = \Delta Z_{23,1}^0$.*

-1.2: *For remaining quartets, guess $IK_{0,2}^0$ and discard the quartets failing $S(IX_{0,2}^0 \oplus IK_{0,2}^0) \oplus S(IX_{1,2}^0 \oplus IK_{1,2}^0) = \Delta Z_{01,2}^0$ and $S(IX_{2,2}^0 \oplus IK_{2,2}^0) \oplus S(IX_{3,2}^0 \oplus IK_{3,2}^0) = \Delta Z_{23,2}^0$.*

-1.3: *For remaining quartets, guess $IK_{0,3}^0$ and discard the quartets failing $S(IX_{0,3}^0 \oplus IK_{0,3}^0) \oplus S(IX_{1,3}^0 \oplus IK_{1,3}^0) = \Delta Z_{01,3}^0$ and $S(IX_{2,3}^0 \oplus IK_{2,3}^0) \oplus S(IX_{3,3}^0 \oplus IK_{3,3}^0) = \Delta Z_{23,3}^0$.*

After this step, one valid quartet remains per difference pair, resulting in 36 quartets total.

- **-3:** *Guess second-round keys: For the 36 remaining quartets, guess $IK^0_{0,0}$ and $IK^1_{0,2}$, and discard quartets failing $S(S(IX^0_{0,0} \oplus IK^0_{0,0}) \oplus IZ^0_{0,2} \oplus IK^1_{0,3}) \oplus S(S(IX^0_{1,0} \oplus IK^0_{1,0}) \oplus IZ^0_{1,2} \oplus IK^1_{1,3}) = \Delta Z^1_{01,3}$ and $S(S(IX^0_{2,0} \oplus IK^0_{2,0}) \oplus IZ^0_{2,2} \oplus IK^1_{2,3}) \oplus S(S(IX^0_{3,0} \oplus IK^0_{3,0}) \oplus IZ^0_{3,2} \oplus IK^1_{3,3}) = \Delta Z^1_{23,3}$.*

For each of the $(\Omega^0_{in})^2$ quartets, the probability that a key is discarded is $(1/\Omega^0_{in})^2$. Thus, the probability that a key survives is $\left(1-(1/\Omega^0_{in})^2\right)^{(\Omega^0_{in})^2} \approx 1/e$. The right key is then identified via exhaustive search among the remaining keys.

C. Automatic key-guessing strategy

Algorithm 1: Automatic key-guessing algorithm

```
1   Input: the number of quartets $\mathcal{Q}$, the set composed of $N_{\mathcal{C}}$ $\varpi_i$-bit conditions $\mathcal{C}_i(k_i, P, C)$ $(0 \le i \le N_{\mathcal{C}} - 1)$
    Output: the time complexity $\mathcal{T}_g$
2   $\mathcal{T}_g := 0$
3   $\mathcal{AK} := deduplicate([k_0, \ldots, k_{N_{\mathcal{C}}-1}])$
4   for $k \in \mathcal{AK}$ do
5       $\varpi_k := 0$
6       $\mathcal{A}_k := [\ ]$
7       for $i \in \{0, \ldots, N_{\mathcal{C}} - 1\}$ do
8           if $k_i = k$ then
9               Add $\mathcal{C}_i(k_i, P, C)$ to $\mathcal{A}_k$
10              $\varpi_k := \varpi_k + \varpi_i$
11  $\mathcal{KN} := [\ ]$
12  while $\mathcal{AK} \neq \emptyset$ do
13      $min\_guess := 10000$
14      $max\_sieve := 0$
15      $gk := \emptyset$
16      for $k \in \mathcal{AK}$ do
17          $l := bitlen(k/\mathcal{KN})$
18          if $l < min\_guess$ then
19              $min\_guess := l$
20              $max\_sieve := \varpi_k$
21              $gk := k$
22          if $l = min\_guess$ and $\varpi_k > max\_sieve$ then
23              $max\_sieve := \varpi_k$
24              $gk := k$
25      $l := bitlen(gk/\mathcal{KN})$
26      $\mathcal{T}_g := \mathcal{T}_g + \mathcal{Q} \cdot 2^l \cdot ratio(\mathcal{A}_{gk}, \mathcal{KN})$
27      $\mathcal{Q} := \mathcal{Q}/2^{\varpi_{gk}}$
28      Remove $gk$ from $\mathcal{AK}$
29      Add $gk$ to $\mathcal{KN}$
```

As stated in Section III-A, after generating the quartets, we filter out incorrect keys using bit conditions from the unused portion of the conditional subgraphs in the pre-guessing phase. The key-guessing order significantly affects the attack complexity, so identifying an optimal order is essential. Prior work has addressed this in other contexts: Hadipour et al. proposed an automatic method for optimal key recovery order in `integral attacks` [17], and Boura et al. introduced a tool for finding optimal guessing sequences in block ciphers with bit-permutation linear layers for `differential attacks` [7]. However, no such method exists for `(RK-)IB attacks`. Therefore, we propose an automatic key-guessing strategy characterized by a prioritization rule that, at each step, selects the key block requiring the fewest guessed bits and enabling the most efficient elimination of invalid quartets. An overview is given in Algorithm 1, followed by a brief explanation.

- Line 3-10: When multiple conditions share the same key blocks, duplicate entries are removed using the “deduplicate” function.
- Lines 13–24: We employ a greedy algorithm to select key blocks for guessing, with priority given first to those requiring fewer key bits and second, in cases of equal bit count, to those that enable the elimination of more

quartets through their associated bit conditions. The list $\mathcal{KN}$ records already-guessed keys; $k/\mathcal{KN}$ denotes the remaining unguessed and required key bits, and *bitlen* computes the number of such bits in a block.

- Line 25-29: For the selected key block and its bit conditions, we compute the time complexity. The term *ratios* represents the ratio of partial encryption/decryption cost in this step to full-round encryption cost, defined similarly to C_E.

IV. Unified Key Recovery Framework of IB Attacks

In this section, we integrate the pre-guessing technique, pre-sieving technique, and automatic key-guessing strategy outlined in Section III to develop a unified key recovery framework (**UF**) for `(RK-)IB attacks`, and formally characterize the overall attack complexity. This framework supports flexible pre-guessing of keys and differences, and incorporates cipher-specific features during key recovery. Importantly, it enables one-pass key recovery without requiring further derivation of the guessing order. Notably, both Impossible Differential Style and Boomerang Style key recovery attacks are specific instances of our framework.

A. *The framework*

For an SPN block cipher E under the related-key setting Equation (2), given an r_d-round `(RK-)IB distinguisher` $(\alpha, \alpha, \beta, \beta)$, an $(r_b + r_d + r_f)$-round `(RK-)IB attack` is constructed by adding r_b rounds before and r_f rounds after the distinguisher. By propagating the input differences backward and the output differences forward, unknown differences are identified. By selecting and pre-guessing partial unknown differences, the directed graphs $\mathcal{G}_b$ and $\mathcal{G}_f$ are constructed. Then the attacker select certain conditional subgraphs from $\mathcal{G}^S_{b,i_b}$ $(0 \le i_b \le l_b - 1)$ and $\mathcal{G}^S_{f,i_f}$ $(0 \le i_f \le l_f - 1)$, and pre-guess their associated keys.

We introduce new notations for key recovery, illustrated in Figure 9 and Figure 10. Figure 9 shows the use of partial pre-guessing (red) and early abort (green); Figure 10 adds the pre-sieving technique (blue). Notably, our automatic key-guessing strategy enables automatic execution of the early abort technique. Let $r_e = r_b + r_d + r_f$.

- $\mathcal{V}_b$ (resp. $\mathcal{V}_f$): the set of source vertices $\mathbb{X}^0_j$ (resp. $\mathbb{Y}^{r_e-1}_j$) in the selected conditional subgraphs from $\mathcal{G}^S_{b,i_b}$ for $0 \le i_b \le l_b - 1$ (resp. $\mathcal{G}^S_{f,i_f}$ for $0 \le i_f \le l_f - 1$).
- D_p: the difference used for pre-guessing.
- Ω_{pin} (resp. Ω_{pout}): the subset of plaintext (resp. ciphertext) differences involved in $\mathcal{V}_b$ (resp. $\mathcal{V}_f$), with $|\Omega_{pin}| = 2^{d_{pin}}$ (resp. $|\Omega_{pout}| = 2^{d_{pout}}$).
- Ω_{rin}, d_{rin} (resp. Ω_{rout}, d_{rout}): Ω_{rin} (resp. Ω_{rout}) denotes the set of the part of plaintext differences (resp. ciphertext differences) not involved in $\mathcal{V}_b$ (resp. $\mathcal{V}_f$), where $d_{rin} = \log_2 |\Omega_{rin}|$ (resp. $d_{rout} = \log_2 |\Omega_{rout}|$).
- Ω_{rin} (resp. Ω_{rout}): the subset of plaintext (resp. ciphertext) differences not involved in $\mathcal{V}_b$ (resp. $\mathcal{V}_f$), with $|\Omega_{rin}| = 2^{d_{rin}}$ (resr. $|\Omega_{rout}| = 2^{d_{rout}}$).
- K_{pin} (resp. K_{pout}): the set of key bits associated with $\mathcal{G}^S_{b,i_b}$ for $0 \le i_b \le l_b - 1$ (resp. $\mathcal{G}^S_{f,i_f}$ for $0 \le i_f \le l_f - 1$).
- K_{rin} (resp. K_{rout}): $K_{rin} = K_{in}/K_{pin}$ (resp. $K_{rout} = K_{out}/K_{pout}$).
- c_{pin} (resp. c_{pout}): the number of bit conditions on (IX^0_0, IX^0_1) or (IX^0_2, IX^0_3) derived from $\mathcal{V}_b$ (resp. $(IY^{r_e-1}_1, IY^{r_e-1}_2)$ or $(IY^{r_e-1}_0, IY^{r_e-1}_3)$ derived from $\mathcal{V}_f$), where IX^0_i (resp. $IY^{r_e-1}_i$) corresponds to the plaintext (resp. ciphertext).
- $2^{-c_{rin}}$ (resp. $2^{-c_{rout}}$): the probability that a plaintext (resp. ciphertext) difference in Ω_{rin} (resp. Ω_{rout}) leads to the input (resp. output) difference α (resp. β) of the `(RK-)IB distinguisher`, given partial values fixed in Ω_{pin} and Ω_{pout}.

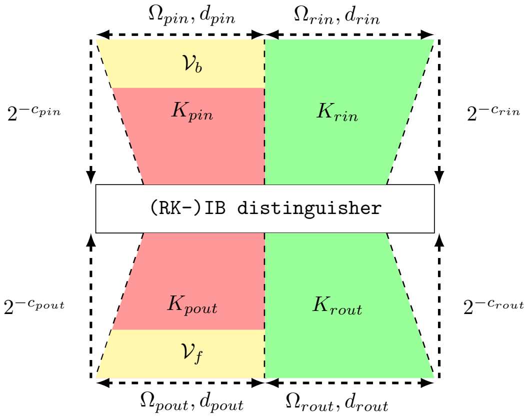


Fig. 9: The key recovery framework without the pre-sieving technique.

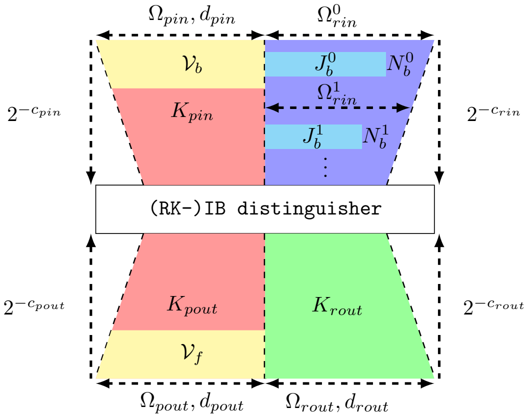


Fig. 10: The key recovery framework with the pre-sieving technique.

- Ω^r_{rin}: the set of input differences linked to K_{rin} that may propagate to the input difference α of the `(RK-)IB distinguisher` in round r under round key differences, derived via the pre-sieving technique; p^r_{in}: the probability of reaching α from the difference within Ω^r_{rin}, for $0 \leq r \leq r_b - 1$.
- N^r_b: the number of active S-boxes in round r not included in the chosen conditional graphs; $J^r_b = \{j^r_0, \ldots, j^r_{N^r_b - 1}\}$: the set of their indices.

With these notations, we present a key recovery framework using the pre-guessing, pre-sieving, and early abort techniques, corresponding to Figure 10. Other key recovery attacks, such as those using only pre-guessing and early abort, can be derived from this framework. Assume the pre-sieving technique is applied to the first r_b rounds (a similar approach applies to the last r_f rounds). We compute Ω^r_{in} $(0 \leq r \leq r_b - 1)$ by back-propagating α under the round key differences, and set $\Omega_{rin} = \Omega^0_{rin}$ with $d_{rin} = \log_2 |\Omega^0_{rin}|$. The attack proceeds as follows:

- **UF.1** (Generate Data): Get plaintext-ciphertext pairs. Identify the d_{in} active plaintext bits using truncated differential propagation rules given the input different α of the `(RK-)IB distinguisher`. Construct 2^s plaintext structures, each with $2^{d_{in}}$ plaintexts. Query ciphertexts for all $2^{s+d_{in}}$ plaintexts under four related keys Equation (2). A total of $\mathcal{D} = 2^{2+s+d_{in}}$ plaintext-ciphertext pairs are required.
- **UF.2**: Pre-guess the difference D_p to determine some unknown differences. Then, pre-guess the key bits in $K_{\text{pin}} \cup K_{\text{pout}}$ to construct quartets as follows:
  - **UF.2a** (Build Tables): Build Tables for pairs satisfying the bit conditions under pre-guessed keys. For each assignment to all bits $\{IX^0_{0,j}\}$ (resp. $\{IX^0_{1,j}\}$) involved in $\mathcal{V}_b$, compute partial encryptions and result in Θ^0 (resp. Θ^1) using pre-guessed keys. Insert $(\{IX^0_{0,j}\}, \Theta^0)$ into hash table H_0, indexed by the bits corresponding to the bit conditions involved in $\mathcal{V}_b$; there are $2^{d_{pin}}$ entries. For each $(\{IX^0_{1,j}\}, \Theta^1)$, find matching entries in H_0 according to the bit conditions, and store $(\{IX^0_{0,j}\}, \{IX^0_{1,j}\})$ in table PT_{01}. Similarly, build tables PT_{23} for $(\{IX^0_{2,j}\}, \{IX^0_{3,j}\})$ involved in $\mathcal{V}_b$, and CT_{12} CT_{03} for $(\{IY^{r_e-1}_{1,j}\}, \{IY^{r_e-1}_{2,j}\})$, $(\{IY^{r_e-1}_{0,j}\}, \{IY^{r_e-1}_{3,j}\})$ involved in $\mathcal{V}_f$.
  - **UF.2b** (Build Pairs): For each plaintext IX^0_0 ($2^{s+d_{in}}$ in total), retrieve $IX^0_{1,j}$s involved in $\mathcal{V}_b$ from PT_{01}, and traverse differences in Ω_{rin} for indices in J^0_b to form IX^0_1. Get their corresponding ciphertexts $(IY^{r_e-1}_0, IY^{r_e-1}_1)$ via lookup tables $T_i, i \in \{0, 1\}$. This yields $\mathcal{P} = |\Omega_{rin}| \cdot 2^{s+(2d_{pin}-c_{pin})+d_{rin}}$ pairs $((IX^0_0, IY^{r_e-1}_0), (IX^0_1, IY^{r_e-1}_1))$. Similarly, generate $\mathcal{P}$ pairs $((IX^0_3, IY^{r_e-1}_3), (IX^0_2, IY^{r_e-1}_2))$.
  - **UF.2c** (Produce Quartets): Build a hash table H_1 storing pairs $((IX^0_0, IY^{r_e-1}_0), (IX^0_1, IY^{r_e-1}_1))$, indexed by the ciphertext bits not in Ω_{out}; for each $((IX^0_3, IY^{r_e-1}_3), (IX^0_2, IY^{r_e-1}_2))$, look up matching entries by the corresponding bits of $IY^{r_e-1}_3$ and $IY^{r_e-1}_2$. For the ciphertext bits j involved in $\mathcal{V}_f$, look up tables CT_{12} and CT_{03} to fix $(\{IY^{r_e-1}_{1,j}\}, \{IY^{r_e-1}_{2,j}\})$, $(\{IY^{r_e-1}_{0,j}\}, \{IY^{r_e-1}_{3,j}\})$. This results in $\overline{\mathcal{Q}} = \mathcal{P}^2 / 2^{2(n-d_{out}+c_{pout})} = |\Omega_{rin}|^2 \cdot 2^{2s+2d_{rin}+4d_{pin}-2c_{pin}+2d_{out}-2n-2c_{pout}}$ quartets, where input pairs have differences in Ω_{in} and output pairs in Ω_{out}, as well as satisfy the bit conditions involved in $\mathcal{V}_b$ and $\mathcal{V}_f$. Since $|D_p|$ bits are pre-guessed, each key candidate corresponds to $\mathcal{Q} = 2^{|D_p|} \cdot \overline{\mathcal{Q}}$ quartets.
  - **UF.2d** (Filter Quartets) For each guess of K_{rin} and K_{rout}, we employ the pre-sieving and early abort techniques to eliminate the invalid quartets.

    -**UF.2dI:** Apply the pre-sieving technique to filter the $\mathcal{Q}$ quartets by guessing K_{rin}. Perform round-by-round filtering starting from round 0, with $\Omega^0_{rin} = \Omega_{rin}$. Let η^r, η'^r be the output differences of the S-box layer in round r $(0 \leq r \leq r_b - 1)$, where $\eta^r, \eta'^r \in \Omega^{r+1}_{rin}$. Let $\mathcal{Q}' = \mathcal{Q} / |\Omega_{rin}|^2$.
    - For each (η^0, η'^0), there are $(\prod_{j \in J^0_b} \mathcal{N}(\eta^0_j)\mathcal{N}(\eta'^0_j))\mathcal{Q}'$ input quartets that may propagate to it. For each $j \in J^0_b$, guess $IK^0_{0,j}$, then derive the S-box output difference and filter quartets using (η^0, η'^0). By Theorem 1, $\mathcal{Q}'$ quartets remain per (η^0_j, η'^0_j).
    - Generally, in round r $(1 \leq r \leq r_b - 1)$, for each (η^r, η'^r), there are $(\prod_{j \in J^r_b} \mathcal{N}(\eta^r_j)\mathcal{N}(\eta'^r_j))\mathcal{Q}'$ such quartets. For each $j \in J^r_b$, guess $IK^r_{0,j}$ and any necessary unguessed key bits from earlier rounds, then derive S-box output difference and filter quartets using (η^r, η'^r). $\mathcal{Q}'$ quartets remain per (η^r_j, η'^r_j), totaling $|\Omega^{r+1}_{rin}|^2 \mathcal{Q}'$ per key guess.
    - Finally, in round $r_b - 1$, $\mathcal{Q}'$ quartets remain per K_{in} guess.

    -**UF.2dII:** Apply the early abort technique to further filter the remaining quartets by guessing K_{rout}.

    Discard the key candidates that are consistent with the final remaining quartets.

-**UF.3:** Exhaustively search the remaining key candidates.

B. The formal complexity

The date complexity is $\mathcal{DC} = 2^{2+s+d_{in}}$. The time complexity $\mathcal{TC}$ consists of six parts:

- Cost of Step **UF.1**: We query the ciphertexts for $2^{s+d_{in}}$ plaintexts under 4 related keys, giving $\mathcal{TC}_1 = 4 \times 2^{s+d_{in}} = 2^{2+s+d_{in}}$.

- Cost of Step **UF.2a**: We construct the tables PT_{01}, PT_{23}, CT_{12}, and CT_{03} under $|K_{\text{pin}} \cup K_{\text{pout}}|$-bit pre-guessed keys and $|D_p|$-bit pre-guessed differences. For PT_{01}, a c_{pin}-bit collision search is performed over $2^{2d_{\text{pin}}}$ possible values of $(IX^0_{0,j}, IX^0_{1,j})$; PT_{23} follows the same construction method. For CT_{12}, a c_{pout}-bit collision search is performed over $2^{2d_{\text{pout}}}$ possible values of $(IY^{r_e-1}_{1,j}, IY^{r_e-1}_{2,j})$; CT_{03} follows the same construction method. Thus, $\mathcal{TC}_{2\text{a}} = 2^{|K_{pin} \cup K_{pout}|+|D_p|} \times 2 \times (2^{2d_{pin}-c_{pin}} + 2^{2d_{pout}-c_{pout}}) \times C^*_E$, where C^*_E is the cost of partial encryption/decryption during table construction.
- Cost of Step **UF.2b**: Under each pre-guessed value, we generate $\mathcal{P}$ pairs $\big((IX^0_0, IY^{r_e-1}_0), (IX^0_1, IY^{r_e-1}_1)\big)$ and $\mathcal{P}$ pairs $\big((IX^0_3, IY^{r_e-1}_3), (IX^0_2, IY^{r_e-1}_2)\big)$. Hence, $\mathcal{TC}_{2\text{b}} = 2^{|K_{\text{pin}} \cup K_{\text{pout}}|+|D_p|} \times 2\mathcal{P}$.
- Cost of Step **UF.2c**: Under each pre-guessed value, we generate $\overline{\mathcal{Q}}$ quartets. Hence, $\mathcal{TC}_{2\text{c}} = 2^{|K_{pin} \cup K_{pout}|+|D_p|} \times \overline{\mathcal{Q}} = 2^{|K_{pin} \cup K_{pout}|} \times \mathcal{Q}$.
- Cost of Step **UF.2d**: Under $|K_{\text{pin}} \cup K_{\text{pout}}|$-bit pre-guessed keys, we apply the pre-sieving technique and early abort technique on $\mathcal{Q}$ quartets. The exact $\mathcal{TC}_{2\text{d}}$ will be analyzed in Section IV-C.
- Cost of Step **UF.3**: If a quartet confirms the input/output differences of the `(RK-)IB distinguisher`-with probability $2^{-2c_{rout}}$-it can discard a key candidate. The probability that a key candidate survives is $p = (1-2^{-2c_{rout}})^{\mathcal{Q}'} = (1-2^{-2c_{rout}})^{\mathcal{Q}/2^{2d_{rin}}}$. Therefore, $\mathcal{TC}_3 = 2^{|K|-2}(1-(1-p)^4)$, as the process of exhaustive search under four related keys is detailed in Section IV-D.

Thus, the time complexity is $\mathcal{TC} = \mathcal{TC}_1 + \mathcal{TC}_{2\text{a}} + \mathcal{TC}_{2\text{b}} + \mathcal{TC}_{2\text{c}} + \mathcal{TC}_{2\text{d}} + \mathcal{TC}_3$. The memory complexity is determined by the storage of data, pairs, quartets, remaining keys, and tables $PT_{01}, PT_{23}, CT_{12}, CT_{03}$: $\mathcal{MC} = \mathcal{D} + \mathcal{P} + \mathcal{Q} + (1-(1-p)^4) \cdot \mathcal{K} + |PT_{01}| + |PT_{23}| + |CT_{12}| + |CT_{03}|$, where $\mathcal{K} = 2^{|K_{in} \cup K_{out}|}$, and a detailed derivation of the term $(1-(1-p)^4) \cdot \mathcal{K}$ is provided in Section IV-D.

C. Detailed discussion of the time complexity for Step ***UF.2d****.*

The time complexity of Step **UF.2d** comes from of two processing techniques: pre-sieving and early aborting. To estimate the pre-sieving cost, we first present a general observation on S-boxes with good cryptographic properties. Then, we analyze the complexity for round 0 (Lemma 2), round r $(1 \le r \le r_b - 2)$ (Lemma 3), and round $r_b - 1$ (Lemma 4).

Lemma 1. *For a q-bit S-box with differential uniformity at most $2^{q/2}$, given two nonzero output differences ν_0 and ν_1, let $\mathcal{U}_0$ and $\mathcal{U}_1$ be the sets of input differences that can propagate to the output differences ν_0 and ν_1, respectively, with $\mathcal{M}_0 = |\mathcal{U}_0|$ and $\mathcal{M}_1 = |\mathcal{U}_1|$. Then, $\mathcal{M}_0\mathcal{M}_1 \ge 2^q$.*

Proof. Since the S-box has differential uniformity at most $2^{q/2}$, we have $\mathcal{M}_0 \ge 2^q/2^{q/2} = 2^{q/2}$ and $\mathcal{M}_1 \ge 2^{q/2}$. Thus, $\mathcal{M}_0\mathcal{M}_1 \ge 2^q$. □

For block ciphers, to resistant the differential attacks, designers usually use S-boxes with small possible differential uniformity. The condition of differential uniformity $\le 2^{q/2}$ is commonly satisfied by practical S-boxes. For example, the S-boxes used in `AES` [11], `Ascon` (final CAESAR portfolio) [13], `MANTIS` (CRYPTO 2016) [1] all meet this criterion. In this paper, we assume this bound when applying pre-sieving techniques to block ciphers. Furthermore, for a block cipher with boxes satisfying

$$\mathcal{M}_0\mathcal{M}_1 > 2^q \text{ for any } \nu_0 \neq 0 \text{ and } \nu_1 \neq 0, \tag{5}$$

the key recovery complexity using the pre-sieving technique can be more precisely estimated, as discussed below.

We list the notations commonly used for formal complexity derivation at the outset.

- C'_E: the ratio of the cost of four S-box operations to that of a full encryption.
- $NK^r_0 (1 \le r \le r_b - 1)$: the number of key bits guessed in K_{rin} in the first $r-1$ rounds; NK^r_1: the number of additional key bits to guess in K_{rin} in the first $r-1$ rounds for the filtering in round r.
- Ψ_f (resp. Ψ_b): the set of i_f (resp. i_b) for which the conditional subgraphs $\mathcal{G}^S_{f,i_f}$ (resp. $\mathcal{G}^S_{b,i_b}$) are not used in pre-guessing, with $|\Psi_f| = l'_f$ (resp. $|\Psi_b| = l'_b$); $\mathcal{K}_{f,i_f}$ (resp. $\mathcal{K}_{b,i_b}$): the set of its associated keys involved in $\mathcal{G}^S_{f,i_f}$ (resp. $\mathcal{G}^S_{b,i_b}$); c_{f,i_f} (resp. c_{b,i_b}): the number of bit conditions involved in $\mathcal{G}^S_{f,i_f}$ (resp. $\mathcal{G}^S_{b,i_b}$), with $2c_{rout} = \sum_{i_f \in \Psi_f} c_{f,i_f}$ (resp. $2c_{rin} = \sum_{i_b \in \Psi_b} c_{b,i_b}$). b and f can be omitted when references are clear.

Lemma 2. *For the key recovery in round 0 of Step* ***UF.2dI****, the time complexity is dominated by $T^0 = N^0_b \cdot T^0_0$, where*

$$T^0_0 = |\Omega^1_{rin}|^2 2^q Q' C'_E \prod_{j \in J^0_b} \mathcal{N}(\eta^0_j)\mathcal{N}(\eta'^0_j).$$

Moreover, if the S-box of the block cipher satisfies Equation (5), then $T^0 = T^0_0$.

Proof. Let $J_b^0 = \{j_0^0, \ldots, j_{N_b^0-1}^0\}$. For each of the $|\Omega_{rin}^1|^2$ values of (η^0, η'^0) and for i from 0 to $N_b^0 - 1$, the cost to process each active S-box sequentially is

$$T_i^0 = (2^q)^{i+1} Q' C'_E \prod_{j \in J_b^0 / \{j_0^0, \ldots, j_{i-1}^0\}} \mathcal{N}(\eta_j^0) \mathcal{N}(\eta_j'^0).$$

By Lemma 1, $T_i^0 \geq T_{i+1}^0$ for $0 \leq i \leq N_b^0 - 2$ and any fixed (η^0, η'^0). Moreover, if the S-box satisfies Equation (5), then $T_i^0 > T_{i+1}^0$. Therefore, the total complexity is dominated by T^0. □

Lemma 3. *For the key recovery in round r of Step **UF.2dI** $(1 \leq r \leq r_b - 2)$, the time complexity is bounded by*

$$T^r = N_b^r \cdot T_0^r, \text{ with } T_0^r = |\Omega_{rin}^{r+1}|^2 2^q 2^{NK_0^r + NK_1^r} Q' C'_E \prod_{j \in J_b^r} \mathcal{N}(\eta_j^r) \mathcal{N}(\eta_j'^r).$$

Moreover, if the S-box of the block cipher satisfies Equation (5), then $T^r = T_0^r$.

Proof. Consider the worst case where, for each (η^r, η'^r), we must guess all NK_1^r key bits when processing each active S-box. Then, for $0 \leq i \leq N_b^r - 1$, the cost is

$$T_i^r = (2^q)^{i+1} 2^{NK_0^r + NK_1^r} Q' C'_E \prod_{j \in J_b^r / \{j_0^r, \ldots, j_{i-1}^r\}} \mathcal{N}(\eta_j^r) \mathcal{N}(\eta_j'^r).$$

By Lemma 1, $T_i^r \geq T_{i+1}^r$ for $0 \leq i \leq N_b^r - 2$ and any fixed (η^r, η'^r). Moreover, if the S-box satisfies Equation (5), it holds $T_i^r > T_{i+1}^r$. Therefore, the total complexity is upper-bounded by T^r. □

In round r_b, since the output difference of the S-box layer is uniquely determined by the input difference α of the `(RK-)IB distinguisher`, there is no need to consider the upper bound in Lemma 3; the exact complexity can be directly determined.

Lemma 4. *For the key recovery in round $r_b - 1$ of Step **UF.2dI**, given a key guessing order $j_{i_0}^{r_b-1}, \ldots, j_{i_\Gamma}^{r_b-1}$ for $J_b^{r_b-1}$ where $\Gamma = N_b^{r_b-1} - 1$, the time complexity is*

$$T^{r_b-1} = \sum_{u=0}^{\Gamma} T_{i_u}^{r_b-1}, \text{ with } T_{i_u}^{r_b-1} = (2^q)^{u+1} 2^{NK_0^{r_b-1} + NK_1^{r_b-1}} Q' C'_E \prod_{j \in J_b^{r_b-1} / \{j_{i_0}^{r_b-1}, \ldots, j_{i_{u-1}}^{r_b-1}\}} \mathcal{N}(\eta_j^{r_b-1}) \mathcal{N}(\eta_j'^{r_b-1}).$$

Here, $NK_1^{r_b-1} = |\cup_{v=0}^{u} \mathcal{K}_{j_{i_v}^{r_b-1}}^S|$, where $\mathcal{K}_{j_i^{r_b-1}}^S$ is the set of additional key bits to guess in K_{rin} in the first $r_b - 2$ rounds for the filtering at the $j_i^{r_b-1}$-th S-box in round $r_b - 1$ for $j_i^{r_b-1} \in J_b^{r_b-1}$.

In the last r_f rounds, we apply the early abort technique to filter the remaining quartets. Recall that we use the conditional subgraphs not used in partial key pre-guessing to derive bit conditions, and the time complexity of key recovery in the last r_f rounds is detailed as follows.

Lemma 5. *For Step **UF.2dII**, given a key guessing order $i_0, \ldots, i_{\Gamma'}$ for Ψ_f where $\Gamma' = l'_f - 1$, the time complexity is*

$$T^{[r_f]} = \sum_{u=0}^{\Gamma'} T_{i_u}^{[r_f]}, \text{ with } T_{i_u}^{[r_f]} = 2^{-(\sum_{v=0}^{u-1} c_{f,i_v})} Q' \cdot 2^{NK^{[r_f]}}.$$

Here, $NK^{[r_f]} = |K_{rin}| + |\cup_{v=0}^{u} \mathcal{K}_{f,i_v}|$. $[r_f]$ denotes spanning across the last r_f rounds.

Based on Lemma 2-Lemma 5, the overall time complexity of filtering the quartets is as follows.

Theorem 2. *Let $j_{i_0}^{r_b-1}, \ldots, j_{i_\Gamma}^{r_b-1}$ be the key guessing order for Step **UF.2dI** in round $r_b - 1$ and $i_0, \ldots, i_{\Gamma'}$ be the key guessing order for Step **UF.2dII**, where $\Gamma = N_b^{r_b-1} - 1$ and $\Gamma' = l'_f - 1$. When applying the pre-sieving technique for the first r_b rounds first, followed by the early abort technique for the last r_f rounds, the time complexity of Step **UF.2d** is*

$$\mathcal{TC}_{2d} = 2^{|K_{pin} \cup K_{pout}|} \times \left(\sum_{r=0}^{r_b-1} T^r + T^{[r_f]} \right),$$

where T^r for $0 \leq r \leq r_b - 1$ and $T^{[r_f]}$ are defined in Lemma 2-Lemma 5.

Next, we consider two variants of the standard key recovery framework. The first variant swaps the order of Step **UF.2dI** and Step **UF.2dII** in the standard key recovery framework, then the overall time complexity of Step **UF.2d** is as follows.

Theorem 3. *Let $j_{i_0}^{r_b-1}, \ldots, j_{i_\Gamma}^{r_b-1}$ be the key guessing order for Step **UF.2dI** in round $r_b - 1$ and $i_0, \ldots, i_{\Gamma'}$ be the key guessing order for Step **UF.2dII**, where $\Gamma = N_b^{r_b-1} - 1$ and $\Gamma' = l'_f - 1$. When applying the early abort technique for the last r_f rounds first, followed by the pre-sieving technique for the first r_b rounds, the time complexity of Step **UF.2d** is $\mathcal{TC}_{2d} = 2^{|K_{pin} \cup K_{pout}|} \times \left(T'^{[r_f]} + \sum_{r=0}^{r_b-1} T'_r \right)$, where $T'^{[r_f]} = \sum_{u=0}^{\Gamma} T'^{[r_f]}_{i_u}$ with $T'^{[r_f]}_{i_u} = 2^{-(\sum_{v=0}^{u-1} c_{f,i_v})} \mathcal{Q} \cdot 2^{|\cup_{v=0}^{u} \mathcal{K}_{f,i_v}|}$, and $T'_r = (2^{|K_{rout}| - 2c_{rout}}) T_r$ for $0 \le r \le r_b - 1$ since K_{rout} is guessed first and the remaining quartets for pre-sieving is reduced by a factor of $2^{2c_{rout}}$.*

The second variant omits the pre-sieving technique from the standard key recovery framework, then the overall time complexity of Step **UF.2d** is as follows.

Theorem 4. *Let $i_0, \ldots, i_{l'-1}$ be the key recovery order for $\Psi_b \cup \Psi_f$ where $\Gamma' = l'_b + l'_f - 1$. When applying only the early abort technique for both the first r_b and last r_f rounds, the time complexity of Step **UF.2d** is*

$$\mathcal{TC}_{2d} = 2^{|K_{pin} \cup K_{pout}|} \times \left(\sum_{u=0}^{l'-1} T_{i_u} \right),$$

where $T_{i_u} = 2^{-(\sum_{v=0}^{u-1} c_{i_v})} \mathcal{Q} \cdot 2^{|\cup_{v=0}^{u} \mathcal{K}_{i_v}|}$.

Discussion: We have now fully characterized the attack complexity. Only the complexity of the quartet filtering step depends on the key-guessing order; those of all other steps remain fixed. When this step dominates the overall complexity, minimizing its cost becomes critical, which requires an optimal key recovery order. Thus, we apply the automatic key-guessing strategy outlined in Section III to determine the globally optimal key guessing order. If this strategy fails, a combinatorial optimization-based search can be used instead. However, this strategy is more efficient than direct combinatorial optimization-based search, and in practice, it consistently yields a sufficiently optimal order. Moreover, this strategy overcomes a inherent limitation of prior attacks: even when an optimal solution is found by bounding with the approximate formula of early abort complexity, previous methods still required manual derivation of detailed recovery steps—an intricate and labor intensive process. Our method automates this derivation.

D. On testing the remaining keys

In the related-key/related-tweakey setting of `RK-IB attacks`, the attacker can query under four keys $(K, K \oplus \Delta K, K \oplus \nabla K, K \oplus \Delta K \oplus \nabla K)$. The exhaustive search complexity is thus reduced from $2^{|K|}$ to $2^{|K|-2}$. The the final exhaustive search complexity in `RK-IB attacks` becomes $2^{|K|-2}(1 - (1-p)^4)$. This important conclusion was first presented by Bonnetain et al. [5]. Here, we provide a more detailed illustration.

The exhaustive search, whose complexity serves as the security upper bound of `RK-IB attacks`, conducts as follows.

1) Query a plaintext P, and obtain the corresponding ciphertexts C_0, C_1, C_2, and C_3 under the keys $K, K \oplus \Delta K, K \oplus \nabla K$, and $K \oplus \Delta K \oplus \nabla K$, respectively.
2) Divide the key space into four non-overlapping parts, as

$$K_i \mid K_i \oplus \Delta K \mid K_i \oplus \nabla K \mid K_i \oplus \Delta K \oplus \nabla K, \ 0 \le i \le 2^{|K|-2} - 1.$$

Encrypt P under K_i for the corresponding ciphertext C', and check for

$$K = \begin{cases} K_i, & \text{if } C' = C_0, \\ K_i \oplus \Delta K, & \text{if } C' = C_1, \\ K_i \oplus \nabla K, & \text{if } C' = C_2, \\ K_i \oplus \Delta K \oplus \nabla K, & \text{if } C' = C_3, \end{cases} \tag{6}$$

to test 4 keys at once. Finally, one i survives and recover K.

The time complexity is $2^{|K|-2}$.

Step **UF.3**, that exhaustively searches the remaining key candidates in `RK-IB attacks`, conducts as follows.

1) Query a plaintext P, and obtain the corresponding ciphertexts C_0, C_1, C_2, and C_3 under the keys $K, K \oplus \Delta K, K \oplus \nabla K$, and $K \oplus \Delta K \oplus \nabla K$, respectively.
2) Divide the key space into four non-overlapping parts, as

$$K_i \mid K_i \oplus \Delta K \mid K_i \oplus \nabla K \mid K_i \oplus \Delta K \oplus \nabla K, \ 0 \le i \le 2^{|K|-2} - 1.$$

Discard the $(K_i \mid K_i \oplus \Delta K \mid K_i \oplus \nabla K \mid K_i \oplus \Delta K \oplus \nabla K)$ that the key bits involved in $K_{in} \cup K_{out}$ of K_i, $K_i \oplus \Delta K$, $K_i \oplus \nabla K$ and $K_i \oplus \Delta K \oplus \nabla K$ are all wrong. Encrypt P under the surviving K_i for the corresponding ciphertext C', and check Equation (6) to test 4 keys at once. Finally, one i survives and recover K.

Since the probability that a key candidate survives prior to the final exhaustive search is p, the probability that the quartet $(K_i \mid K_i \oplus \Delta K \mid K_i \oplus \nabla K \mid K_i \oplus \Delta K \oplus \nabla K)$ is discarded is $(1-p)^4$. Accordingly, the time complexity is $2^{|K|-2}\left(1-(1-p)^4\right)$. Meanwhile, the key bits of surviving quartets $(K_i \mid K_i \oplus \Delta K \mid K_i \oplus \nabla K \mid K_i \oplus \Delta K \oplus \nabla K)$ involved in $K_{in} \cup K_{out}$ need to be stored, while the remaining bits can be recovered via exhaustive search, and thus the memory complexity is $2^{|K_{\text{in}} \cup K_{\text{out}}|}(1-(1-p)^4)$.

V. Applications

In this section, we apply our framework to achieve the full-round attack on ARADI and 34-round attack on SKINNYe v2. The search for (RK-)IB distinguishers was conducted on an AMD @2.60GHz platform with 80.00 GB RAM running 64-bit Ubuntu 18.04, using parallel search with 32 processes to accelerate computation.

A. *Full-round attack on ARADI*

We present the first full-round attack on ARADI, a block cipher designed by the NSA. We construct several 11-round RK-IB distinguishers, then extend two rounds before and three rounds after each to form a 16-round RK-IB attack.

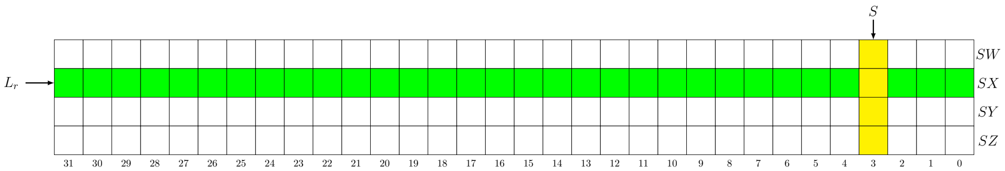


Fig. 11: One round of ARADI.

1) Specifications of ARADI: The block cipher ARADI is a bit-slice cipher based on Toffoli gates, with a 128-bit block size and a 256-bit key size [16]. Its encryption function is defined as:

$$E = \tau_{k_{16}} \circ (\Lambda_{15}\pi\tau_{k_{15}}) \circ \cdots \circ (\Lambda_2\pi\tau_{k_2}) \circ (\Lambda_1\pi\tau_{k_1}) \circ (\Lambda_0\pi\tau_{k_0}).$$

Here,

- π: the S-box layer, operating on four 32-bit words (SW, SX, SY, SZ) that represent the 128-bit state, as

$$SX \leftarrow SX \oplus SW \odot SY,\ SZ \leftarrow SZ \oplus SX \odot SY,\ SY \leftarrow SY \oplus SW \odot SZ,\ SW \leftarrow SW \oplus SX \odot SZ,$$

where $\odot$ denotes bitwise AND. Equivalently, π applies 32 identical 4-bit S-boxes S in parallel. Note that the S-box of ARADI satisfies Equation (5).

- Λ_r $(0 \le r \le 15)$: the r-th linear map. In round r, it transforms the state as

$$\Lambda_r((SW, SX, SY, SZ)) = (L_r(SW), L_r(SX), L_r(SY), L_r(SZ)),$$

where L_r is an involutory operation on a 32-bit word. It splits the input into two 16-bit words (u, v) and computes:

$$(u, v) \to (u \oplus S_{16}^{a_r}(u) \oplus S_{16}^{c_r}(v),\ v \oplus S_{16}^{a_r}(v) \oplus S_{16}^{b_r}(u)),$$

with S_{16}^{p} denoting left-shifting a 16-bit value by p bits, and

$$(a_r, b_r, c_r) = \begin{cases} (11, 8, 14), & r \mod 4 = 0, \\ (10, 9, 11), & r \mod 4 = 1, \\ (9, 4, 14), & r \mod 4 = 2, \\ (8, 9, 7), & r \mod 4 = 3. \end{cases}$$

- τ_{k_r} $(0 \le r \le 15)$: the key addition layer, where the 128-bit round key k_r is XORed with the state.

One round of ARADI is shown in Fig. 11.

The internal state at the r-th step of the key schedule of ARADI is an array of eight 32-bit words $(K_0^r, K_1^r, \ldots, K_7^r)$, and the round key k_r is derived as:

$$k_r = \begin{cases} K_0^r \| K_1^r \| K_2^r \| K_3^r, & r \bmod 2 = 0, \\ K_4^r \| K_5^r \| K_6^r \| K_7^r, & r \bmod 2 = 1. \end{cases}$$

At each step, $K_0^r \| K_1^r$ and $K_4^r \| K_5^r$ are transformed via a 64-bit linear map M_0, while $K_2^r \| K_3^r$ and $K_6^r \| K_7^r$ are processed by another 64-bit linear map M_1. Then, a word-level permutation $P_{r \bmod 2}$ is applied, where $P_0 = (1, 2)(5, 6)$ and $P_1 = (1, 4)(3, 6)$. The maps M_0 and M_1 operate on the 32-bit inputs (a, b) as:

$$M_0((a, b)) = \left(S_{32}^{1}(a) \oplus b,\ S_{32}^{3}(b) \oplus S_{32}^{1}(a) \oplus b\right),$$
$$M_1((a, b)) = \left(S_{32}^{9}(a) \oplus b,\ S_{32}^{28}(b) \oplus S_{32}^{9}(a) \oplus b\right),$$

where S_{32}^{j} denotes a left circular shift by j bits on a 32-bit word.

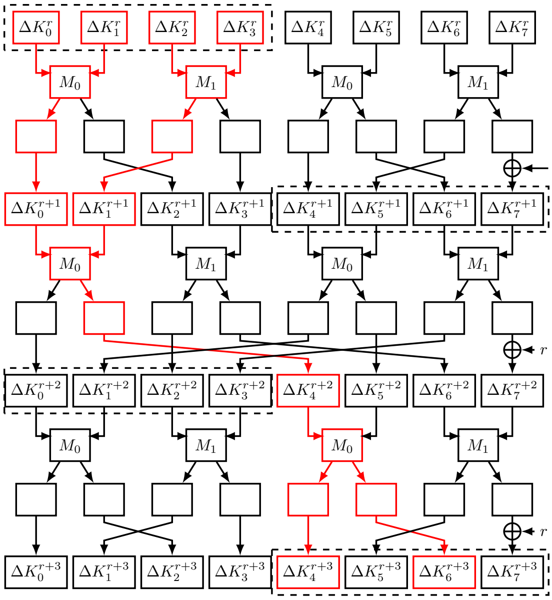

ΔK_0^r ΔK_1^r ΔK_2^r ΔK_3^r ΔK_4^r ΔK_5^r ΔK_6^r ΔK_7^r
M_0
M_1
r
ΔK_0^{r+1} ΔK_1^{r+1} ΔK_2^{r+1} ΔK_3^{r+1} ΔK_4^{r+1} ΔK_5^{r+1} ΔK_6^{r+1} ΔK_7^{r+1}
$r+1$
ΔK_0^{r+2} ΔK_1^{r+2} ΔK_2^{r+2} ΔK_3^{r+2} ΔK_4^{r+2} ΔK_5^{r+2} ΔK_6^{r+2} ΔK_7^{r+2}
$r+2$
ΔK_0^{r+3} ΔK_1^{r+3} ΔK_2^{r+3} ΔK_3^{r+3} ΔK_4^{r+3} ΔK_5^{r+3} ΔK_6^{r+3} ΔK_7^{r+3}

Fig. 12: Type-I 3-round probability-1 related-key differential of ARADI.

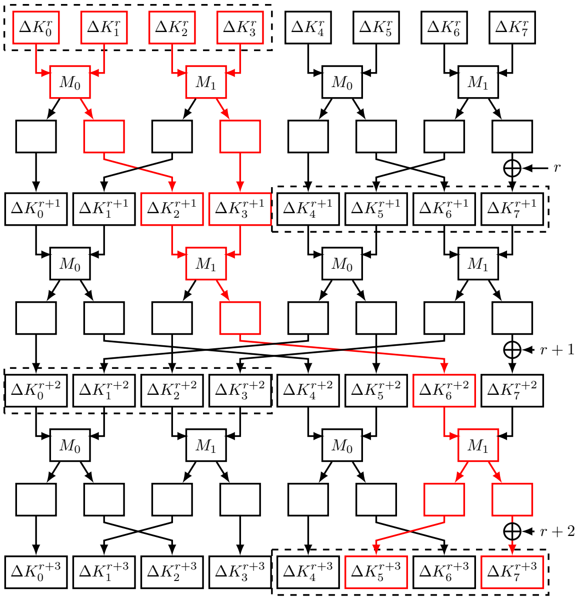

ΔK_0^r ΔK_1^r ΔK_2^r ΔK_3^r ΔK_4^r ΔK_5^r ΔK_6^r ΔK_7^r
M_0
M_1
r
ΔK_0^{r+1} ΔK_1^{r+1} ΔK_2^{r+1} ΔK_3^{r+1} ΔK_4^{r+1} ΔK_5^{r+1} ΔK_6^{r+1} ΔK_7^{r+1}
$r+1$
ΔK_0^{r+2} ΔK_1^{r+2} ΔK_2^{r+2} ΔK_3^{r+2} ΔK_4^{r+2} ΔK_5^{r+2} ΔK_6^{r+2} ΔK_7^{r+2}
$r+2$
ΔK_0^{r+3} ΔK_1^{r+3} ΔK_2^{r+3} ΔK_3^{r+3} ΔK_4^{r+3} ΔK_5^{r+3} ΔK_6^{r+3} ΔK_7^{r+3}

Fig. 13: Type-II 3-round probability-1 related-key differential of ARADI.

2) The 11-round `RK-IB distinguishers` of `ARADI`: The key schedule of ARADI has the following property, enabling the construction of 3-round probability-1 related-key differentials.

Property 1. *For $\forall\kappa \in \mathbb{F}_2^{32}$, let Δk^r be the round key difference in round r. Then $\Delta k^r = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$, $\Delta k^{r+1} = \Delta k^{r+2} = 0$ $\Delta k^{r+3} = (\omega_0, 0, \omega_1, 0)$ for $r \bmod 2 = 0$, where $(\chi_0, 0) = M_0(\lambda_0, \lambda_1)$, $(\chi_1, 0) = M_1(\lambda_2, \lambda_3)$, $(0, \kappa) = M_0(\chi_0, \chi_1)$, and $(\omega_0, \omega_1) = M_0(\kappa, 0)$.*

Moreover, the master key difference ΔK can be derived by propagating $(\lambda_0, \lambda_1, \lambda_2, \lambda_3, 0, 0, 0, 0)$ backward through r rounds.

Proof. As shown in Fig. 12, set $\Delta K_4^{r+2} = \kappa$, then this theorem holds. □

Referring to Fig. 2, let Δx^r denote the difference before key addition in round r, and Δy^r the difference after. Then

$$\Delta x^r = \Delta k^r \to 0 \to 0 \to \Delta k^{r+3} = \Delta y^{r+3}. \tag{7}$$

Thus, $(\Delta k^r, \Delta k^{r+3})$ forms a 3-round probability-1 related-key differential under master key difference ΔK. Similarly, as shown in Fig. 13, another type of such differentials can be constructed for ARADI, with detailed round key differences as follows.

Property 2. *For $\forall\kappa \in \mathbb{F}_2^{32}$, let Δk^r denote the round key difference in round r. Then $\Delta k^r = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$, $\Delta k^{r+1} = \Delta k^{r+2} = 0$ and $\Delta k^{r+3} = (0, \omega_0, 0, \omega_1)$ for $r \bmod 2 = 0$, where $(0, \chi_0) = M_0(\lambda_0, \lambda_1)$, $(0, \chi_1) = M_1(\lambda_2, \lambda_3)$, $(\kappa, 0) = M_0(\chi_0, \chi_1)$, and $(\omega_0, \omega_1) = M_0(\kappa, 0)$.*

Moreover, the master key difference ΔK can be derived by propagating $(\lambda_0, \lambda_1, \lambda_2, \lambda_3, 0, 0, 0, 0)$ backward through r rounds.

Next, we construct the `RK-IB distinguisher` by extending short-round `RK-IB distinguisher` instances using the above 3-round probability-1 related-key differentials as follows.

Construction. Let (α, α_{core}) be a 3-round probability-1 related-key differential under the master key difference ΔK, spanning rounds r_0 to r_0+3, and let (β_{core}, β) be another such differential under the master key difference ∇K, spanning rounds $r_1 - 3$ to r_1, both in the form of Eq. (7), with $r_0 \bmod 2 = 0$ and $(r_1 - 3) \bmod 2 = 0$. If $(\alpha_{core}, \alpha_{core}, \beta_{core}, \beta_{core})$ is an $(r_1 - r_0 - 6)$-round `RK-IB distinguisher` under the master key difference $(\Delta K, \nabla K, \Delta K, \nabla K)$, then $(\alpha, \alpha, \beta, \beta)$ is an $(r_1 - r_0)$-round `RK-IB distinguisher`. By Property 1 and Property 2, α, α_{core}, and ΔK are fully determined by a 32-bit value κ_α, and similarly β, β_{core}, and ∇K by another 32-bit value κ_β. Thus, `RK-IB distinguisher` can be found by traversing κ_α and κ_β.

Result. To maximize the number of rounds that can be added before and after the distinguisher, the input and output differences should have minimal Hamming weight. We therefore set κ_α and κ_β to each contain only one active bit. This reduces the search space to $32 \times 32 \times 4 = 2^{12}$. Using the $\mathcal{HJF}$-method to search for `RK-IB distinguishers` in

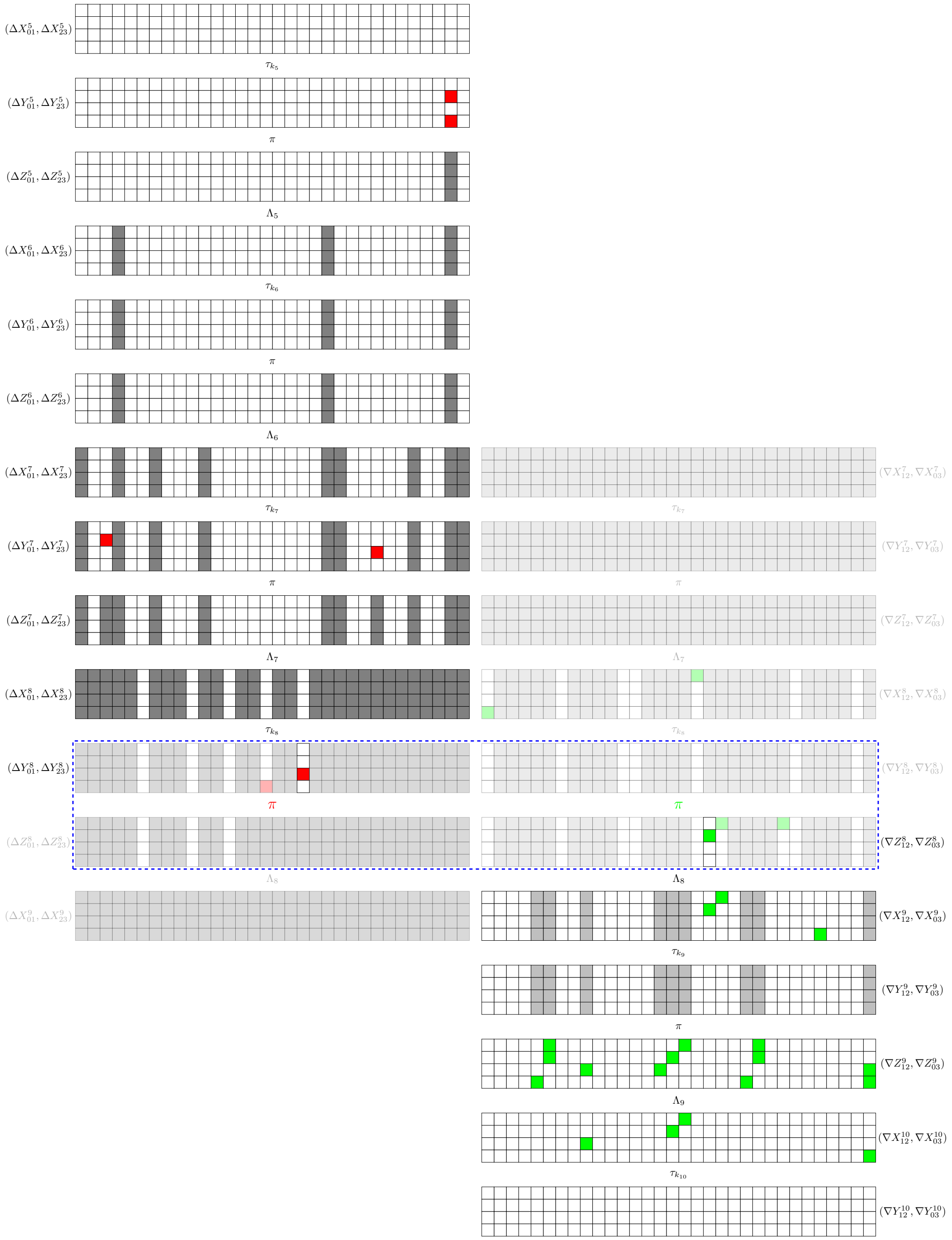

$(\Delta X^5_{01}, \Delta X^5_{23})$
τ_{k_5}
$(\Delta Y^5_{01}, \Delta Y^5_{23})$
π
$(\Delta Z^5_{01}, \Delta Z^5_{23})$
Λ_5
$(\Delta X^6_{01}, \Delta X^6_{23})$
τ_{k_6}
$(\Delta Y^6_{01}, \Delta Y^6_{23})$
π
$(\Delta Z^6_{01}, \Delta Z^6_{23})$
Λ_6
$(\Delta X^7_{01}, \Delta X^7_{23})$
$(\nabla X^7_{12}, \nabla X^7_{03})$
τ_{k_7}
τ_{k_7}
$(\Delta Y^7_{01}, \Delta Y^7_{23})$
$(\nabla Y^7_{12}, \nabla Y^7_{03})$
π
π
$(\Delta Z^7_{01}, \Delta Z^7_{23})$
$(\nabla Z^7_{12}, \nabla Z^7_{03})$
Λ_7
Λ_7
$(\Delta X^8_{01}, \Delta X^8_{23})$
$(\nabla X^8_{12}, \nabla X^8_{03})$
τ_{k_8}
τ_{k_8}
$(\Delta Y^8_{01}, \Delta Y^8_{23})$
$(\nabla Y^8_{12}, \nabla Y^8_{03})$
π
π
$(\Delta Z^8_{01}, \Delta Z^8_{23})$
$(\nabla Z^8_{12}, \nabla Z^8_{03})$
Λ_8
Λ_8
$(\Delta X^9_{01}, \Delta X^9_{23})$
$(\nabla X^9_{12}, \nabla X^9_{03})$
τ_{k_9}
$(\nabla Y^9_{12}, \nabla Y^9_{03})$
π
$(\nabla Z^9_{12}, \nabla Z^9_{03})$
Λ_9
$(\nabla X^{10}_{12}, \nabla X^{10}_{03})$
$\tau_{k_{10}}$
$(\nabla Y^{10}_{12}, \nabla Y^{10}_{03})$

Fig. 14: The core of 11-round `RK-IB distinguisher`. † The grey square denotes an unknown difference bit, the red or green square denotes an active and known difference bit, and the white square denotes an inactive difference bit. The dotted box indicates a contradiction.

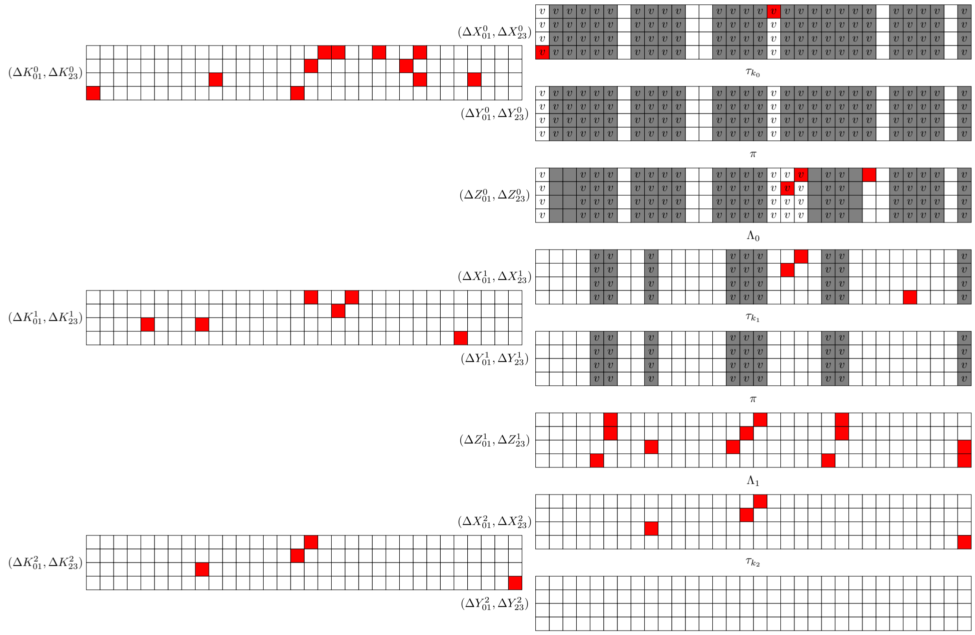

$(\Delta X^0_{01}, \Delta X^0_{23})$
$(\Delta K^0_{01}, \Delta K^0_{23})$
τ_{k_0}
$(\Delta Y^0_{01}, \Delta Y^0_{23})$
π
$(\Delta Z^0_{01}, \Delta Z^0_{23})$
Λ_0
$(\Delta X^1_{01}, \Delta X^1_{23})$
$(\Delta K^1_{01}, \Delta K^1_{23})$
τ_{k_1}
$(\Delta Y^1_{01}, \Delta Y^1_{23})$
π
$(\Delta Z^1_{01}, \Delta Z^1_{23})$
Λ_1
$(\Delta X^2_{01}, \Delta X^2_{23})$
$(\Delta K^2_{01}, \Delta K^2_{23})$
τ_{k_2}
$(\Delta Y^2_{01}, \Delta Y^2_{23})$
v

Fig. 15: Top 2 rounds added for key recovery in full-round attack on `ARADI`. † The white square denotes an inactive difference bit, the red square denotes an active and known difference bit, and the gray square denotes an unknown difference bit. The square labeled v indicates that the bit's state must be known for bit conditions.

`ARADI`, we find 381 11-round `RK-IB distinguishers` in approximately 23.97 hours. One such distinguisher is shown in Distinguisher 1, and its correctness has been verified by manual derivation.

Distinguisher 1. *Let $\alpha = (\alpha_0, \ldots, \alpha_{31}) \in (\mathbb{F}_2^4)^{32}$ be the input difference, $\beta = (\beta_0, \ldots, \beta_{31}) \in (\mathbb{F}_2^4)^{32}$ be the output difference, $\Delta\kappa_r = (\Delta\kappa_{r,0}, \ldots, \Delta\kappa_{r,31}) \in (\mathbb{F}_2^8)^{32}$ be the key difference in round r of the upper trail, and $\nabla\kappa_r = (\nabla\kappa_{r,0}, \ldots, \nabla\kappa_{r,31}) \in (\mathbb{F}_2^8)^{32}$ be that of the lower trail. Then $(\alpha, \alpha, \beta, \beta)$ is an 11-round `RK-IB distinguisher` for `ARADI` under key differences $(\Delta\kappa_0, \Delta\kappa_0, \nabla\kappa_0, \nabla\kappa_0)$, where $\nabla\kappa_0$ is uniquely determined by $\nabla\kappa_{11}$ via the key schedule, and*

$$\begin{cases} \alpha_0 = 8, \alpha_{15} = 1, \alpha_{16} = 2, \alpha_{23} = 4 \textit{ and } \alpha_i = 0 \textit{ for } i \in \mathbb{Z}_{32}/\{0, 15, 16, 23\}, \\ \beta_1 = 10 \textit{ and } \beta_i = 0 \textit{ for } i \in \mathbb{Z}_{32}/\{1\}, \\ \Delta\kappa_{0,0} = 8, \Delta\kappa_{0,15} = 1, \Delta\kappa_{0,16} = 2, \Delta\kappa_{0,23} = 4 \textit{ and } \Delta\kappa_{0,i} = 0 \textit{ for } i \in \mathbb{Z}_{32}/\{0, 15, 16, 23\}, \\ \nabla\kappa_{11,1} = 10 \textit{ and } \nabla\kappa_{11,i} = 0 \textit{ for } i \in \mathbb{Z}_{32}/\{1\}. \end{cases}$$

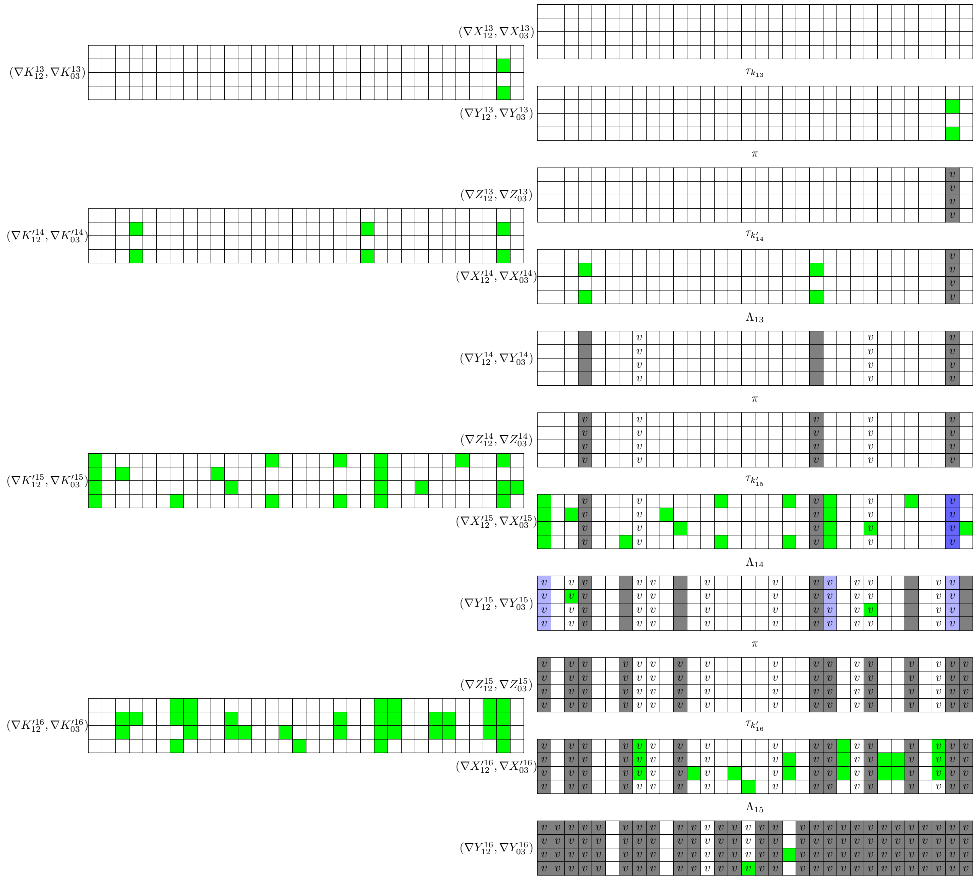
Proof. (Proof by contradiction) Place the distinguisher from Round 2 to Round 13. Suppose $(\alpha, \alpha, \beta, \beta)$ is not an `RK-IB distinguisher` for 11-round `ARADI` under key differences $(\Delta\kappa_0, \Delta\kappa_0, \nabla\kappa_0, \nabla\kappa_0)$. Let α_{core} be a 32-dimensional tuple with $\alpha_{core,1} = 10$ and $\alpha_{core,i} = 0$ for $i \in \mathbb{Z}_{32} \setminus \{1\}$, and let β_{core} be a tuple with $\beta_{core,0} = 8$, $\beta_{core,15} = 1$, $\beta_{core,16} = 2$, $\beta_{core,23} = 4$, and $\beta_{core,i} = 0$ for $i \in \mathbb{Z}_{32} \setminus \{0, 15, 16, 23\}$. By Property 1, both (α, α_{core}) and (β_{core}, β) are 3-round probability-1 related-key differentials. As shown in Figure 14, we have $\Delta X^5_{01} = \Delta X^5_{23} = \alpha_{\text{core}}$ and $\nabla X^{10}_{12} = \nabla X^{10}_{03} = \beta_{\text{core}}$. Propagating $(\Delta X^5_{01}, \Delta X^5_{23})$ forward three rounds and $(\nabla X^{10}_{12}, \nabla X^{10}_{03})$ backward two rounds under the corresponding related keys yields $\Delta Y^8_{01,13} = \Delta Y^8_{23,13} = 4$ and $\nabla Z^8_{12,13} = \nabla Y^8_{03,13} = 2$. Hence, $\text{GEBCT}(4, 4, *, *, *, *, 2, 2) \neq 0$.

However, for the S-box S used in `ARADI`, the following system has no solution:

$$\begin{cases} x_0 \oplus x_1 = 4, \\ x_2 \oplus x_3 = 4, \\ S(x_1) \oplus S(x_2) = 2, \\ S(x_0) \oplus S(x_3) = 2. \end{cases}$$

Therefore, by the definition of GEBCT, it follows that $\text{GEBCT}(4, 4, *, *, *, *, 2, 2) = 0$, which is a contradiction. □

3) Full-round related-key impossible boomerang attack on `ARADI`: We add two rounds before and three rounds after the 11-round `RK-IB distinguisher` from Distinguisher 1 to launch a full-round attack on `ARADI`. The attack overview is shown in Figure 10. Three techniques are used: pre-guessing, pre-sieving, and early abort. Meanwhile, the automatic key-guessing strategy is applied to determine the optimal key recovery order for early abort.



$(\nabla X_{12}^{13}, \nabla X_{03}^{13})$
$(\nabla K_{12}^{13}, \nabla K_{03}^{13})$
$\tau_{k_{13}}$
$(\nabla Y_{12}^{13}, \nabla Y_{03}^{13})$
π
$(\nabla Z_{12}^{13}, \nabla Z_{03}^{13})$
$(\nabla K_{12}'^{14}, \nabla K_{03}'^{14})$
$\tau_{k'_{14}}$
$(\nabla X_{12}'^{14}, \nabla X_{03}'^{14})$
Λ_{13}
$(\nabla Y_{12}^{14}, \nabla Y_{03}^{14})$
π
$(\nabla Z_{12}^{14}, \nabla Z_{03}^{14})$
$(\nabla K_{12}'^{15}, \nabla K_{03}'^{15})$
$\tau_{k'_{15}}$
$(\nabla X_{12}'^{15}, \nabla X_{03}'^{15})$
Λ_{14}
$(\nabla Y_{12}^{15}, \nabla Y_{03}^{15})$
π
$(\nabla Z_{12}^{15}, \nabla Z_{03}^{15})$
$(\nabla K_{12}'^{16}, \nabla K_{03}'^{16})$
$\tau_{k'_{16}}$
$(\nabla X_{12}'^{16}, \nabla X_{03}'^{16})$
Λ_{15}
$(\nabla Y_{12}^{16}, \nabla Y_{03}^{16})$

Fig. 16: Bottom 3 rounds added for key recovery in full-round attack on ARADI. † The white square denotes an inactive difference bit, the green square denotes an active and known difference bit, and the gray square denotes an unknown difference bit. The blue square denotes a pre-guessed difference bit, and the pale blue square denotes a determined difference bit by the pre-guessed bits. The square labeled v indicates that the bit's state must be known for bit conditions.

TABLE II: The known differences and the key nibbles to guess in E_f for ARADI.

Known difference	Keys to guess
$\nabla X_{12,0}'^{15}, \nabla X_{03,0}'^{15}$	$IK_{0,0}'^{16}, IK_{0,28}'^{16}$
$\nabla X_{12,4}'^{15}, \nabla X_{03,4}'^{15}$	$IK_{0,4}'^{16}, IK_{0,11}'^{16}$
$\nabla X_{12,21}'^{15}, \nabla X_{03,21}'^{15}$	$IK_{0,21}'^{16}, IK_{0,28}'^{16}$
$\nabla X_{12,25}'^{15}, \nabla X_{03,25}'^{15}$	$IK_{0,11}'^{16}, IK_{0,25}'^{16}$
$\nabla X_{12,11}'^{14}, \nabla X_{03,11}'^{14}$	$IK_{0,1}'^{15}, IK_{0,11}'^{15}, IK_{0,1}'^{16}, IK_{0,2}'^{16}, IK_{0,8}'^{16}, IK_{0,11}'^{16}, IK_{0,23}'^{16}, IK_{0,29}'^{16}$
$\nabla X_{12,28}'^{14}, \nabla X_{03,28}'^{14}$	$IK_{0,1}'^{15}, IK_{0,28}'^{15}, IK_{0,1}'^{16}, IK_{0,8}'^{16}, IK_{0,14}'^{16}, IK_{0,19}'^{16}, IK_{0,28}'^{16}, IK_{0,29}'^{16}$
$\nabla Y_{12,1}'^{13}, \nabla Y_{03,1}'^{13}$	$IK_{0,1}'^{14}, IK_{0,1}'^{15}, IK_{0,7}'^{15}, IK_{0,24}'^{15}, IK_{0,1}'^{16}, IK_{0,7}'^{16}, IK_{0,8}'^{16}, IK_{0,10}'^{16}, IK_{0,14}'^{16}, IK_{0,19}'^{16}, IK_{0,24}'^{16}, IK_{0,29}'^{16}, IK_{0,31}'^{16}$

Specifically, we pre-guess the 8-bit differences $\nabla X_{12,1}'^{15}$ and $\nabla X_{03,1}'^{15}$, the 12-bit key $IK_{0,j}'^{0}$ ($j \in \{7, 12, 13\}$), and the 20-bit key $K_{0,j}^{16}$ ($j \in \{1, 7, 10, 29, 31\}$). The pre-sieving technique and early abort technique with the automatic key-guessing strategy are applied in the first two rounds and in the last three rounds, respectively. Bit conditions are shown in Table II. The total data, time, and memory complexities sum to less than 2^{254}, confirming an effective full-round attack. The differential propagation in the top two rounds and bottom three rounds are illustrated in Fig. 15 and Fig. 16, respectively. Note that we equivalently swap τ_{k_i} and Λ_{i-1} for $i = 14, 15, 16$ in the bottom three rounds; the resulting difference, state, and key are marked with $'$.

We introduce notations for the detailed exposition of the attack on ARADI. Given specific values $\Delta K_{01}^0 = \Delta K_{23}^0$,

$\Delta K^1_{01} = \Delta K^1_{23}$, and $\Delta Z^1_{01} = \Delta Z^1_{23}$, define:

$$\begin{aligned}
\mathcal{S}_{Y_1} &= \{\epsilon | \epsilon = \pi^{-1}(x) \oplus \pi^{-1}(x \oplus \Delta Z^1_{01}), x \in \mathbb{F}_2^{128}\},\\
\mathcal{S}_{X_1} &= \{\theta | \theta = \epsilon \oplus \Delta K^1_{01}, \epsilon \in \mathcal{S}_{Y_1}\},\\
\mathcal{S}_{Z_0} &= \{\lambda | \lambda = \Lambda_0(\theta), \theta \in \mathcal{S}_{X_1}\},\\
\mathcal{S}_{Y_0} &= \{\sigma | \sigma = \pi^{-1}(x) \oplus \pi^{-1}(x \oplus \overline{\lambda}), x \in \mathbb{F}_2^{128}, \overline{\lambda} = (\overline{\lambda}_{31}, \ldots, \overline{\lambda}_0) \text{ where}\\
&\qquad \overline{\lambda}_i = 0 \text{ for } i \in I = \{7, 12, 13\} \text{ and } \overline{\lambda}_i = \lambda_i \text{ for } i \in \mathbb{Z}_{32}/I, \lambda \in \mathcal{S}_{Z_0}\},\\
\mathcal{S}_{X_0} &= \{\omega | \omega = \sigma \oplus \Delta K^0_{01}, \sigma \in \mathcal{S}_{Y_0}\}.
\end{aligned}$$

Then $|\mathcal{S}_{Z_0}| = |\mathcal{S}_{X_1}| = |\mathcal{S}_{Y_1}| = \prod_{j \in J^1_b} \mathcal{N}(\Delta Z^1_{01,j})$, where $J^1_b = \{0, 10, 15, 16, 17, 23, 27, 9, 26\}$, and $|\mathcal{S}_{X_0}| = |\mathcal{S}_{Y_0}| = \sum_{\gamma \in \mathcal{S}_{Z_0}} \prod_{j \in J^0_b} \mathcal{N}^0_j(\gamma_j)$, where $J^0_b = \{0, 2, 3, 4, 5, 8, 9, 10, 11, 15, 16, 17, 18, 21, 22, 23, 24, 26, 27, 28, 29, 30\}$. Additionally, we obtain $|\mathcal{S}_{X_0}| \approx 2^{80.07}$.

The entire attack proceeds as follows.

- 1 (Generate Data): For all 2^n plaintexts P, obtain the corresponding ciphertexts (C_0, C_1, C_2, C_3) under four related keys $(K, K \oplus \Delta K, K \oplus \Delta K \oplus \nabla K, K \oplus \nabla K)$, where ΔK and ∇K are initial key differences derived from Distinguisher 1. Partially decrypt to compute $IX'^{16}_i = \Lambda^{-1}_{15}(C_i)$ for $0 \leq i \leq 3$, and denote the sets of plaintext-ciphertext pairs as $T_i = \{(P, IX'^{16}_i)\}$ for $0 \leq i \leq 3$.
- 2: Pre-guess the 8-bit differences $\nabla X'^{15}_{12,1}$ and $\nabla X'^{15}_{03,1}$, and pre-guess the 32-bit keys $IK^0_{0,j}$ $(j \in \{7, 12, 13\})$ and $K^{16}_{0,j}$ $(j \in \{1, 7, 10, 29, 31\})$, thus determine the differences $(\nabla Y'^{15}_{12,j}, \nabla Y'^{15}_{03,j})$ for $j \in \{1, 10, 31\}$. Then, construct the quartets as follows:
  - 2a (Build Tables): For $IX^0_{0,j}$ and $IX^0_{1,j}$ $(j = 7, 12, 13)$, perform partial encryption using the guessed keys and verify:

    $$S(IX^0_{0,j} \oplus IK^0_{0,j}) \oplus S(IX^0_{1,j} \oplus IK^0_{0,j} \oplus \Delta K^0_{01,j}) = \Delta Z_{01,j}.$$

    Store $(\{IX^0_{0,j} | j = 7, 12, 13\}, \{IX^0_{1,j} | j = 7, 12, 13\})$ in table PT_{01}. Similarly, for $IX'^{16}_{1,j}$ and $IX'^{16}_{2,j}$ $(j = 1, 7, 10, 29, 31)$, since the differences $(\nabla Y'^{15}_{12,j}, \nabla Y'^{15}_{03,j})$ for $j \in \{1, 10, 31\}$ are determined by pre-guessing $\nabla X'^{15}_{12,1}$ and $\nabla X'^{15}_{03,1}$, perform partial decryption using the pre-guessed keys and verify:

    $$S(IX'^{16}_{1,j} \oplus IK^{16}_{0,j} \oplus \Delta K^0_{01,j}) \oplus S(IX'^{16}_{2,j} \oplus IK^{16}_{0,j} \oplus \Delta K^0_{01,j} \oplus \nabla K^0_{12,j}) = \nabla Y'^{15}_{12,j}.$$

    Store $(\{IX'^{16}_{1,j} | j = 1, 7, 10, 29, 31\}, \{IX'^{16}_{2,j} | j = 1, 7, 10, 29, 31\})$ in table CT_{12}. Construct tables PT_{23} and CT_{03} analogously.
  - 2b (Build Pairs): For all possible values of IX^0_0, generate IX^0_1 by: retrieving $IX^0_{1,j}$ $(j = 7, 12, 13)$ from table PT_{01}; setting $IX^0_{1,j} = IX^0_{0,j}$ for $j = 1, 6, 14, 19, 20, 25, 31$; and setting $IX^0_{1,j} = IX^0_{0,j} \oplus \omega_j$ for remaining j and all differences $\omega \in \mathcal{S}_{X_0}$. Generate the plaintext-ciphertext pairs using table T_i $(i = 0, 1)$, and get $\mathcal{P} = 2^n |\mathcal{S}_{X_0}| = 2^{208.07}$ pairs of $((IX^0_0, IX'^{16}_0), (IX^0_1, IX'^{16}_1))$. Similarly get pairs for indices 2 and 3. Note that these pairs depend only on $IK^0_{0,j}$ $(j \in \{7, 12, 13\})$; thus, they can be reused across other key guesses.
  - 2c (Produce Quartets): Build a hash table H_1 storing pairs $((IX^0_0, IX'^{16}_0), (IX^0_1, IX'^{16}_1))$ indexed by the bits of IX'^{16}_0 and IX'^{16}_1 corresponding to the inactive nibbles of $\nabla X'^{16}_{12}, \nabla X'^{16}_{03}$; for each $\big((IX^0_3, IX'^{16}_3), (IX^0_2, IX'^{16}_2)\big)$, look up matching entries by the corresponding bits of IX'^{16}_3 and IX'^{16}_2. For $j = 1, 7, 10, 29, 31$, look up tables CT_{12} and CT_{03} to fix $(\{IX'^{16}_{1,j} | j = 1, 7, 10, 29, 31\}, \{IX'^{16}_{2,j} | j = 1, 7, 10, 29, 31\})$, $(\{IX'^{16}_{0,j} | j = 1, 7, 10, 29, 31\}, \{IX'^{16}_{3,j} | j = 1, 7, 10, 29, 31\})$. This yields $\overline{\mathcal{Q}} = |\mathcal{S}_{X_0}|^2 \cdot 2^{2d_{\text{rout}}} = 2^{208.14}$ quartets, with $d_{\text{rout}} = 4 \cdot (11 - 5)$. Total quartets per key candidate are $Q = 2^8 \cdot \overline{\mathcal{Q}} = 2^{216.14}$ after accounting for the two 4-bit pre-guessed differences.
  - 2d (Filter Quartets): For each guess of K_{rin} and K_{rout}, eliminate the invalid quartets as follows.
    - 2dI: Apply the pre-sieving technique in the first two rounds. Perform round-by-round filtering, with $\Omega^0_{rin} = \mathcal{S}_{X_0}$. Let $\mathcal{Q}' = \mathcal{Q}/|\Omega_{rin}|^2 = 2^{2d_{rout}+8}$ and $\rho = 4/(32 \cdot 16)$ for the basic cost of partial quartet encryption/decryption.
      - 2dI1 (For $r = 0$): Guess the keys for the remaining active differences in the first round. For $\epsilon, \epsilon' \in \mathcal{S}_{Y_1}$, there are $\mathcal{Q}' \prod_{j \in J^0_b} \mathcal{N}(\lambda_j)\mathcal{N}(\lambda'_j)$ quartets that may propagate to (ϵ, ϵ'), where $\lambda = \Lambda_0(\epsilon \oplus \Delta K^1_{01})$, $\lambda' = \Lambda_0(\epsilon' \oplus \Delta K^1_{01})$ and $J^0_b = \{j^0_0, \ldots, j^0_{21}\} = \{0, 2, 3, 4, 5, 8, 9, 10, 11, 15, 16, 17, 18, 21, 22, 23, 24, 26, 27, 28, 29, 30\}$. Let $J^{0,i}_b = J^0_b / \{j^0_0, \ldots, j^0_{i-1}\}$ for $1 \leq i \leq 22$. Perform Step 2dI1.i for $i \in \{1, 2, \ldots, 22\}$ successively as follows.
        - 2dI1.i: Let $q = j^0_{i-1}$. Guess 2^4 values of $IK^0_{0,p}$, and partially encrypt $(IX^0_{0,q}, IX^0_{1,q}, IX^0_{2,q}, IX^0_{3,q})$ through one S-box. Use the known difference (λ_q, λ'_q) to filter the quartets. The number of remaining quartets is approximately

          $$\frac{1}{\mathcal{N}(\lambda_p)\mathcal{N}(\lambda'_p)} \cdot \mathcal{Q}' \prod_{j \in J^{0,i-1}_b} \mathcal{N}(\lambda_j)\mathcal{N}(\lambda'_j) = \mathcal{Q}' \prod_{j \in J^{0,i}_b} \mathcal{N}(\gamma_j)\mathcal{N}(\gamma'_j).$$

The time complexity is

$$2^{32} \cdot (2^4)^i \cdot \rho \cdot \mathcal{Q}' \prod_{j \in J_b^{0,i-1}} \mathcal{N}(\lambda_j)\mathcal{N}(\lambda'_j) = 2^{33+2d_{rout}+4i} \prod_{j \in J_b^{0,i-1}} \mathcal{N}(\lambda_j)\mathcal{N}(\lambda'_j). \tag{8}$$

After all 22 steps, for each possible (ϵ, ϵ') and key guess $\{IK^0_{0,q}, q \in J^0_b\}$, about $\mathcal{Q}'$ quartets remain. Since there are $\prod_{j \in J_b^1} (\mathcal{N}(\Delta Z^1_{01,j}))^2$ such (ϵ, ϵ') pairs, where $J^1_b = \{0, 10, 15, 16, 17, 23, 27, 9, 26\}$, the total number of remaining quartets per key guess is $\mathcal{Q}' \prod_{j \in J_b^1} (\mathcal{N}(\Delta Z^1_{01,j}))^2$.

- 2dI2 (For $r = 1$): Guess remaining keys in the first two rounds. Let $J^1_b = \{j^1_0, \ldots, j^1_8\} = \{0, 10, 15, 16, 17, 23, 27, 9, 26\}$ and define $J^{1,i}_b = J^1_b / \{j^1_0, \ldots, j^1_{i-1}\}$ for $1 \le i \le 9$. Based on ARADI's linear layer, for $i \in \{1, 2, \ldots, 9\}$ and $q = j^1_{i-1}$, derive

$$IX^1_{l,q} = \bigoplus_{\mu \in P_{i-1}} IZ^0_{l,\mu}$$

for $l = 0, 1, 2, 3$, where $P_0 = \{0, 5, 24\}$, $P_1 = \{10, 15, 18\}$, $P_2 = \{4, 15, 23\}$, $P_3 = \{2, 16, 21\}$, $P_4 = \{3, 17, 22\}$, $P_5 = \{9, 23, 28\}$, $P_6 = \{13, 16, 27\}$. Perform Step 2dI2.i for $i \in \{1, 2, \ldots, 7\}$ successively as follows.

  - 2dI2.i: Let $q = j^1_{i-1}$. Guess 2^4 values of $IK^1_{0,q}$, and partially encrypt $(IX^1_{0,q}, IX^1_{1,q}, IX^1_{2,q}, IX^1_{3,q})$ through one S-box. Use the known difference $(\Delta Z^1_{01,q}, \Delta Z^1_{01,q})$ to filter the quartets. The number of remaining quartets is approximately

$$\frac{1}{\mathcal{N}(\Delta Z^1_{01,j})^2} \cdot \mathcal{Q}' \prod_{j \in J_b^{1,i-1}} \mathcal{N}(\Delta Z^1_{01,j})^2 = \mathcal{Q}' \prod_{j \in J_b^{1,i}} \mathcal{N}(\Delta Z^1_{01,j})^2.$$

  The time complexity is

$$2^{120} \cdot (2^4)^i \cdot \rho \cdot \mathcal{Q}' \prod_{j \in J_b^{1,i-1}} \mathcal{N}(\Delta Z^1_{01,j})^2 = 2^{121+2d_{rout}+4i} \prod_{j \in J_b^{1,i-1}} \mathcal{N}(\Delta Z^1_{01,j})^2. \tag{9}$$

After such 7 steps, proceed with Step 2dI1.i for $i \in \{8, 9\}$ as follows.

  - 2dI2.8: Guess 2^8 values of $IK^0_{0,14}, IK^1_{0,9}$, and partially encrypt $(IX^0_{0,14}, IX^0_{1,14}, IX^0_{2,14}, IX^0_{3,14})$ and $(IX^1_{0,9}, IX^1_{1,9}, IX^1_{2,9}, IX^1_{3,9})$ through one S-box. Use the known difference $(\Delta Z^1_{01,9}, \Delta Z^1_{01,9})$ to filter the quartets. The number of remaining quartets is approximately

$$\frac{1}{\mathcal{N}(\Delta Z^1_{01,9})^2} \cdot \mathcal{Q}' \prod_{j \in J_b^{1,7}} \mathcal{N}(\Delta Z^1_{01,j})^2 = \mathcal{Q}'\mathcal{N}(\Delta Z^1_{01,26})^2.$$

  The time complexity is

$$2^{148} \cdot (2^8)^i \cdot 2\rho \cdot \mathcal{Q}' \prod_{j \in J_b^{1,7}} \mathcal{N}(\Delta Z^1_{01,j})^2 = 2^{158+2d_{rout}} \prod_{j \in J_b^{1,7}} \mathcal{N}(\Delta Z^1_{01,j})^2.$$

  - 2dI2.9: Guess 2^8 values of $IK^0_{0,31}, IK^1_{0,26}$, and partially encrypt $(IX^0_{0,31}, IX^0_{1,31}, IX^0_{2,31}, IX^0_{3,31})$ and $(IX^1_{0,26}, IX^1_{1,26}, IX^1_{2,26}, IX^1_{3,26})$ through one S-box. Use the known difference $(\Delta Z^1_{01,26}, \Delta Z^1_{01,26})$ to filter the quartets. The number of remaining quartets is approximately

$$\frac{1}{\mathcal{N}(\Delta Z^1_{01,26})^2} \cdot \mathcal{Q}'\mathcal{N}(\Delta Z^1_{01,26})^2 = \mathcal{Q}'.$$

  The time complexity is

$$2^{156} \cdot (2^8)^i \cdot 2\rho \cdot \mathcal{Q}'\mathcal{N}(\Delta Z^1_{01,26})^2 = 2^{166+2d_{rout}}\mathcal{N}(\Delta Z^1_{01,26})^2.$$

So far, 164 key bits have been guessed, leaving $\mathcal{Q}'$ remaining quartets.

- 2dII: Apply the early abort technique in the last three rounds with steps outlined in Table III according to the bit conditions in Table II. The overall time complexity is dominated by Step 2dII.7, with a cost of $2^{220+(2d_{rout}-40)+16} \cdot 4\rho = 2^{191+2d_{out}}$.

- 3: Exhaustively search the remaining key.

TABLE III: The key recovery steps of the early abort technique for `ARADI` ($d_{rout} = 24, \rho = 4/(16 \cdot 32)$).

Step	Known difference	Guessed keys	Time complexity	Remained quartets
2dII.1	$\nabla X'^{15}_{12,0}, \nabla X'^{15}_{03,0}$	$IK'^{16}_{0,0}, IK'^{16}_{0,28}$	$2^{164+(2d_{rout}+8)+8} \cdot 2\rho$	$2^{2d_{rout}}$
2dII.2	$\nabla X'^{15}_{12,21}, \nabla X'^{15}_{03,21}$	$IK'^{16}_{0,21}$	$2^{172+(2d_{rout})+4} \cdot 1\rho$	$2^{2d_{rout}-8}$
2dII.3	$\nabla X'^{15}_{12,4}, \nabla X'^{15}_{03,4}$	$IK'^{16}_{0,4}, IK'^{16}_{0,11}$	$2^{176+(2d_{rout}-8)+8} \cdot 2\rho$	$2^{2d_{rout}-16}$
2dII.4	$\nabla X'^{15}_{12,25}, \nabla X'^{15}_{03,25}$	$IK'^{16}_{0,25}$	$2^{184+(2d_{rout}-16)+4} \cdot 1\rho$	$2^{2d_{rout}-24}$
2dII.5	$\nabla X'^{14}_{12,28}, \nabla X'^{14}_{03,28}$	$IK'^{15}_{0,1}, IK'^{15}_{0,28}, IK'^{16}_{0,8}, IK'^{16}_{0,14}, IK'^{16}_{0,19}$	$2^{188+(2d_{rout}-24)+20} \cdot 5\rho$	$2^{2d_{rout}-32}$
2dII.6	$\nabla X'^{14}_{12,11}, \nabla X'^{14}_{03,11}$	$IK'^{15}_{0,11}, IK'^{16}_{0,2}, IK'^{16}_{0,23}$	$2^{208+(2d_{rout}-32)+12} \cdot 3\rho$	$2^{2d_{rout}-40}$
2dII.7	$\nabla Y'^{13}_{12,1}, \nabla Y'^{13}_{03,1}$	$IK'^{14}_{0,1}, IK'^{15}_{0,7}, IK'^{15}_{0,24}, IK'^{16}_{0,24}$	$2^{220+(2d_{rout}-40)+16} \cdot 4\rho$	$2^{2d_{rout}-48}$

4) Complexity: The data complexity is 2^{130}. The time complexity includes the following components:

- Cost of Step 2a: Using 32-bit key and 8-bit difference pre-guesses, partial encryption and decryption for building tables PT_{01}, PT_{23}, CT_{12}, and CT_{03} requires a time complexity of $2^{40} \times 2 \times (2^{12 \cdot 2} \times 2/(32 \cdot 16) + 2^{20 \cdot 2} \times 2/(32 \cdot 16)) \approx 2^{73}$, and a memory complexity of $2 \times 2^{20} = 2^{21}$ (reusable across keys).
- Cost of Step 2b: Using 12-bit key pre-guesses $IK^0_{0,j}$ $(j \in \{7, 12, 13\})$, pairs are constructed from 2^{128} plaintexts, with time and memory complexity both approximately $2^{12} \times 2 \times 2^{128} \times |\mathcal{S}_{X_0}| \approx 2^{221.07}$.
- Cost of Step 2c: Using 32-bit key and 8-bit difference pre-guesses, quartets are generated, with a time complexity of $2^{40} \times (2^n|\mathcal{S}_{X_0}|)^2/2^{2(n-d_{\text{rout}})} \approx 2^{248.14}$ and a memory complexity of $2^8 \times (2^n|\mathcal{S}_{X_0}|)^2/2^{2(n-d_{\text{rout}})} \approx 2^{216.14}$.
- Cost of Step 2dI1: In Equation (8), since $\mathcal{N}(\gamma_j) > 4$, the time complexity of 2dI1.i decreases as i increases ($1 \leq i \leq 22$). Thus, the time complexity is dominated by $2^{33+2d_{rout}+4}|\mathcal{S}_{X_0}|^2 \approx 2^{245.14}$.
- Cost of Step 2dI2: In Equation (9), since $\mathcal{N}(\Delta Z^1_{01,j}) > 4$, the time complexity of Step 2dI2.i decreases as i increases ($1 \leq i \leq 7$). For $J^{1,0} = \{0, 9, 10, 15, 16, 17, 23, 26, 27\}$, and $(\Delta Z^1_{01,0}, \Delta Z^1_{01,9}, \Delta Z^1_{01,10}, \Delta Z^1_{01,15}, \Delta Z^1_{01,16}, \Delta Z^1_{01,17}, \Delta Z^1_{01,23}, \Delta Z^1_{01,26}, \Delta Z^1_{01,27}) = (12, 3, 8, 1, 2, 4, 4, 3, 8)$, $\prod_{j \in J^{1,0}} \mathcal{N}(\Delta Z^1_{01,j})^2 \approx 2^{44.19}$. Thus, the time complexity of Step 2dI2.$i(1 \leq i \leq 7)$ is $2^{121+2d_{rout}+4} \prod_{j \in J^{1,0}} \mathcal{N}(\Delta Z^1_{01,j})^2 \approx 2^{217.19}$. For Step 2dI2.8, for $J^{1,7} = \{9, 26\}$ and $(\Delta Z^1_{01,9}, \Delta Z^1_{01,26}) = (3, 3)$, we have $\prod_{j \in J^{1,7}} \mathcal{N}(\Delta Z^1_{01,j})^2 \approx 2^{10.34}$. Thus, the time complexity of Step 2dI2.8 is $2^{158+2d_{rout}} \prod_{j \in J^{1,7}} \mathcal{N}(\Delta Z^1_{01,j})^2 \approx 2^{216.34}$. For Step 2dI2.9, we have $\mathcal{N}(\Delta Z^1_{01,26})^2 \approx 2^{5.17}$ Thus, the time complexity of Step 2dI2.9 is $2^{166+2d_{rout}}(\mathcal{N}(\Delta Z^1_{01,26}))^2 \approx 2^{219.17}$.
- Cost of Step 2dII: The time complexity is $2^{191+2d_{rout}} \approx 2^{239}$.
- Cost of exhaustively search: The time complexity is $2^{256-2}(1-(1-1/e)^4) \approx 2^{253.75}$.

The overall data complexity is 2^{130}, the time complexity is $2^{253.75} + 2^{248.14} \approx 2^{253.78}$, and the memory complexity is $(1-(1-1/e)^4)2^{236} \approx 2^{235.75}$. Since $2^{130} + 2^{253.78} + 2^{235.75} < 2^{254}$, the attack can be carried out with a total complexity lower than that of exhaustive key search. Thus, we present the first successful full-round attack on `ARADI`.

B. The 34-round attack on SKINNYe v2

We present the first 34-round attack on `SKINNYe v2`, a block cipher proposed at EUROCRYPT 2020. We construct several 23-round `RK-IB distinguishers`, then extend six rounds before and five rounds after each to form a 34-round `RK-IB attack`.

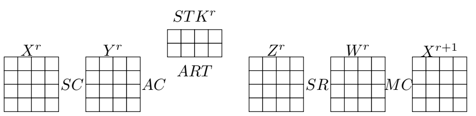


Fig. 17: One round encryption of `SKINNYe v2`

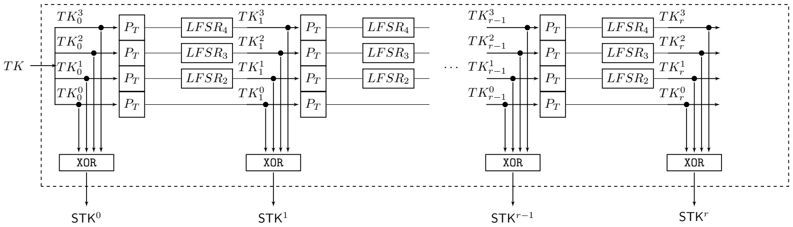


Fig. 18: The tweakey schedule of `SKINNYe v2`

1) Specifications of `SKINNYe v2`: `SKINNYe v2` [26] is a tweakable block cipher with 64-bit block and 256-bit tweakey. The 64-bit state is viewed as a 4×4 matrix of nibbles. `SKINNYe v2` consists of 44 rounds, and one round has five operations—ubCells (SC), AddConstants (AC), AddRoundTweakey (ART), ShiftRows (SR), and MixColumns (MC)—as shown in Figure 17:

- SubCells (SC): Apply the 4-bit S-box to each nibble of the state in parallel.

- AddConstants (AC): XOR the constant with parts of the state.
- AddRoundTweakey (ART): XOR the subtweakey with the top two rows of the state.
- ShiftRows (SR): Rotate the i-th row ($0 \le i \le 3$) of the state right by i bytes.
- MixColumns (MC): Multiply each column of the state by the binary matrix

$$\boldsymbol{M} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

The 256-bit tweakey state consists of four 4×4 matrices of nibbles, denoted as TK_i^r ($0 \le i \le 3$); the top two rows of $\oplus_{i=0}^{3} TK_i^r$ form the subtweakey STK^r. As shown in Figure 18, at round r, the first and second rows of TK_i^r ($1 \le i \le 3$) are updated cell by cell via $LFSR_i$. Specifically, for $(x_0, x_1, x_2, x_3) \in \mathbb{F}_2^4$:

$$\begin{aligned} LFSR_1 &: (x_3 \| x_2 \| x_1 \| x_0) \to (x_2 \| x_1 \| x_0 \| x_3 \oplus x_2), \\ LFSR_2 &: (x_3 \| x_2 \| x_1 \| x_0) \to (x_0 \oplus x_3 \| x_3 \| x_2 \| x_1), \\ LFSR_3 &: (x_3 \| x_2 \| x_1 \| x_0) \to (x_1 \| x_0 \| x_3 \oplus x_2 \| x_2 \oplus x_1). \end{aligned}$$

Then, all tweak states TK_i^r ($0 \le i \le 3$) are permuted using $P_T = [9, 15, 8, 13, 10, 14, 12, 11, 0, 1, 2, 3, 4, 5, 6, 7]$.

2) The 23-round `RK-IB distinguishers` of `SKINNYe v2`: The tweakey schedule of `SKINNYe v2` has the following property, enabling the construction of 8-round probability-1 upper related-key differentials.

Property 3. *Let $\mathbf{0} = (0,0,0,0,0,0,0,0)$, $DSTK^r$ denote the subtweakey difference at round r, and ΔTK_i^r ($0 \le i \le 3$) denote the difference of the i-th tweakey state at round r. For any r and any value of $DSTK^r$ or $DSTK^{r+8}$, there exist unique $\Delta TK_0^r, \Delta TK_1^r, \Delta TK_2^r, \Delta TK_3^r \in (\mathbb{F}_2^4)^{16}$ such that $DSTK^{r+i} = \mathbf{0}$ for $1 \le i \le 7$.*

Proof. Suppose that a single cell of $\Delta TK_0^r, \Delta TK_1^r, \Delta TK_2^r, \Delta TK_3^r$ are active, and a_0, a_1, a_2, a_3 are the corresponding values. For any non-zero value $u \in \mathbb{F}_2^4$, since the LFSRs are linear, the equations

$$\begin{cases} a_0 \oplus a_1 \oplus a_2 \oplus a_3 = u, \\ a_0 \oplus \mathrm{LFSR}_1^1(a_1) \oplus \mathrm{LFSR}_2^1(a_2) \oplus \mathrm{LFSR}_3^1(a_3) = 0, \\ a_0 \oplus \mathrm{LFSR}_1^2(a_1) \oplus \mathrm{LFSR}_2^2(a_2) \oplus \mathrm{LFSR}_3^2(a_3) = 0, \\ a_0 \oplus \mathrm{LFSR}_1^3(a_1) \oplus \mathrm{LFSR}_2^3(a_2) \oplus \mathrm{LFSR}_3^3(a_3) = 0, \end{cases}$$

have exactly one solution for (a_0, a_1, a_2, a_3), where LFSR_j^i denotes the operation of applying LFSR_j i times ($i > 0$, $j = 1, 2, 3$). For these values of a_0, a_1, a_2, and a_3, we have $DSTK^{r+i} = \mathbf{0}$ for $1 \le i \le 7$.

Since LFSRs operate independently on each cell, when multiple cells of $\Delta TK_0^r, \Delta TK_1^r, \Delta TK_2^r, \Delta TK_3^r$ are active, we can conduct the proof by analogy. □

Similarly, an 8-round probability-1 lower related-key differential can be constructed. We then extend the short-round `RK-IB distinguisher` using the above 8-round differentials to build the full `RK-IB distinguisher`.

Construction. Let (α, α_{core}) be an 8-round probability-1 related-key differential under master tweakey difference ΔTK from round r_0 to $r_0 + 7$, and let (β_{core}, β) be an 8-round probability-1 related-key differential under master tweakey difference ∇TK from round $r_1 - 7$ to r_1. If $(\alpha_{core}, \alpha_{core}, \beta_{core}, \beta_{core})$ forms an $(r_1 - r_0 - 15)$-round `RK-IB distinguisher` under master tweakey differences $(\Delta TK, \Delta TK, \nabla TK, \nabla TK)$, then $(\alpha, \alpha, \beta, \beta)$ is an $(r_1 - r_0 + 1)$-round `RK-IB distinguisher` under $(\Delta TK, \Delta TK, \nabla TK, \nabla TK)$. By Property 3, α, α_{core}, and ΔTK are uniquely determined by $DSTK^{r_0}$, and β, β_{core}, ∇TK are uniquely determined by $DSTK'^{r_1}$. Thus, `RK-IB distinguishers` can be found by traversing $DSTK^{r_0}$ and $DSTK'^{r_1}$.

Result. To maximize the number of rounds added before and after the distinguisher, the input and output differences should have minimal weight. We set $DSTK^{r_0}$ and $DSTK'^{r_1}$ to each contain one active nibble, reducing the search space to $(8 \times 15)^2 = 14400$. Using the GEBCT-based $\mathcal{HJF}$-method, we find 7 distinct 23-round 1-active-nibble `RK-IB distinguishers` in approximately 1.82 hours. One is shown in Distinguisher 2. Cross-validation via the state-based $\mathcal{HJF}$-method confirms its correctness.

Distinguisher 2. *Let $\Delta TK^r = (\Delta TK_0^r, \Delta TK_1^r, \Delta TK_2^r, \Delta TK_3^r)$ and $\nabla TK^r = (\nabla TK_0^r, \nabla TK_1^r, \nabla TK_2^r, \nabla TK_3^r)$ denote the tweakey differences at round r in the upper and lower trails, respectively. Let α and β be the input and output*

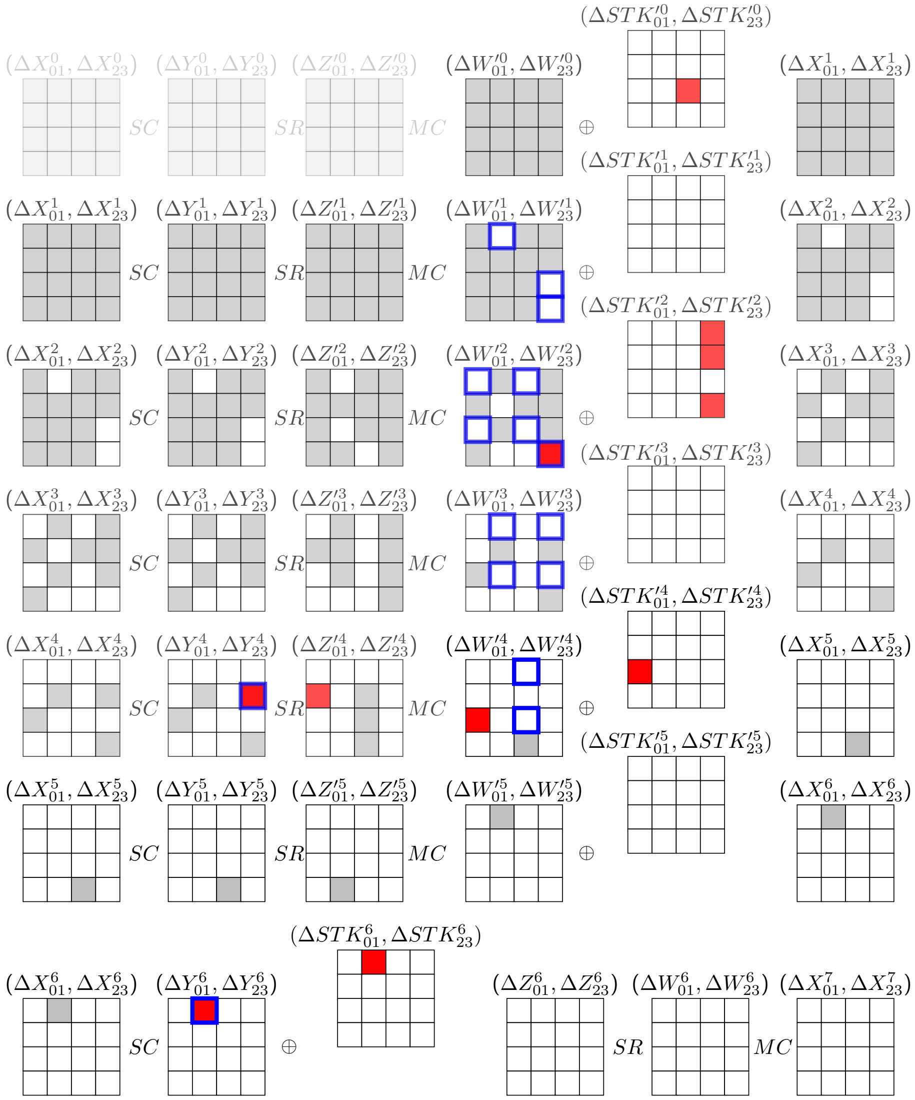

$(\Delta STK'^0_{01}, \Delta STK'^0_{23})$
$(\Delta X^0_{01}, \Delta X^0_{23})$ $(\Delta Y^0_{01}, \Delta Y^0_{23})$ $(\Delta Z'^0_{01}, \Delta Z'^0_{23})$ $(\Delta W'^0_{01}, \Delta W'^0_{23})$ $(\Delta X^1_{01}, \Delta X^1_{23})$
SC SR MC ⊕
$(\Delta STK'^1_{01}, \Delta STK'^1_{23})$
$(\Delta X^1_{01}, \Delta X^1_{23})$ $(\Delta Y^1_{01}, \Delta Y^1_{23})$ $(\Delta Z'^1_{01}, \Delta Z'^1_{23})$ $(\Delta W'^1_{01}, \Delta W'^1_{23})$ $(\Delta X^2_{01}, \Delta X^2_{23})$
SC SR MC ⊕
$(\Delta STK'^2_{01}, \Delta STK'^2_{23})$
$(\Delta X^2_{01}, \Delta X^2_{23})$ $(\Delta Y^2_{01}, \Delta Y^2_{23})$ $(\Delta Z'^2_{01}, \Delta Z'^2_{23})$ $(\Delta W'^2_{01}, \Delta W'^2_{23})$ $(\Delta X^3_{01}, \Delta X^3_{23})$
SC SR MC ⊕
$(\Delta STK'^3_{01}, \Delta STK'^3_{23})$
$(\Delta X^3_{01}, \Delta X^3_{23})$ $(\Delta Y^3_{01}, \Delta Y^3_{23})$ $(\Delta Z'^3_{01}, \Delta Z'^3_{23})$ $(\Delta W'^3_{01}, \Delta W'^3_{23})$ $(\Delta X^4_{01}, \Delta X^4_{23})$
SC SR MC ⊕
$(\Delta STK'^4_{01}, \Delta STK'^4_{23})$
$(\Delta X^4_{01}, \Delta X^4_{23})$ $(\Delta Y^4_{01}, \Delta Y^4_{23})$ $(\Delta Z'^4_{01}, \Delta Z'^4_{23})$ $(\Delta W'^4_{01}, \Delta W'^4_{23})$ $(\Delta X^5_{01}, \Delta X^5_{23})$
SC SR MC ⊕
$(\Delta STK'^5_{01}, \Delta STK'^5_{23})$
$(\Delta X^5_{01}, \Delta X^5_{23})$ $(\Delta Y^5_{01}, \Delta Y^5_{23})$ $(\Delta Z'^5_{01}, \Delta Z'^5_{23})$ $(\Delta W'^5_{01}, \Delta W'^5_{23})$ $(\Delta X^6_{01}, \Delta X^6_{23})$
SC SR MC ⊕
$(\Delta STK^6_{01}, \Delta STK^6_{23})$
$(\Delta X^6_{01}, \Delta X^6_{23})$ $(\Delta Y^6_{01}, \Delta Y^6_{23})$ $(\Delta Z^6_{01}, \Delta Z^6_{23})$ $(\Delta W^6_{01}, \Delta W^6_{23})$ $(\Delta X^7_{01}, \Delta X^7_{23})$
SC ⊕ SR MC

Fig. 19: Top 6 rounds added for key recovery in 34-round attack on SKINNYe v2. † The white square denotes an inactive difference nibble. The red square denotes an active and known difference nibble. The gray square denotes an unknown difference nibble. The square enclosed by blue lines corresponds to a known difference in Table IV.

differences. Then $(\alpha, \alpha, \beta, \beta)$ *forms a 23-round* RK-IB distinguisher *for* SKINNYe v2 *under the tweakey differences* $(\Delta TK, \Delta TK, \nabla TK, \nabla TK)$*, where*

$$\begin{cases} \alpha = 0x0700000000000000, \ \beta = 0x0008000800000008, \\ \Delta TK_0^r = 0x0e00000000000000, \ \Delta TK_1^r = 0x0700000000000000, \\ \Delta TK_2^r = 0x0600000000000000, \ \Delta TK_3^r = 0x0800000000000000, \\ \nabla TK_0^{22} = 0x0008000000000000, \ \nabla TK_1^{22} = 0x0002000000000000, \\ \nabla TK_2^{22} = 0x0009000000000000, \ \nabla TK_3^{22} = 0x000b000000000000, \end{cases}$$

and ∇TK *is uniquely determined by* ∇TK^{22} *via the tweakey schedule.*

3) The 34*-round* RK-IB attack *of* SKINNYe v2*:* We extend the 23-round RK-IB distinguisher in Distinguisher 2 by adding six rounds before and five rounds after to achieve a 34-round attack on SKINNYe v2. Let $(ISTK_0^r, ISTK_1^r, ISTK_2^r, ISTK_3^r)$ denote the subtweakey quartet in round r, with ΔSTK_{01}^r, ΔSTK_{23}^r for upper subtweakey differences and ∇STK_{12}^r, ∇STK_{03}^r for lower subtweakey differences. The attack overview is shown in Figure 9. Bit conditions are shown in Table IV and Table V, where $ISTK_{0,A}^r = \cup_{j \in A}\{ISTK_{0,j}^r\}$. Two techniques are used: pre-guessing and early abort, with an automatic key-guessing strategy to optimize the guessing order for early abort. Specifically, we pre-guess 26×4-bit tweakeys given in Table IV and use the remaining bit conditions for early abort with automatic optimal tweakey-guessing. The differential propagation in the top six rounds and bottom five rounds are illustrated in Fig. 19

TABLE IV: The known differences and the tweakeys to guess in E_b for `SKINNYe v2`.

Known difference	Keys to guess
$\Delta W'^{1}_{01,1}, \Delta W'^{1}_{23,1}$	$ISTK^{0}_{0,\{1,2,6\}}$
$\Delta W'^{1}_{01,11}, \Delta W'^{1}_{23,11}$	$ISTK^{0}_{0,\{2,4\}}$
$\Delta W'^{1}_{01,15}, \Delta W'^{1}_{23,15}$	$ISTK^{0}_{0,\{3,4\}}$
$\Delta W'^{2}_{01,0}, \Delta W'^{2}_{23,0}$	$ISTK^{0}_{0,\{0,1,5,6,7\}}, ISTK^{1}_{0,\{0,1,5\}}$
$\Delta W'^{2}_{01,8}, \Delta W'^{2}_{23,8}$	$ISTK^{0}_{0,\{1,3,7\}}, ISTK^{1}_{0,\{3,5\}}$
$\Delta W'^{2}_{01,2}, \Delta W'^{2}_{23,2}$	$ISTK^{0}_{0,\{2,3,5,7\}}, ISTK^{1}_{0,\{2,7\}}$
$\Delta W'^{2}_{01,10}, \Delta W'^{2}_{23,10}$	$ISTK^{0}_{0,\{1,3,5\}}, ISTK^{1}_{0,\{1,7\}}$
$\Delta W'^{2}_{01,15}, \Delta W'^{2}_{23,15}$	$ISTK^{0}_{0,\{0,3,4,6\}}, ISTK^{1}_{0,\{3,4\}}$
$\Delta W'^{3}_{01,1}, \Delta W'^{3}_{23,1}$	$ISTK^{0}_{0,\{0,1,2,4,6,7\}}, ISTK^{1}_{0,\{1,2,4,6\}}, ISTK^{2}_{0,\{1,6\}}$
$\Delta W'^{3}_{01,3}, \Delta W'^{3}_{23,3}$	$ISTK^{0}_{0,\{0,1,2,3,4,5,6,7\}}, ISTK^{1}_{0,\{0,3,4,5,6\}}, ISTK^{2}_{0,\{0,3,4\}}$
$\Delta W'^{3}_{01,9}, \Delta W'^{3}_{23,9}$	$ISTK^{0}_{0,\{0,1,2,5,6\}}, ISTK^{1}_{0,\{0,2,4\}}, ISTK^{2}_{0,\{0,6\}}$
$\Delta W'^{3}_{01,11}, \Delta W'^{3}_{23,11}$	$ISTK^{0}_{0,\{0,2,3,4,7\}}, ISTK^{1}_{0,\{0,2,6\}}, ISTK^{2}_{0,\{2,4\}}$
$\Delta Y^{4}_{01,7}, \Delta Y^{4}_{23,7}$	$ISTK^{0}_{0,\{0,3,4,5,6\}}, ISTK^{1}_{0,\{0,3,4\}}, ISTK^{2}_{0,\{3\}}, ISTK^{3}_{0,\{3\}}$
$\Delta W'^{4}_{01,2}, \Delta W'^{4}_{23,2}$	$ISTK^{0}_{0,\{0,1,2,3,4,5,6\}}, ISTK^{1}_{0,\{0,1,3,4,6,7\}}, ISTK^{2}_{0,\{3,4,5\}}, ISTK^{3}_{0,\{3,7\}}$
$\Delta W'^{4}_{01,10}, \Delta W'^{4}_{23,10}$	$ISTK^{0}_{0,\{0,1,2,3,4,5,6,7\}}, ISTK^{1}_{0,\{1,2,3,6,7\}}, ISTK^{2}_{0,\{1,3,5\}}, ISTK^{3}_{0,\{1,7\}}$
$\Delta Y^{6}_{01,1}, \Delta Y^{6}_{23,1}$	$ISTK^{0}_{0,\{0,1,2,3,4,5,6,7\}}, ISTK^{1}_{0,\{0,1,2,3,4,5,6,7\}},$ $ISTK^{2}_{0,\{0,1,2,3,4,5,6,7\}}, ISTK^{3}_{0,\{1,2,4,6,7\}}, ISTK^{4}_{0,\{1,2,6\}}, ISTK^{5}_{0,\{1\}}$

† The gray-shaded regions represent the pre-guessed tweakey nibbles.

TABLE V: The known differences and the tweakeys to guess in E_f for `SKINNYe v2`.

Known difference	Keys to guess
$\nabla W^{32}_{12,12}, \nabla W^{32}_{03,12}$	$ISTK^{33}_{0,\{0\}}$
$\nabla W^{32}_{12,5}, \nabla W^{32}_{03,5}$	$ISTK^{33}_{0,\{5\}}$
$\nabla W^{32}_{12,6}, \nabla W^{32}_{03,6}$	$ISTK^{33}_{0,\{6\}}$
$\nabla W^{32}_{12,11}, \nabla W^{32}_{03,11}$	$ISTK^{33}_{0,\{7\}}$
$\nabla W^{31}_{12,12}, \nabla W^{31}_{03,12}$	$ISTK^{33}_{0,\{3,4\}}, ISTK^{32}_{0,\{0\}}$
$\nabla W^{31}_{12,6}, \nabla W^{31}_{03,6}$	$ISTK^{33}_{0,\{1,4,7\}}, ISTK^{32}_{0,\{6\}}$
$\nabla W^{31}_{12,11}, \nabla W^{31}_{03,11}$	$ISTK^{33}_{0,\{2,4\}}, ISTK^{32}_{0,\{7\}}$
$\nabla W^{31}_{12,15}, \nabla W^{31}_{03,15}$	$ISTK^{33}_{0,\{2,7\}}, ISTK^{32}_{0,\{3\}}$
$\nabla W^{30}_{12,6}, \nabla W^{30}_{03,6}$	$ISTK^{33}_{0,\{0,2,3,4,5\}}, ISTK^{32}_{0,\{1,4,7\}}, ISTK^{31}_{0,\{6\}}$
$\nabla W^{30}_{12,7}, \nabla W^{30}_{03,7}$	$ISTK^{33}_{0,\{1,3,5,6\}}, ISTK^{32}_{0,\{2,4\}}, ISTK^{31}_{0,\{7\}}$
$\nabla W^{30}_{12,15}, \nabla W^{30}_{03,15}$	$ISTK^{33}_{0,\{1,4,6\}}, ISTK^{32}_{0,\{2,7\}}, ISTK^{31}_{0,\{3\}}$
$\nabla W^{29}_{12,7}, \nabla W^{29}_{03,7}$	$ISTK^{33}_{0,\{0,1,2,5,6,7\}}, ISTK^{32}_{0,\{1,3,5,6\}}, ISTK^{31}_{0,\{2,4\}}, ISTK^{30}_{0,\{7\}}$
$\nabla W^{29}_{12,15}, \nabla W^{29}_{03,15}$	$ISTK^{33}_{0,\{0,3,5,6,7\}}, ISTK^{32}_{0,\{1,4,6\}}, ISTK^{31}_{0,\{2,7\}}, ISTK^{30}_{0,\{3\}}$
$\nabla X^{29}_{12,3}, \nabla X^{29}_{03,3}$	$ISTK^{33}_{0,\{0,1,2,6,7\}}, ISTK^{32}_{0,\{3,5,6\}}, ISTK^{31}_{0,\{4\}}, ISTK^{30}_{0,\{7\}}, ISTK^{29}_{0,\{3\}}$
$\nabla X^{29}_{12,7}, \nabla X^{29}_{03,7}$	$ISTK^{33}_{0,\{0,1,2,3,4,5,6,7\}}, ISTK^{32}_{0,\{0,1,2,6,7\}}, ISTK^{31}_{0,\{3,5,6\}}, ISTK^{30}_{0,\{4\}}, ISTK^{29}_{0,\{7\}}$
$\nabla X^{29}_{12,15}, \nabla X^{29}_{03,15}$	$ISTK^{33}_{0,\{2,3,4,5,6\}}, ISTK^{32}_{0,\{0,5,7\}}, ISTK^{31}_{0,\{1,6\}}, ISTK^{30}_{0,\{2\}}$

TABLE VI: The key recovery steps of the early abort technique of `SKINNYe v2` ($\rho = 4/(16 \cdot 34)$).

Step	Difference	Guessed tweakeys	Time complexity	Remained quartets
2d1	$\nabla W^{32}_{12,12}, \nabla W^{32}_{03,12}$	$ISTK^{33}_{0,0}$	$2^{26c+c+34c} \cdot 1\rho$	2^{34c-2c}
2d2	$\nabla W^{32}_{12,5}, \nabla W^{32}_{03,5}$	$ISTK^{33}_{0,5}$	$2^{27c+c+32c} \cdot 1\rho$	2^{32c-2c}
2d3	$\nabla W^{32}_{12,6}, \nabla W^{32}_{03,6}$	$ISTK^{33}_{0,6}$	$2^{28c+c+30c} \cdot 1\rho$	2^{30c-2c}
2d4	$\nabla W^{32}_{12,11}, \nabla W^{32}_{03,11}$	$ISTK^{33}_{0,7}$	$2^{29c+c+28c} \cdot 1\rho$	2^{28c-2c}
2d5	$\nabla W^{31}_{12,15}, \nabla W^{31}_{03,15}$	$ISTK^{33}_{0,2}, ISTK^{32}_{0,3}$	$2^{30c+2c+26c} \cdot 2\rho$	2^{26c-2c}
2d6	$\nabla W^{31}_{12,11}, \nabla W^{31}_{03,11}$	$ISTK^{33}_{0,4}, ISTK^{32}_{0,7}$	$2^{32c+2c+24c} \cdot 2\rho$	2^{24c-2c}
2d7	$\nabla W^{31}_{12,6}, \nabla W^{31}_{03,6}$	$ISTK^{33}_{0,1}, ISTK^{32}_{0,6}$	$2^{34c+2c+22c} \cdot 2\rho$	2^{22c-2c}
2d8	$\nabla W^{31}_{12,12}, \nabla W^{31}_{03,12}$	$ISTK^{33}_{0,3}, ISTK^{32}_{0,0}$	$2^{36c+2c+20c} \cdot 2\rho$	2^{20c-2c}
2d9	$\nabla W^{30}_{12,15}, \nabla W^{30}_{03,15}$	$ISTK^{32}_{0,2}, ISTK^{31}_{0,3}$	$2^{38c+2c+18c} \cdot 2\rho$	2^{18c-2c}
2d10	$\nabla W^{30}_{12,7}, \nabla W^{30}_{03,7}$	$ISTK^{32}_{0,4}, ISTK^{31}_{0,7}$	$2^{40c+2c+16c} \cdot 2\rho$	2^{16c-2c}
2d11	$\nabla W^{30}_{12,6}, \nabla W^{30}_{03,6}$	$ISTK^{32}_{0,1}, ISTK^{31}_{0,6}$	$2^{42c+2c+14c} \cdot 2\rho$	2^{14c-2c}
2d12	$\nabla W^{29}_{12,15}, \nabla W^{29}_{03,15}$	$ISTK^{31}_{0,2}, ISTK^{30}_{0,3}$	$2^{44c+2c+12c} \cdot 2\rho$	2^{12c-2c}
2d13	$\nabla W^{29}_{12,7}, \nabla W^{29}_{03,7}$	$ISTK^{32}_{0,5}, ISTK^{31}_{0,4}, ISTK^{30}_{0,7}$	$2^{46c+3c+10c} \cdot 3\rho$	2^{10c-2c}
2d14	$\nabla X^{29}_{12,3}, \nabla X^{29}_{03,3}$	$ISTK^{29}_{0,3}$	$2^{49c+c+8c} \cdot 1\rho$	2^{8c-2c}
2d15	$\nabla X^{29}_{12,15}, \nabla X^{29}_{03,15}$	$ISTK^{31}_{0,1}, ISTK^{30}_{0,2}$	$2^{50c+2c+6c} \cdot 2\rho$	2^{6c-2c}
2d16	$\nabla X^{29}_{12,7}, \nabla X^{29}_{03,7}$	$ISTK^{31}_{0,5}, ISTK^{30}_{0,4}, ISTK^{29}_{0,7}$	$2^{52c+2c+4c} \cdot 3\rho$	2^{4c-2c}
2d17	$\Delta Y^{6}_{01,1}, \Delta Y^{6}_{23,1}$	$ISTK^{2}_{0,7}, ISTK^{3}_{0,2}, ISTK^{3}_{0,4}, ISTK^{3}_{0,6}$ $ISTK^{4}_{0,1}, ISTK^{4}_{0,2}, ISTK^{4}_{0,6}, ISTK^{5}_{0,1}$	$2^{54c+6c+2c} \cdot 6\rho$	2^{2c-2c}

† Red-marked tweakeys are derived from the tweakey schedule by previously guessed tweakeys.

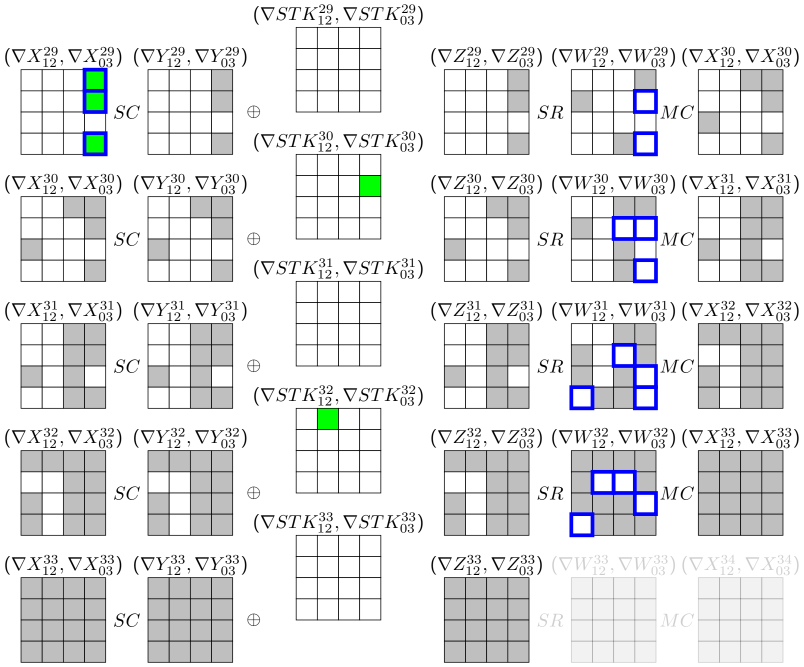

$(\nabla STK^{29}_{12}, \nabla STK^{29}_{03})$
$(\nabla X^{29}_{12}, \nabla X^{29}_{03})$
$(\nabla Y^{29}_{12}, \nabla Y^{29}_{03})$
$(\nabla Z^{29}_{12}, \nabla Z^{29}_{03})$
$(\nabla W^{29}_{12}, \nabla W^{29}_{03})$
$(\nabla X^{30}_{12}, \nabla X^{30}_{03})$
SC
⊕
SR
MC
$(\nabla STK^{30}_{12}, \nabla STK^{30}_{03})$
$(\nabla X^{30}_{12}, \nabla X^{30}_{03})$
$(\nabla Y^{30}_{12}, \nabla Y^{30}_{03})$
$(\nabla Z^{30}_{12}, \nabla Z^{30}_{03})$
$(\nabla W^{30}_{12}, \nabla W^{30}_{03})$
$(\nabla X^{31}_{12}, \nabla X^{31}_{03})$
$(\nabla STK^{31}_{12}, \nabla STK^{31}_{03})$
$(\nabla X^{31}_{12}, \nabla X^{31}_{03})$
$(\nabla Y^{31}_{12}, \nabla Y^{31}_{03})$
$(\nabla Z^{31}_{12}, \nabla Z^{31}_{03})$
$(\nabla W^{31}_{12}, \nabla W^{31}_{03})$
$(\nabla X^{32}_{12}, \nabla X^{32}_{03})$
$(\nabla STK^{32}_{12}, \nabla STK^{32}_{03})$
$(\nabla X^{32}_{12}, \nabla X^{32}_{03})$
$(\nabla Y^{32}_{12}, \nabla Y^{32}_{03})$
$(\nabla Z^{32}_{12}, \nabla Z^{32}_{03})$
$(\nabla W^{32}_{12}, \nabla W^{32}_{03})$
$(\nabla X^{33}_{12}, \nabla X^{33}_{03})$
$(\nabla STK^{33}_{12}, \nabla STK^{33}_{03})$
$(\nabla X^{33}_{12}, \nabla X^{33}_{03})$
$(\nabla Y^{33}_{12}, \nabla Y^{33}_{03})$
$(\nabla Z^{33}_{12}, \nabla Z^{33}_{03})$
$(\nabla W^{33}_{12}, \nabla W^{33}_{03})$
$(\nabla X^{34}_{12}, \nabla X^{34}_{03})$

Fig. 20: Bottom 5 rounds added for key recovery in 34-round attack on `SKINNYe v2`. † The white square denotes an inactive difference nibble. The green square denotes an active and known difference nibble. The gray square denotes an unknown difference nibble. The square enclosed by blue lines corresponds to a known difference in Table V.

and Fig. 20, respectively. The total data, time, and memory complexities sum to less than 2^{254}, confirming an effective 34-round attack. Note that we equivalently swap $SR \circ SM$ and ART in the top six rounds; the resulting difference, state, and key are marked with $'$.

By analyzing tweakey dependencies, we find that any four tweakeys from each of the following sets can be used to derive the remaining tweakey in that set:

$$\begin{cases}\{ISTK^{0}_{0,1}, ISTK^{2}_{0,0}, ISTK^{4}_{0,2}, ISTK^{30}_{0,7}, ISTK^{32}_{0,1}\},\\ \{ISTK^{1}_{0,3}, ISTK^{3}_{0,7}, ISTK^{5}_{0,1}, ISTK^{31}_{0,5}, ISTK^{33}_{0,3}\},\\ \{ISTK^{1}_{0,0}, ISTK^{3}_{0,2}, ISTK^{29}_{0,7}, ISTK^{31}_{0,1}, ISTK^{33}_{0,0}\}.\end{cases}$$

Let $c = 4$. The entire attack process is as follows.

- 1 (Generate Data): For all 2^{16c} plaintexts P, partially encrypt to compute IW'^{0}, and obtain ciphertexts (C_0, C_1, C_2, C_3) under four related tweakeys $(MK_0, MK_1, MK_2, MK_3) = (TK, TK \oplus \Delta TK, TK \oplus \Delta TK \oplus \nabla TK, TK \oplus \nabla TK)$, where ΔTK and ∇TK are the initial-round tweakey differences derived from Distinguisher 2. Denote the sets of plaintext-ciphertext pairs as $T_i = \{(IW'^{0}_{i}, C_i)\}$ for $0 \le i \le 3$.
- 2: Pre-guess $ISTK^{0}_{0,\{0,1,2,3,4,5,6,7\}}$, $ISTK^{1}_{0,\{0,1,2,3,4,5,6,7\}}$, $ISTK^{2}_{0,\{0,1,2,3,4,5,6\}}$, and $ISTK^{3}_{0,\{1,3,7\}}$. Then, construct the quartets as follows:
  - 2a (Build Tables): For each IW'^{0}_{i}, perform partial encryption using the guessed tweakeys and verify:

$$\begin{cases}IW'^{1}_{0,j} \oplus IW'^{1}_{1,j} = 0, (j = 1, 11, 15),\\ IW'^{2}_{0,j} \oplus IW'^{2}_{1,j} = 0, (j = 0, 2, 8, 10, 15),\\ IW'^{3}_{0,j} \oplus IW'^{3}_{1,j} = 0, (j = 1, 3, 9, 11),\\ IY^{4}_{0,7} \oplus IY^{4}_{1,7} = 0,\\ IW'^{4}_{0,j} \oplus IW'^{4}_{1,j} = 0, (j = 2, 10).\end{cases}$$

    and store $(IW'^{0}_{0}, IW'^{0}_{1})$ in table PT_{01}. Similarly, construct PT_{23} for $(IW'^{0}_{2}, IW'^{0}_{3})$. Thus, $|PT_{01}| = |PT_{23}| = (2^{16c})^2/(2^{15c}) = 2^{17c}$.
  - 2b (Build Pairs): For all 2^{2c} possible values of IW'^{0}_{0}, retrieve IW'^{0}_{1} from PT_{01}, then obtain corresponding ciphertexts $(IZ^{33}_{0}, IZ^{33}_{1})$ from tables T_0 and T_1. This yields $\mathcal{P} = 2^{17c}$ pairs $\big((IW'^{0}_{0}, IZ^{33}_{0}), (IW'^{0}_{1}, IZ^{33}_{1})\big)$. Similarly generate $\mathcal{P}$ pairs for $(IW'^{0}_{2}, IZ^{33}_{2})$ and $(IW'^{0}_{3}, IZ^{33}_{3})$.
  - 2c (Produce Quartets): Since all differences in the ciphertext are unknown, directly form the $\mathcal{Q} = (2^{17c})^2$ quartets $\big((IW'^{0}_{0}, IZ^{33}_{0}), (IW'^{0}_{1}, IZ^{33}_{1}), (IW'^{0}_{2}, IZ^{33}_{2}), (IW'^{0}_{3}, IZ^{33}_{3})\big)$.

- 2d (Filter Quartets): Apply the early abort technique by guessing the remaining tweakeys stepwise according to the bit conditions in Table IV and Table V, as detailed in Table VI. Define $\rho = 4/(16 \cdot 34)$ as the basic cost of partial quartet encryption/decryption. The overall time complexity is dominated by Step 2d17, with a cost of $2^{54c+6c+2c} \cdot 5\rho$.

- 3: Exhaustively search the remaining tweakeys.

Complexity. The number of tweakeys to guess is 2^{60c}. After applying the early abort technique, the probability of retaining a tweakey is $p = (1 - 2^{-34c})^Q = e^{-1}$. The data complexity is $2^2 \cdot 2^{16c} = 2^{66}$. For the time complexity:

- Cost of Step 2a to Step 2c: $2^2 \cdot 2^{26c} \cdot (2^{16c})^2 \cdot 60/(16 \cdot 34) + 2^{26c} \cdot (2^{17c})^2 \approx 2^{240}$, for partial encryption that uses $(16 + 16 + 13 + 5)$ S-boxes per $(W_i'^0, Z_i^{33})$.
- Cost of Step 2d: $2^{62c} \cdot (5 \cdot 4)/(16 \cdot 34) \approx 2^{243.24}$, as shown in Table VI.
- Cost of Step 3: $2^{256-2}(1 - (1 - 1/e)^4) \approx 2^{253.75}$.

The main memory cost comes from storing candidate tweakeys during the early abort phase. As we guess 2^{60c} tweakeys, the memory complexity is $2^{240}(1 - (1 - 1/e)^4) \approx 2^{239.75}$.

In summary, we successfully attack 34-round SKINNYe v2 with 2^{66} data complexity, $2^{253.75}$ time complexity, and $2^{239.75}$ memory complexity.

VI. Conclusion and Future Works

In this paper, building on the state-of-the-art search method for (RK-)IB distinguishers, we propose the pre-sieving technique, pre-guessing technique, and automatic key-guessing strategy—together forming a unified framework with rigorous complexity analysis for (RK-)IB attacks. This framework is the first to offer flexible pre-guessing of keys and differences while incorporating cipher-specific details in the key recovery phase. Notably, it enables the automatic generation of step-by-step key recovery procedures. As a result, we successfully achieve the first full-round attack on ARADI, a block cipher proposed by NSA, and the first 34-round RK-IB attack on SKINNYe v2, a block cipher proposed at EUROCRYPT 2020. These results highlight the framework's significance and its substantial improvement in (RK-)IB attacks.

While instantiated here for ARADI and SKINNYe v2, the framework can be extended to other block ciphers with SPN, Feistel, and ARX structures. We leave such extensions for future work.

References

[1] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016. I, IV-C

[2] Emanuele Bellini, Mattia Formenti, David Gérault, Juan Grados, Anna Hambitzer, Yun Ju Huang, Paul Huynh, Mohamed Rachidi, Raghvendra Rohit, and Sharwan K. Tiwari. Claasping ARADI: automated analysis of the ARADI block cipher. In Sourav Mukhopadhyay and Pantelimon Stanica, editors, *Progress in Cryptology - INDOCRYPT 2024 - 25th International Conference on Cryptology in India, Chennai, India, December 18-21, 2024, Proceedings, Part II*, volume 15496 of *Lecture Notes in Computer Science*, pages 90–113. Springer, 2024. I

[3] Emanuele Bellini, Mohamed Rachidi, Raghvendra Rohit, and Sharwan K. Tiwari. Mind the composition of toffoli gates: Structural algebraic distinguishers of ARADI. *IACR Cryptol. ePrint Arch.*, page 1559, 2024. I

[4] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999. I

[5] Xavier Bonnetain, Margarita Cordero, Virginie Lallemand, Marine Minier, and María Naya-Plasencia. On impossible boomerang attacks application to simon and skinnyee. *IACR Trans. Symmetric Cryptol.*, 2024(2):222–253, 2024. I, I, II-B, 2, II-D, II-D, IV-D

[6] Hamid Boukerrou, Paul Huynh, Virginie Lallemand, Bimal Mandal, and Marine Minier. On the feistel counterpart of the boomerang connectivity table introduction and analysis of the FBCT. *IACR Trans. Symmetric Cryptol.*, 2020(1):331–362, 2020. I

[7] Christina Boura, Nicolas David, Patrick Derbez, Rachelle Heim Boissier, and María Naya-Plasencia. A generic algorithm for efficient key recovery in differential attacks - and its associated tool. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 217–248. Springer, 2024. 29

[8] Debasmita Chakraborty, Hosein Hadipour, Phuong Hoa Nguyen, and Maria Eichlseder. Finding complete impossible differential attacks on andrx ciphers and efficient distinguishers for ARX designs. *IACR Trans. Symmetric Cryptol.*, 2024(3):84–176, 2024. I

[9] Yincen Chen, Qinggan Fu, Ning Zhao, Jiahao Zhao, Ling Song, and Qianqian Yang. A holistic framework for impossible boomerang attacks. *IACR Commun. Cryptol.*, 2(2):18, 2025. 1, I

[10] Carlos Cid, Tao Huang, Thomas Peyrin, Yu Sasaki, and Ling Song. Boomerang connectivity table: A new cryptanalysis tool. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 683–714. Springer, 2018. I

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

[12] Stéphanie Delaune, Patrick Derbez, and Mathieu Vavrille. Catching the fastest boomerangs application to SKINNY. *IACR Trans. Symmetric Cryptol.*, 2020(4):104–129, 2020. I

[13] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.*, 34(3):33, 2021. IV-C

[14] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2010. I

[15] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3g telephony. *J. Cryptol.*, 27(4):824–849, 2014. I

[16] Patricia Greene, Mark Motley, and Bryan Weeks. ARADI and LLAMA: low-latency cryptography for memory encryption. *IACR Cryptol. ePrint Arch.*, page 1240, 2024. I, V-A1

[17] Hosein Hadipour, Simon Gerhalter, Sadegh Sadeghi, and Maria Eichlseder. Improved search for integral, impossible differential and zero-correlation attacks application to ascon, forkskinny, skinny, mantis, PRESENT and qarmav2. *IACR Trans. Symmetric Cryptol.*, 2024(1):234–325, 2024. I, 29

[18] Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder. Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 128–157. Springer, 2023. I

[19] Xichao Hu, Lin Jiao, Dengguo Feng, Yonglin Hao, Xinxin Gong, Yongqiang Li, and Siwei Sun. Impossible boomerang distinguishers revisited. Cryptology ePrint Archive, Paper 2024/1008, 2024. I, II-B, 2

[20] Sunyeop Kim, Insung Kim, Dongjae Lee, Deukjo Hong, Jaechul Sung, and Seokhie Hong. Byte-wise equal property of ARADI. *IACR Cryptol. ePrint Arch.*, page 1772, 2024. I

[21] Lars Knudsen. Deal - a 128-bit block cipher. In *NIST AES Proposal*, 1998. I

[22] Jiqiang Lu. Cryptanalysis of block ciphers. *PhD thesis, University of London UK, 2008.* I, 1

[23] Jiqiang Lu. The (related-key) impossible boomerang attack and its application to the AES block cipher. *Des. Codes Cryptogr.*, 60(2):123–143, 2011. I, I, 1, 2, 1

[24] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 370–386. Springer, 2008. I, 3, II-C

[25] Sean Murphy. The return of the cryptographic boomerang. *IEEE Trans. Inf. Theory*, 57(4):2517–2521, 2011. I

[26] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Lightweight authenticated encryption mode suitable for threshold implementation. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 705–735. Springer, 2020. I, V-B1

[27] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Secret can be public: Low-memory AEAD mode for high-order masking. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 315–345. Springer, 2022. I

[28] Ling Song, Huimin Liu, Qianqian Yang, Yincen Chen, Lei Hu, and Jian Weng. Generic differential key recovery attacks and beyond. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VII*, volume 15490 of *Lecture Notes in Computer Science*, pages 361–391. Springer, 2024. I

[29] Haoyang Wang and Thomas Peyrin. Boomerang switch in multiple rounds. application to AES variants and deoxys. *IACR Trans. Symmetric Cryptol.*, 2019(1):142–169, 2019. I

[30] Senpeng Wang, Dengguo Feng, Tairong Shi, Bin Hu, Jie Guan, Kai Zhang, and Ting Cui. New methods for bounding the length of impossible differentials of SPN block ciphers. *IEEE Trans. Inf. Theory*, 70(12):9165–9178, 2024. I

[31] Jianing Zhang and Haoyang Wang. Optimizing key recovery in impossible cryptanalysis and its automated tool. *IACR Cryptol. ePrint Arch.*, page 158, 2025. 1

[32] Jianing Zhang, Haoyang Wang, and Deng Tang. Impossible boomerang attacks revisited applications to deoxys-bc, joltik-bc and SKINNY. *IACR Trans. Symmetric Cryptol.*, 2024(2):254–295, 2024. I, I, 1, II-B, II-D, II-D

[33] Kai Zhang, Senpeng Wang, Xuejia Lai, Lei Wang, Jie Guan, Bin Hu, and Tairong Shi. Impossible differential cryptanalysis and a security evaluation framework for AND-RX ciphers. *IEEE Trans. Inf. Theory*, 70(8):6025–6040, 2024. I