

Impossible Polytopic Attack Revisited: Low-Data Distinguishers and Attacks

Yongqiang Li

Abstract

Block ciphers including several variants of the well-known AES, the newly proposed tweakable block cipher Deoxys-BC (standardized by ISO/IEC and renamed Deoxys-TBC), and ChiLow (EUROCRYPT 2025) adopt key sizes larger than their block sizes. These designs offer security higher than the block size. This paper evaluates the security of such ciphers by revisiting the Impossible Polytopic Attack (IP attack, proposed by Tyge Tiessen at EUROCRYPT 2016). We show that IP attack can build longer-round distinguishers, enabling attacks on more rounds. Moreover, the attack is applicable under the known-plaintext (KP) setting. Towards this end, we first formalize the distinguisher from Tiessen’s original work and establish a generic framework for distinguisher construction. We further propose two novel methods to lower the corresponding construction complexity. Moreover, we develop two dedicated key-recovery techniques, namely the plaintext-grouping technique and the partition-guess-filter technique. The former allows cryptanalysis on more rounds of target ciphers, while the latter substantially lowers the overall attack complexity. Finally, we build the first framework for IP attack. We apply our method to the chosen-plaintext/ciphertext (CP/CC) and KP scenarios under the single-key setting. As a result, we obtain new distinguishers and attacks against AES, Deoxys-BC, Joltik-BC, LED-128, and ChiLow-32. Notably, 10-round attacks are constructed on Deoxys-BC-384 and Joltik-BC-192. Compared with impossible differential attacks, which are closely related and extensively studied, the proposed results outperform such attacks by one round. Furthermore, a novel full-round attack on ChiLow-32 is constructed under the KP setting, achieving the state-of-the-art attack with optimal data complexity and overall complexity.

Index Terms

Impossible polytopic attack, Impossible polytopic transition, Structure evaluation, Block cipher

I. INTRODUCTION

BLOCK ciphers with a key size larger than their block size are a prevalent design paradigm. Prominent examples include two variants of AES [1], namely AES-192 and AES-256, two instances of Deoxys-BC [2]—the underlying tweakable block ciphers of Deoxys-II, a portfolio-selected scheme from the CAESAR competition—namely Deoxys-BC-256 and Deoxys-BC-384, and ChiLow [3], proposed at EUROCRYPT 2025 and tailored for code encryption. These designs achieve a security level exceeding the block size, owing to their larger key size.

The IP attack [4] was proposed by Tyge Tiessen at EUROCRYPT 2016. It achieves prominent improvements for low-data attacks, especially when targeting standardized block ciphers including AES and DES [5]. The IP attack is a variant of the Differential attack [6] and can also be regarded as a generalization of the Impossible Differential attack (ID attack) [7]. Whereas the ID attack exploits incompatibility between one or multiple pairs of input and output differences, the IP attack captures incompatibility across one or multiple pairs of input and output d -differences. Here, d denotes a positive integer, and a d -difference refers to a tuple of d differences. Much like the Differential attack and ID attack, the IP attack proceeds in two sequential stages: constructing a distinguisher (named impossible polytopic transition, IP transition for simplicity) and extending it over additional rounds to recover the key.

Distinguishers constitute a fundamental and versatile tool in the cryptanalysis of modern block ciphers, serving two primary applications. The first and most conventional scenario is key-recovery attacks. The second is to quantify the statistical discrepancy between a target primitive and its ideal random permutation, a direction extensively explored in AES-related works, including the multiple-of- n property [8], the yoyo trick [9], and the exchange attack [10]. Consequently, dedicated construction methods for distinguishers are of particular importance. To date, there exist two primary methodologies for constructing IP transitions.

- Tiessen’s method: This manual derivation technique is a bound-estimation-based method; i.e., it propagates input d -differences round by round to enumerate all feasible output d -differences. Given the full set of reachable output d -differences, one can deduce the impossible output d -differences.
- State-based method [11], [12]: This is an SAT/SMT-based automatic approach that incorporates the full cipher specifications. It has yielded novel distinguishers for lightweight block ciphers.

Key-recovery techniques constitute another critical factor affecting the overall attack performance. For IP attack, Tyge Tiessen laid the foundational pioneering work and proposed low-data attacks against AES and DES, demonstrating the advantages of this attack. This research direction still holds substantial potential for further investigation.

Yongqiang Li is with State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and is also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China, email: xchao_h@163.com

Motivation. First, block ciphers with a key size larger than their block size can achieve a security level exceeding their block size. In particular, Deoxys-BC-384 and ChiLow-32 adopt key sizes three or four times their block size. This feature provides extra degrees of freedom for adversaries. Unlike the state-based method, which is mainly applicable to lightweight block ciphers with small d , Tiessen’s method offers greater generality at the cost of higher computational complexity and performs better in high-security-level scenarios. Although Tiessen’s method has been successfully applied to the attack, the corresponding research is far from systematic. Therefore, it is imperative to establish rigorous formal definitions and develop generalized and optimized construction methods. Second, existing key-recovery techniques extend cipher rounds only on a single side of the distinguisher to achieve low-data attacks. To mount attacks over more cipher rounds, the more general two-sided extension technique remains to be further explored. Third, existing key-recovery techniques adopt a one-time full-key guessing strategy, which may lead to high attack complexity. How to effectively reduce the computational complexity remains a worthwhile and nontrivial problem.

Our Contributions. We conduct an in-depth study on IP attack. Regarding distinguisher construction, building upon Tyge Tiessen’s original idea, we formally define bound-estimation-based IP transition and propose a general framework for distinguisher construction. We further develop two strategies to lower the complexity of constructing IP transition: the large-Sbox strategy and the subtlety-aware strategy. The former treats the unkeyed component as an S-box to reduce the set of possible output d -differences, while the latter conducts refined analysis of nonlinear functions to minimize feasible output d -differences via careful optimization of d -difference propagation.

Regarding attacks over longer rounds, we propose the *plaintext-grouping technique*, which builds a plaintext-grouping table based on the keys to be guessed in these pre-distinguisher extended rounds. When mounting the attack, we pre-guess these keys and query the table to locate plaintexts such that the d -difference after encryption over the extended rounds matches the input d -difference of the distinguisher. Consequently, we can launch key-recovery attacks on both sides of the distinguisher and cover more rounds.

Regarding key guessing techniques, we propose the *partition-guess-filter technique*, which partitions candidate keys according to the positions of output d -differences and performs stepwise key guessing and filtering to further reduce the overall attack complexity. Let K_b and K_f denote the sets of guessed keys for the rounds extended before and after the distinguisher, respectively, and let L_K denote the master key size. This technique allows $|K_b \cup K_f| = L_K$ to hold, implying that the entire key can be guessed in the key-recovery phase. For ID attacks, as mentioned in [13]: “It is important to note that in an impossible differential attack, the condition $|K_b \cup K_f| < L_K$ must hold”. While IP attacks and ID attacks bear close similarities, as both leverage incompatible relations between input and output differences, the former employs d -differences while the latter relies on a single difference. Powered by the partition-guess-filter technique, the IP attack attack supports full-key recovery, which potentially allows more rounds to be extended before and after the distinguisher.

By integrating these new techniques, we establish a framework for key-recovery attacks. It is observed that IP attacks can work not only in CP/CC scenarios but also in KP scenarios. We apply our method to AES, Deoxys-BC, Joltik-BC [14], LED-128 [15], and ChiLow-32 in the single-key setting for two aims: *low-data* and *low-complexity*. The former aims to construct distinguishers or implement key-recovery attacks with minimal data, while the latter pursues the lowest possible overall complexity. As a result, we obtain the following new results regarding distinguishers and corresponding attacks. Given the strong similarities between IP attacks and ID attacks and substantial prior work on ID distinguishers (the distinguishers of ID attacks) and ID attacks, the results of ID distinguishers and ID attacks serve as the primary baseline for comparison in this work.

In the CP/CC setting, the distinguishers are shown in Table I. Only the distinguisher for ChiLow-32 is constructed under the CC setting.

- Currently, the longest key-recovery impossible differential for AES and AES-like ciphers cover 4 rounds. For Deoxys-BC-384, Joltik-BC-192, LED-128, AES-256, Deoxys-BC-256, and Joltik-BC-128, our distinguishers exceed their impossible differential counterparts by 1.5, 1.5, 3.5, 0.5, 0.5, and 0.5 rounds, respectively. In particular, for LED-128, whose round keys are applied every four rounds, the large-Sbox strategy is adopted.
- ChiLow-32 is a tweakable block cipher proposed at EUROCRYPT 2025, which allows adversaries to launch attacks under either a single tweakkey or multiple tweakkeys. Furthermore, the data complexity is capped at 2^8 in the single-tweakkey setting. We construct the distinguisher by adopting the subtlety-aware strategy under the single-tweakkey setting. As shown in Table I, the distinguisher we obtain is the longest known distinguisher under the single-tweakkey setting to the best of our knowledge, and it requires minimal data.

In the CP setting, the attacks are shown in Table II.

- For Deoxys-BC-384 and Joltik-BC-192, we construct 10-round IP attacks. To the best of our knowledge, our result extends the ID attack by one round. This is the first demonstration that IP attack achieves a higher round count than the ID attack among all known results. The core reason is that our IP attack exploits a 5-round distinguisher, while the ID attack is limited to a 4-round distinguisher.

TABLE I: Our distinguishers in the CP/CC setting (ID: impossible differential, IP: IP transition, LC: low-complexity distinguisher, LD: low-data distinguisher. Our distinguisher can distinguish $\#R + 0.5$ rounds of the cipher from a random permutation.)

Cipher	#R	Data	Time	Mem.	Method	Type	Ref
AES/AES-like	4	-	-	-	ID	-	[16], [17]
Deoxys-BC-384	5.5	44	$2^{338.90}$	$2^{337.43}$	IP	LC	Section IV-A
	5.5	13	$2^{362.93}$	$2^{361.59}$	IP	LD	Section IV-A
Joltik-BC-192	5.5	16	$2^{177.94}$	$2^{177.50}$	IP	LC	Section IV-B
	5.5	13	$2^{181.96}$	$2^{181.59}$	IP	LD	Section IV-B
LED-128	7.5	15	$2^{108.86}$	$2^{108.81}$	IP	LC	Section IV-C
	7.5	9	$2^{116.92}$	2^{117}	IP	LD	Section IV-C
AES-256 / Deoxys-BC-256	4.5	28	$2^{210.47}$	$2^{208.76}$	IP	LC	Section IV-E
	4.5	49	$2^{234.56}$	2^{233}	IP	LD	Section IV-E
Joltik-BC-128	4.5	15	$2^{109.53}$	$2^{108.81}$	IP	LC	Section IV-F
	4.5	9	$2^{117.59}$	2^{117}	IP	LD	Section IV-F
ChiLow-32	3	2^{19}	2^{19}	-	Differential	-	[3]
	3	2^{28}	2^{28}	-	Linear	-	[3]
	3.5	2^8	-	-	Integral	-	[18]
	4.5	11	$2^{121.61}$	$2^{119.02}$	IP	-	Section IV-D

TABLE II: Our attacks in the CP setting (ID: impossible differential attack, MITM: Meet-in-the-Middle attack, IP: IP attack, LC: low-complexity attack, LD: low-data attack, †: These results are obtained by implementing and running the one-step method proposed in [19], and represent the best known results of ID attacks to date.)

Cipher	#R	Data	Time	Mem.	Method	Type	Ref
Deoxys-BC-384	9	2^{124}	$2^{374.83}$	2^{245}	ID†	-	Appendix B
	10	2^{32}	$2^{331.89}$	$2^{329.61}$	IP	LC	Section IV-A
	10	2^{24}	2^{360}	$2^{369.61}$	IP	LD	Section IV-A
Joltik-BC-192	9	2^{64}	$2^{186.83}$	2^{125}	ID†	-	Appendix B
	10	2^{16}	2^{180}	$2^{185.61}$	IP	-	Section IV-B
Deoxys-BC-256	8	2^{117}	2^{248}	2^{182}	ID†	-	Appendix C
	8	2^{32}	$2^{203.04}$	$2^{201.04}$	IP	LC	Section IV-E
	8	2^{24}	2^{232}	$2^{241.04}$	IP	LD	Section IV-E
Joltik-BC-128	8	2^{60}	2^{123}	2^{125}	ID†	-	Appendix C
	8	2^{16}	2^{116}	$2^{121.04}$	IP	-	Section IV-F
AES-256	7	2^{26}	2^{146}	2^{40}	MITM	-	[20]
	8	2^{107}	2^{196}	2^{96}	MITM	-	[21]
	8	2^{113}	2^{196}	2^{129}	ID	-	[22]
	8	2^{32}	$2^{195.20}$	$2^{201.04}$	IP	LC	Section IV-G
	8	2^{24}	2^{232}	$2^{241.04}$	IP	LD	Section IV-G
AES-192	7	2^{26}	$2^{146.3}$	2^{40}	Multiple-of-8	-	[20]
	7	2^{103}	2^{103}	2^{129}	ID	-	[22]
	7	2^{16}	2^{176}	$2^{152.39}$	IP	-	Section IV-H

- For Deoxys-BC-256 and Joltik-BC-128, we construct 8-round IP attack attacks. Our attack matches the maximum round count of existing ID attacks, yet achieves lower data and overall complexity.
- For the 7-round attack on AES-192 and the 8-round attack on AES-256, the state-of-the-art low-data attacks are MITM and Multiple-of-8 attacks, respectively. Our work achieves the lowest data complexity among all known 7-round attacks on AES-192 and 8-round attacks on AES-256.

In the KP setting, the distinguishers and attacks are shown in Table III. For distinguishers, ID attacks are typically constructed under the CP or CC setting. Two prominent attack families working in the KP setting are MITM and linear attacks. Nevertheless, MITM attacks generally do not rely on KP distinguishers — the δ -set construct qualifies as a distinguisher, yet it is built under CP or CC settings [24]. Meanwhile, distinguishers for linear attacks usually incur high complexity. In this context, our method yields the first low-data distinguisher in the KP setting, to the best of our knowledge. Moreover, for ChiLow-32, linear distinguishers represent the longest known KP distinguishers. Compared with these linear distinguishers, our method yields distinguishers with more rounds and lower data complexity.

For KP IP transitions, building upon the KP distinguishers and the proposed partition-guess-filter technique, we construct new full-round attacks on ChiLow-32. These attacks attain optimal overall complexity and the minimal data

TABLE III: Our distinguishers and attacks in the KP setting (LC: low-complexity distinguisher or attack, LD: low-data distinguisher or attack. Our distinguisher can distinguish $\#R + 0.5$ rounds of the cipher from a random permutation.)

Cipher	#R	Data	Time	Mem.	Method	Type	Ref
Distinguishers							
Deoxys-BC-384	3.5	39	$2^{293.30}$	$2^{297.25}$	IP	LC	Section V-B
	3.5	9	$2^{371.63}$	$2^{369.58}$	IP	LD	Section V-B
AES-256 and Deoxys-BC-256	2.5	22	$2^{170.9}$	$2^{168.46}$	IP	LC	Section V-C
	2.5	7	$2^{243.44}$	$2^{241.17}$	IP	LD	Section V-C
AES-192	2.5	22	$2^{170.9}$	$2^{168.46}$	IP	LC	Section V-D
	2.5	9	$2^{187.04}$	$2^{184.59}$	IP	LD	Section V-D
ChiLow-32	3	2^{28}	2^{28}	-	Linear	-	[3]
	3.5	9	$2^{115.85}$	$2^{112.91}$	IP	-	Section V-A
Attacks							
ChiLow-32	4	-	2^{126}	-	MITM	-	[3]
	6	$2^{6.32}$	$2^{76.55}$	-	Integral	-	[18]
	7	$2^{6.32}$	$2^{108.55}$	-	Integral	-	[18]
	Full-round	160	$2^{120.34}$	$2^{98.32}$	MITM	-	[23]
	Full-round	32	$2^{114.87}$	$2^{114.90}$	IP	LC	Section V-A
	Full-round	13	$2^{124.35}$	$2^{113.49}$	IP	LD	Section V-A

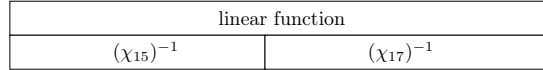


Fig. 1: Structural view on the inverse of $\mathbb{X}_{32}$

complexity known to date. Notably, ChiLow-32 adopts $\mathbb{X}_{32}$ as its nonlinear function, which is a parallel concatenation of χ_{15} and χ_{17} [25], followed by a linear function. In our attack, we study the propagation of d -difference through $(\mathbb{X}_{32})^{-1}$. The structural view on $(\mathbb{X}_{32})^{-1}$ is shown in Figure 1. We only treat $(\chi_{15})^{-1}$ and $(\chi_{17})^{-1}$ as large S-boxes. Thus, replacing this nonlinear function with any function consisting of a parallel concatenation of a 15-bit S-box and a 17-bit S-box followed by a linear function cannot immunize against our attack. It is also worth emphasizing that the proposed attack recovers the complete 128-bit key within the guess-and-determine phase. As previously noted, ID attacks cannot guess the full key in the guess-and-determine phase, whereas the IP attack technique circumvents this limitation. The attack on ChiLow-32 provides a concrete demonstration of this advantage. This property is essential to the success of the attack and validates the practicality of the proposed Partition-Guess-Filter technique.

Outline. Section II covers the preliminaries and related work. The new techniques for distinguisher construction and key-recovery attacks are presented in Section III. In Section IV and Section V, we apply our method to the CP/CC setting and the KP setting, respectively. Finally, Section VI concludes this paper.

II. PRELIMINARIES

A. Notations

Definition 1 ($((d+1)$ -polytope [4]). *An $(d+1)$ -polytope in $\mathbb{F}_2^n$ is an $(d+1)$ -tuple of values in $\mathbb{F}_2^n$.*

Definition 2 (d -difference [4]). *A d -difference $(\alpha_0, \dots, \alpha_{d-1})$ over $\mathbb{F}_2^n$ refers to a d -tuple of elements from $\mathbb{F}_2^n$ that encodes the relative offsets of the plaintexts in a $(d+1)$ -polytope $(x_0, \dots, x_d)$, where each component satisfies $\alpha_i = x_0 \oplus x_{i+1}$ for $0 \leq i \leq d-1$.*

Based on the two definitions above, we fix the notation used throughout this paper as follows.

- $\mathbf{X}^r$: $\mathbf{X}^r = (X_0^r, \dots, X_d^r)$ denotes the $(d+1)$ -polytope at round r , where X_ℓ^r stands for the ℓ -th component. Moreover, $X_{\ell,i}^r$ refers to the i -th bit, nibble, or byte of the ℓ -th component. Throughout this paper, bold uppercase Latin letters are adopted to represent $(d+1)$ -polytopes.
- $\boldsymbol{\alpha}^r$: $\boldsymbol{\alpha}^r = (\alpha_0^r, \dots, \alpha_{d-1}^r)$ denotes the d -difference at round r , where α_ℓ^r represents the ℓ -th component. Moreover, $\alpha_{\ell,i}^r$ corresponds to the i -th bit, nibble, or byte of the ℓ -th component. Throughout this paper, bold Greek letters are used to denote d -differences.
- $\triangleright$: The notation $\mathbf{X}^r \triangleright \boldsymbol{\alpha}^r$ means $\alpha_\ell^r = X_0^r \oplus X_{\ell+1}^r$ for all $0 \leq \ell \leq d-1$.
- $\mathbf{X}_{*,i}^r$: $\mathbf{X}_{*,i}^r = (X_{0,i}^r, \dots, X_{d,i}^r)$ denotes the i -th byte or bit of the $(d+1)$ -polytope at round r according to the context. Similarly, the symbol $\boldsymbol{\alpha}_{*,i}^r$ follows the same definition.
- K^r and $K^{eq,r}$: K^r denotes the round key at round r , and $K^{eq,r}$ represents the round key obtained by commuting the key XOR operation and the MixColumns operation of AES-like ciphers.

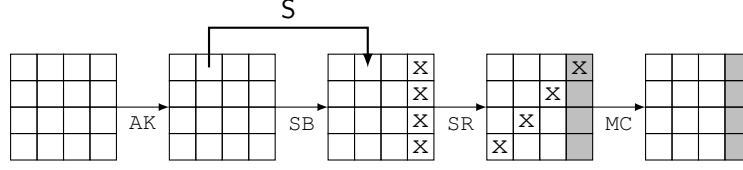


Fig. 2: One round of block cipher AES

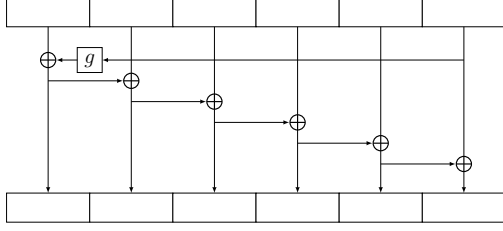


Fig. 3: The key schedule of AES-192

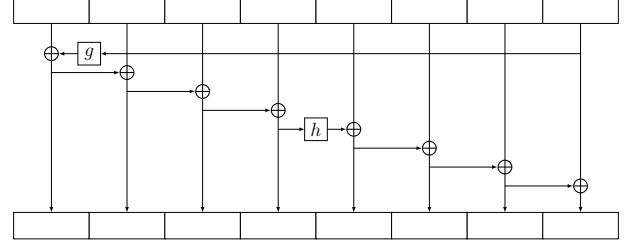


Fig. 4: The key schedule of AES-256

- $\mathbf{X}''$ and α'' : When commuting the key XOR operation and the MixColumns operation, the corresponding $(d+1)$ -polytope and d -difference after the key XOR operation are denoted as $\mathbf{X}''$ and α'' .
- Active bit/byte: For a d -difference, a bit or byte is called to be active if it remains active in every constituent difference.
- Target bit/byte: For the output d -difference, we usually focus only on partial output bits or bytes. Thus, a target bit or byte of an output d -difference refers to the corresponding bit or byte of interest for every constituent difference. For an intermediate d -difference, a bit or byte is called a target bit or byte if it influences the output target d -difference.

The color convention adopted in the figures of this paper is as follows.

- ■: Known d -differences.
- ■: Unknown d -differences.
- ■: Intermediate target position. An additional $\checkmark$ is attached for the output target position.
- ■/□: The key, $(d+1)$ -polytope, or d -difference required for key recovery. For clarity, ■ is occasionally used to denote the key bits to be guessed.

B. Specification of AES

AES [1] is among the most prominent block ciphers to date, whose design principles have exerted a lasting influence on the development of block cipher constructions. AES operates on 128-bit blocks and supports key lengths of 128, 192, and 256 bits, with an 8-bit S-box. Following the SPN structure, AES adopts an MDS matrix to guarantee strong diffusion properties. Its internal state can be represented as a square array of bytes as follows, where $s_i \in \mathbb{F}_2^8$ for $0 \leq i \leq 15$.

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}. \quad (1)$$

One encryption round of AES is depicted in Figure 2, and it consists of the following four operations:

- AddRoundKey (AK): The round key derived from the key schedule is XORed with the internal state.
- SubBytes (SB): The 8-bit S-box is applied to each byte of the cipher's internal state in parallel.
- ShiftRows (SR): The i -th row, where $0 \leq i \leq 3$, of the internal state is rotated right by i bytes.
- MixColumns (MC): Each column of the internal state is multiplied by the MDS matrix.

The key schedules of AES-192 and AES-256 are shown in Figure 3 and Figure 4, respectively. The master key is split into 6, and 8 32-bit words for AES-192 and AES-256, respectively, where each word consists of four bytes. These words are subsequently processed to produce the round keys. The function g rotates an input word left by one byte, followed by four parallel S-box lookups. The function h performs four parallel S-boxes on its input word.

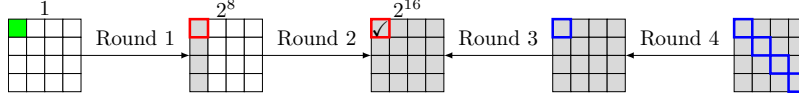


Fig. 5: The 4-round IP attack on AES

C. The 4-round attack on AES in [4]

In this part, we revisit the 4-round IP attack of AES in [4]. As illustrated in Figure 5, consider an input 7-difference that only activates the first byte, meaning all other bytes are zero. Such a 7-difference can propagate into at most 2^8 distinct 7-differences after round 1. When 7-differences are constrained to the first column after round 2, at most 2^{16} distinct 7-differences exist within this column. Specifically, the first byte admits at most 2^{16} distinct 7-differences, as highlighted by the checkmark in Figure 5.

By exchanging the AK and MC operations, 40 bits of key material must be guessed to recover the differences of the first byte at round 2 via ciphertext decryption. For each key guess, if the corresponding 7-difference is not contained within the set of 2^{16} feasible candidates, the guess can be discarded as invalid. There are 2^{56} possible 7-differences over a byte, while only 2^{16} such differences can propagate from the selected input 7-difference. This yields a probability of 2^{-40} for an incorrect key guess to yield a valid 7-difference. On average, 2 candidate guesses for the 40 key bits are retained, including the correct guess. A single additional plaintext-ciphertext pair suffices to validate the correct key candidate.

The data complexity of the attack is bounded by 8 chosen plaintexts. The time complexity is dominated by computing the 7-difference of the first byte after round 2 for every candidate of the 40 key bits and verifying membership within the set of 2^{16} feasible values. On average, fewer than 16 table lookups are required per key guess (binary search table lookup). The resulting time complexity equals $2^{40} \cdot 2^{-2} = 2^{38}$ 4-round AES encryption equivalents, under the assumption that a single 4-round encryption costs $4 \cdot 16$ table lookups. The memory complexity is bounded by a lookup table storing all 2^{16} possible 7-differences for a single byte, which occupies 2^{19} bytes, or equivalently 2^{15} plaintext equivalents.

D. Complexity analysis of table construction and lookup

In this paper, tables are constructed solely to store data of a set and determine whether a given element belongs to the set; thus, the stored data are static. We consider static sorted-array schemes to store a static set of N elements.

- For construction, all elements of the set are written into an array at once, and a sorting algorithm is applied to arrange the array in global ascending or descending order to obtain an immutable static sorted array. The total construction time complexity is $O(N \log N)$, consisting of $O(N)$ time for data writing and $O(N \log N)$ time for sorting, while the memory complexity is $O(N)$.
- To test whether a target element exists in the set, binary search [26], [27] is adopted: we initialize left and right boundaries covering the whole array, repeatedly compare the target with the middle element to halve the search range, and terminate the procedure once the target is matched or the search interval becomes empty. The query time complexity is $O(\log N)$.

Here, we adopt merge sort [26], [27] with time complexity $O(N \log N)$. Each iteration within the merging procedure involves two basic operations: element comparison and element movement. Accordingly, we treat each iteration as equivalent to two table lookups. This equivalence serves as a simplified cost model to unify the measurement of computational overhead.

III. NEW TECHNIQUES FOR DISTINGUISHER CONSTRUCTION AND KEY-RECOVERY ATTACKS

In this section, we propose novel techniques for distinguisher construction and key-recovery attacks. Regarding distinguisher construction, we formally define the bound-estimation-based IP transitions, elaborate its general construction framework, and present two enhancement strategies to improve existing construction methods. For key-recovery attacks, we introduce the plaintext-grouping technique and the partition-guess-filter technique. The former extends the number of attackable rounds, whereas the latter effectively reduces the overall attack complexity. Finally, by integrating these new techniques, we establish a framework for key-recovery attacks.

A. Bound-estimation-based distinguisher revisited

Apart from key-recovery attacks, distinguishers can be leveraged to tell block ciphers apart from random permutations. The bound-estimation-based method, originally proposed by Tiessen, is primarily utilized to construct distinguishers for key-recovery attacks [4]. Nevertheless, this distinguisher still possesses great untapped research

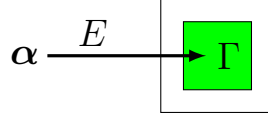


Fig. 6: Illustration for bound-estimation-based IP transition

potential for ciphers where the key length exceeds the block length. Motivated by this observation, we systematically investigate this distinguisher and first present its formal definition.

Definition 3. Let E denote a block cipher, α an input d -difference, τ the number of target bits of the output d -difference, and Γ the set of all possible output target d -differences. Suppose the size of Γ is bounded by 2^t . Given $d + 1$ plaintext-ciphertext pairs with $d > t$, E can be distinguished from a random permutation with success probability $1 - 2^{t-d\tau}$. The pair (α, Γ) is called a bound-estimation-based IP transition of E , with (d, t, τ) referred to as its parameters.

In this paper, the notation IP transition always denotes bound-estimation-based IP transition. In contrast to classical distinguishers such as differentials, truncated differentials and impossible differentials, constructing such distinguishers incurs both time and memory complexity. A valid distinguisher cannot exceed the cipher's claimed security complexity. For convenience, we take AES-like cipher as an example to illustrate the complexity-calculation method.

Let $E = E_{R-1} \circ \dots \circ E_1 \circ E_0$ denote an R -round block cipher. Two propagation trails are obtained by forward-propagating the active bytes from α and backward-propagating the target bytes from Γ , respectively.

$$\begin{aligned} \text{Upper trail : } \mathcal{A}_0 &\xrightarrow[f]{E_0} \mathcal{A}_1 \xrightarrow[f]{E_1} \mathcal{A}_2 \xrightarrow[f]{E_2} \dots \xrightarrow[f]{E_{R-2}} \mathcal{A}_{R-1} \xrightarrow[f]{E_{R-1}} \mathcal{A}_R, \\ \text{Lower trail : } \mathcal{T}_0 &\xrightarrow[b]{E_0} \mathcal{T}_1 \xrightarrow[b]{E_1} \mathcal{T}_2 \xrightarrow[b]{E_2} \dots \xrightarrow[b]{E_{R-2}} \mathcal{T}_{R-1} \xrightarrow[b]{E_{R-1}} \mathcal{T}_R. \end{aligned}$$

Here, $\mathcal{A}_r$ denotes the set of positions of active bytes at round r , where $\mathcal{A}_0$ is derived from the active bytes of the input d -difference, and $\mathcal{A}_{r+1} = \{j \mid \text{the } j\text{-th byte is affected by bytes in } \mathcal{A}_r\}$. Similarly, $\mathcal{T}_r$ denotes the set of positions of target bytes at round r , where $\mathcal{T}_R$ is derived from the target bytes of the output d -difference, and $\mathcal{T}_r = \{j \mid \text{the } j\text{-th byte affects bytes in } \mathcal{T}_{r+1}\}$.

By selecting the switch round R' , we derive all possible d -differences at round R' from the input d -difference following the upper trail; then the set Γ can be computed from these all possible d -differences at round R' according to the lower trail, since we are only interested in the number of output target d -differences. For $r \leq R'$, let Γ_r denote the number of all possible d -differences at round r ; for $r > R'$, let Γ_r denote the number of all possible target d -differences at round r . Then,

$$\begin{cases} |\Gamma_0| &= 2^0, \\ |\Gamma_1| &= |\Gamma_0| 2^{s|\mathcal{A}_0|} = 2^{s|\mathcal{A}_0|}, \\ &\dots, \\ |\Gamma_{R'}| &= |\Gamma_{R'-1}| 2^{s|\mathcal{A}_{R'-1}|} = 2^{s(\sum_{r=0}^{R'-1} |\mathcal{A}_r|)}, \\ |\Gamma_{R'+1}| &= |\Gamma_{R'}| 2^{s|\mathcal{T}_{R'}|} = 2^{s(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + |\mathcal{T}_{R'}|)}, \\ |\Gamma_{R'+2}| &= |\Gamma_{R'+1}| 2^{s|\mathcal{T}_{R'+1}|} = 2^{s(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + \sum_{r=R'}^{R'+1} |\mathcal{T}_{R'}|)}, \\ &\dots, \\ |\Gamma_R| &= |\Gamma_{R-1}| 2^{s|\mathcal{T}_{R-1}|} = 2^{s(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + \sum_{r=R'}^{R-1} |\mathcal{T}_r|)}. \end{cases}$$

Thus, $|\Gamma| = |\Gamma_R| = 2^{s(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + \sum_{r=R'}^{R-1} |\mathcal{T}_r|)}$. Moreover, since $|\Gamma_r| < |\Gamma_{r+1}|$, the memory complexity is dominated by storing all d -differences in Γ . Namely,

$$\mathcal{MC}_{\text{dom}}^D = d \times s \times |\mathcal{T}_r| \times |\Gamma|$$

bits.

Let $\mathcal{TC}_r$ denote the time complexity for computing Γ_r . Then,

$$\mathcal{TC}_r = \begin{cases} |\Gamma_{r-1}| 2^{s|\mathcal{A}_{r-1}|} (d+1) |\mathcal{A}_{r-1}| \phi_{r-1}^D = 2^{s(\sum_{r'=0}^{r-1} |\mathcal{A}_{r'}|)} (d+1) |\mathcal{A}_{r-1}| \phi_{r-1}^D, & \text{if } 1 \leq r \leq R', \\ |\Gamma_{r-1}| 2^{s|\mathcal{T}_{r-1}|} (d+1) |\mathcal{T}_{r-1}| \phi_{r-1}^D = 2^{s(\sum_{r'=0}^{R'-1} |\mathcal{A}_{r'}| + \sum_{r'=R'}^{r-1} |\mathcal{T}_{r'}|)} (d+1) |\mathcal{T}_{r-1}| \phi_{r-1}^D, & \text{if } r > R'. \end{cases}$$

Besides, we also need to sort the elements of Γ , and the corresponding time complexity is

$$\mathcal{TC}'_R = s \left(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + \sum_{r=R'}^{R-1} |\mathcal{T}_r| \right) 2^s \left(\sum_{r=0}^{R'-1} |\mathcal{A}_r| + \sum_{r=R'}^{R-1} |\mathcal{T}_r| \right) \phi_R^{D'}.$$

Here, ϕ_r^D ($1 \leq r \leq R$) and $\phi_R^{D'}$ denote the equivalent factors with respect to the operation count of the encryption function. Since $\mathcal{TC}_r < \mathcal{TC}_{r+1}$, the overall time complexity is dominated by $\mathcal{TC}_{\text{dom}}^D = \mathcal{TC}_R + \mathcal{TC}'_R$.

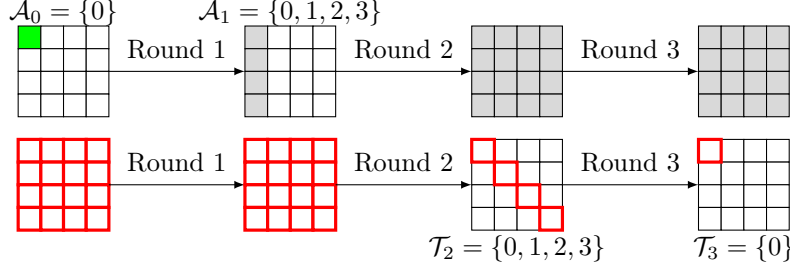


Fig. 7: Example for calculating complexity

Example 1. Using the 3-round distinguisher that is employed to mount a 5-round attack in [4] as an example, where $d = 14$ is set in the original work. As shown in Figure 7, the active byte of the input d -differences is 0, and the target byte of the input d -differences is 0. By choosing $R' = 2$, we obtain $|\Gamma| \leq 2^{72}$. Then 3-round AES can be distinguished from a random permutation with success probability $1 - 2^{72-14 \times 8} = 1 - 2^{-40}$. The memory complexity is $14 \times 8 \times 1 \times 2^{72}/128 \approx 2^{71.81}$ block sizes, and the time complexity is $2^{40} \times 2^{32} \times 15 \times 4/(3 \times 16) + 72 \times 2^{72} \times 2/(3 \times 16) \approx 2^{74.09}$ 3-round encryptions.

The choice of the parameter R' is closely related to the complexity. We can iterate over all possible values of R' to achieve the minimal overall complexity. Although the above method is illustrated with AES-like ciphers, it is also applicable to ciphers with S-boxes of different sizes. Meanwhile, for ciphers without S-boxes, their nonlinear components are treated as equivalent S-boxes for computational evaluation. For instance, this strategy is adopted for ChiLow-32 in Section IV-D and Section V-A.

To improve the complexity for constructing the distinguisher, we propose two strategies to evaluate $|\Gamma_r|$ ($r < R'$) more precisely.

- **The large-Sbox strategy.** Unlike round-by-round propagation, several consecutive unkeyed rounds of the cipher can be viewed as a single large S-box. Suppose the size of this S-box is s' . Given an input d -difference, the number of all possible output d -differences is at most $2^{s'}$, which yields a tighter bound on the output d -differences compared with round-by-round propagation, if the size of the resulting output d -difference set is smaller. In Section IV-C, we apply this technique to LED-128 and successfully construct 7.5-round distinguishers.
- **The subtlety-aware strategy.** Given an input d -difference, we take the fine-grained details of nonlinear functions into account to generate valid output d -differences. This enables a more precise characterization of all possible d -differences emerging in the first several rounds. In Section IV-D, we apply this technique to ChiLow-32 and successfully construct a 5-round distinguisher.

B. The plaintext-grouping technique

To achieve minimal data complexity, the original work introduces an elegant paradigm in which key-recovery attacks are performed exclusively at the bottom of the distinguisher. This well-engineered strategy yields low data complexity, yet inherently enforces an upper bound on the total number of attackable rounds.

Aiming to support attacks covering more rounds, researchers attempt to conduct key-recovery attacks on both sides of the distinguisher to mitigate this inherent limitation. Unfortunately, directly appending extra rounds to the top of the distinguisher yields a large set of candidate input d -differences. The resulting data complexity becomes prohibitively high even for a small number of appended rounds. To overcome this bottleneck, the plaintext-grouping technique is exploited to extend the number of rounds preceding the distinguisher.

Let Θ denote the set consisting of all components of the d -difference. Specifically, any d differences selected from Θ serve as valid input d -differences for the distinguisher. Let r_b be the number of additional rounds appended to the top of the distinguisher. By propagating each $\theta \in \Theta$ backward through r_b encryption rounds, we derive the corresponding active input bits. We use ν to represent the number of active bits induced by each difference $\theta \in \Theta$.

Let $\mathbb{S}$ denote the plaintext set wherein the aforementioned ν active bits take arbitrary values and all remaining bits are fixed. For a fixed plaintext $P^0 \in \mathbb{S}$, let K be the key that yields the difference θ for plaintexts in $\mathbb{S}$, and let

$E_K^{r_b}$ denote the corresponding partial encryption function over r_b rounds. Then, we can construct a plaintext-grouping table.

Definition 4. A plaintext-grouping table is a table for classifying the plaintext set according to key values, where the K -th row is indexed by K and defined as $\mathcal{P}_K = \{P^1 \in \mathbb{S} \mid E_K^{r_b}(P^0) \oplus E_K^{r_b}(P^1) = \theta, \theta \in \Theta\}$.

Given P^0 and $\theta \in \Theta$, there exists exactly one P^1 such that $E_K^{r_b}(P^0) \oplus E_K^{r_b}(P^1) = \theta$, which directly follows from rewriting the equality as $E_K^{r_b}(P^1) = E_K^{r_b}(P^0) \oplus \theta$. Meanwhile, for this unique P^1 , the difference $P^0 \oplus P^1$ propagates to θ over the first r_b rounds. Consequently, P^1 differs from P^0 in exactly ν bits, meaning $P^1 \in \mathbb{S}$. Hence, there is exactly one $P^1 \in \mathbb{S}$ satisfying $E_K^{r_b}(P^0) \oplus E_K^{r_b}(P^1) = \theta$. As a result, $\Pr[E_K^{r_b}(P^0) \oplus E_K^{r_b}(P^1) = \theta] = 2^{-\nu}$, and $|\mathcal{P}_K| = |\Theta|$.

In summary, the plaintext set can be grouped into $2^{|K|}$ subsets, each of size $|\Theta|$. The size of the plaintext-grouping table is $|\Theta| \cdot 2^{|K|}$, and the time complexity for constructing this table is $2^{|K|+\nu} \phi^B$, where ϕ^B denotes the computational cost equivalent to one full encryption operation. Here, we do not require additional sorting, since the table inherently stores its entries in an order dictated by the value of K . The plaintext-grouping table permits at most $|\Theta|$ valid data pairs to be constructed per key guess. For $d \leq |\Theta|$, key candidates are pre-enumerated, and $d+1$ data pairs satisfying the input d -difference constraint of the distinguisher are gathered. With these valid data pairs, additional rounds can be prepended to the distinguisher to mount the extended attack.

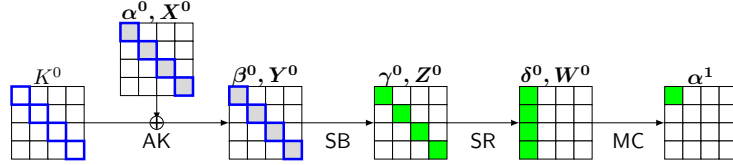


Fig. 8: Add one round at the top of the example IP transition

Example 2. Based on the illustrative example in Section II-C, we append one additional round to the top of the distinguisher, as visualized in Figure 8. Since α^1 is a 7-difference with all differential components active at the 0-th byte, we define Θ as the set of all such possible differences, where $|\Theta| = 2^8 - 1$. By propagating each $\theta \in \Theta$ backward over one round, every differential $\theta \in \Theta$ induces exactly 32 active input bits. Let $K = K_0^0 \| K_5^0 \| K_{10}^0 \| K_{15}^0 \in \mathbb{F}_2^{32}$. We fix the plaintext $P^0 = P_0^0 \| P_1^0 \| P_2^0 \| P_3^0 \in \mathbb{F}_2^{32}$, while $P^1 = P_0^1 \| P_1^1 \| P_2^1 \| P_3^1 \in \mathbb{F}_2^{32}$ is allowed to take arbitrary values over the active bits. Let E_K^1 denote the one-round encryption function. The corresponding plaintext set can then be grouped as $\mathcal{P}_K = \{(P^0, P^1) \mid E_K^1(P^0) \oplus E_K^1(P^1) = \theta, \theta \in \Theta\}$, where the probability $p = \Pr[E_K^1(P^0) \oplus E_K^1(P^1) = \theta]$ equals 2^{-32} for any $\theta \in \Theta$. For a fixed key K , there exist $|\Theta| \cdot 2^{32} \cdot 2^{-32}$ valid choices of P^1 that satisfy the one-round differential equation. We further select plaintext pairs $P^i \in \mathcal{P}_K$ for $0 \leq i \leq 7$. For $0 \leq \ell \leq 7$, we set $X_{\ell,0}^0 = P_0^\ell$, $X_{\ell,5}^0 = P_1^\ell$, $X_{\ell,10}^0 = P_2^\ell$, and $X_{\ell,15}^0 = P_3^\ell$, while keeping all other bytes unchanged. The resulting 7-difference after one-round encryption strictly matches the target differential α^1 . By guessing the 32 bits of K^0 and 40 bits of K^4 and K^5 , we construct a 5-round IP attack, which extends the original attack by one round. In this extended attack, the memory complexity remains unchanged, the data complexity is 2^{32} , and the time complexity is 2^{32} times that of the original attack.

C. The partition-guess-filter technique

Contrary to existing key-recovery techniques that guess all required subkeys simultaneously, we propose the partition-guess-filter technique for multi-step key recovery, which further lowers the overall attack complexity.

For the τ output target bits, each bit corresponds to certain subkeys involved in key recovery. These bits can be partitioned into Q groups, where each group is associated with at least one distinct subkey. Assume the i -th group corresponds to ι_i output target bits ($0 \leq i \leq Q-1$). We conduct the key recovery over Q steps. At the i -th step, we exploit $\sum_{j=0}^i \iota_j$ focused bits for key recovery. Suppose that at the i -th step, we need to guess NK_i bits of additional subkeys and sieve out a fraction φ_i ($0 \leq i \leq Q-1$) of the key candidates. Then

$$\varphi_0 = 2^{t-d\iota_0}, \quad \varphi_i = \frac{2^{t-d \sum_{j=0}^i \iota_j}}{2^{t-d \sum_{j=0}^{i-1} \iota_j}} = 2^{-d\iota_i},$$

since we already exploit $\sum_{j=0}^{i-1} \iota_j$ focused bits to recover keys.

The overall time complexity is formulated as

$$\begin{aligned} \mathcal{TC} = & (d+1) \cdot 2^{NK_0} \cdot \phi_0^P + (d+1) \cdot 2^{NK_0\varphi_0+NK_1} \cdot \phi_1^P + \dots \\ & + (d+1) \cdot 2^{((NK_0\varphi_0+NK_1)\varphi_1+\dots)\varphi_{Q-2}+NK_{Q-1}} \cdot \phi_{Q-1}^P. \end{aligned}$$

Here, ϕ_i^P denotes the computational ratio of partial decryption relative to a single full encryption operation.

Parameter selection strategy. In general, owing to the diffusion property of the cipher, extending a small number of rounds after the distinguisher renders each output bit of the distinguisher consistent with many candidate keys. Consequently, 2^{NK_0} is much larger than 2^{NK_i} for all $1 \leq i \leq Q - 1$. Meanwhile, as d increases, each fraction φ_i ($0 \leq i \leq Q - 1$) decreases. Therefore, by selecting an appropriate value of d , the set of inequalities

$$\begin{aligned} NK_0 &> NK_0 \cdot \varphi_0 + NK_1, \\ NK_0 &> ((NK_0 \cdot \varphi_0 + NK_1) \cdot \varphi_1 + NK_2), \\ &\vdots \\ NK_0 &> \left((\cdots ((NK_0 \cdot \varphi_0 + NK_1) \cdot \varphi_1 + NK_2) \cdot \varphi_2 + \cdots) \cdot \varphi_{Q-2} + NK_{Q-1} \right) \end{aligned} \quad (2)$$

may hold. Whenever the above condition is satisfied, every fraction φ_i ($0 \leq i \leq Q - 1$) decreases as d grows. The lower bound of d such that the above condition holds is denoted as L_d . Our goal is to reduce the attack complexity. When the above condition holds, the complexity of the attack can be lowered. Under this conditions, we adopt the partition-guess-filter technique, yielding

$$\mathcal{TC}_{dom} = (d + 1) \cdot 2^{NK_0} \cdot \varphi_0.$$

We will use this technique to break ChiLow-32.

D. The framework of key-recovery attack

Given an r_d -round IP transition, we add r_b rounds before the distinguisher and r_f rounds after the distinguisher to mount the $(r_b + r_d + r_f)$ -round attack. Here, we only analyze the case $r_b \neq 0$. For $r_b = 0$, please refer to Appendix A.

- 1) **Attempt to use the partition-guess-filter technique.** From the output target position,
 - a) Partition the output target positions of the distinguisher into Q groups, where each group corresponds to at least one distinct candidate key. Assume the i -th group corresponds to ι_i target-output bits ($0 \leq i \leq Q - 1$).
 - b) If $Q > 1$ and there exist ι_0 and L_d such that Equation (2) holds, let K_{out} denote the set of keys corresponding to the ι_0 bits. Otherwise, let K_{out} denote the set of keys corresponding to the output target position, and set $L_d = t/\tau + 1$. Then, we impose the constraint $d \geq L_d$.
- 2) **Apply the plaintext-grouping technique.** From the input d -difference α , apply the plaintext-grouping technique in Section III-B. Let Θ be the set consisting of components of the d -difference. Let ν denote the number of active bits induced by each $\theta \in \Theta$. Let K_{in} denote the set of subkeys involved in the r_b rounds. Consider a set of plaintexts $\mathbb{S}$, which consists of plaintexts where the ν bits take arbitrary values, while the remaining bits are fixed. Construct a plaintext-grouping table, where the K -th row is indexed by K and defined as

$$\mathcal{P}_{K_{in}} = \{P^1 \in \mathbb{S} \mid E_{K_{in}}^{r_b}(P^0) \oplus E_{K_{in}}^{r_b}(P^1) = \theta, \theta \in \Theta\}.$$

Then, $Pr[E_{K_{in}}^{r_b}(P^0) \oplus E_{K_{in}}^{r_b}(P^1) = \theta] = 2^{-\nu}$.

- 3) **Perform guess-and-determine on the key.** Guess candidates in K_{in} ,
 - a) Select $d + 1$ plaintexts according to $\mathcal{P}_{K_{in}}$, and obtain the corresponding ciphertexts.
 - b) For keys in $K_{out} \setminus K_{in}$, sieve the candidate keys using the distinguisher.
- 4) **Perform an exhaustive search over the remaining keys.** After the above steps, there remain $p = 2^{t-d\tau}$ candidate keys for $K_{in} \cup K_{out}$. We then perform an exhaustive search over the remaining candidate keys.

Complexity. The data complexity is 2^ν .

The overall memory complexity arises from two independent components.

- The first component corresponds to the distinguisher, yielding a memory cost of $d\tau|\Gamma|\phi_0^M$.
- The second component originates from the lookup table built in Step 2(b), with a memory cost of $|\Theta|2^{|K_{in}|}\phi_1^M$.

Herein, ϕ_0^M and ϕ_1^M denote memory-overhead fractions, measured in units of a single block-cipher operation.

When mounting the attack, we treat distinguisher construction as a precomputation step. Consequently, the time complexity of constructing the distinguisher is not counted towards the attack complexity. Thus, the time complexity stems from four components:

- Step 2, which constructs the tables, incurs a time cost of $2^{|K_{in}+\nu|}\phi_0^T$.
- In Step 3(a), for each candidate key in , we perform table lookups over a table of size $|\Theta|2^{|K_{in}|}$, and we only obtain the result by looking up K . The corresponding time complexity is $2^{|K_{in}|}|K_{in}|\phi_1^T$.
- In Step 3(b), we guess the candidate key out and look up all possible output d -differences contained in Γ . The associated time complexity is $(d + 1) \cdot 2^{|K_{in} \cup K_{out}|}\phi_2^T + 2^{|K_{in} \cup K_{out}|}\log(|\Gamma|)\phi_3^T$.
- Step 4 contributes a time complexity of $2^{|K|-p \cdot |K_{in} \cup K_{out}|}\phi_4^T$.

Herein, $\phi_0^T, \phi_1^T, \phi_2^T, \phi_3^T, \phi_4^T$ denote the equivalent factors corresponding to the operation count of the encryption function.

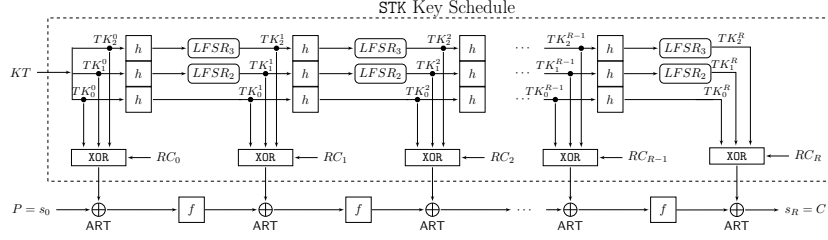


Fig. 9: The TWEAKEY framework for Deoxys-BC-384

IV. THE CP/CC DISTINGUISHERS AND ATTACKS

In this section, we apply our method to Deoxys-BC-384, Joltik-BC-192, LED-128, ChiLow-32, Deoxys-BC-256, Joltik-BC-128, AES-256 and AES-192 in the CP/CC setting.

A. Deoxys-BC-384

Deoxys-BC [2] is an AES-based tweakable block cipher proposed by Jean, Nikolić, and Peyrin within the TWEAKEY framework [28]. It serves as the underlying primitive of the Deoxys authenticated encryption family, a portfolio finalist of the CAESAR competition. Deoxys-BC operates on a 128-bit internal state and combines the secret key and public tweak into a unified input referred to as the *tweakey*. The cipher employs the standard AES round functions, namely SB, SR and MC, and utilizes the Super TWEAKEY (STK) schedule to inject subtweakeys in every round. Deoxys-BC-384 is an instance of Deoxys-BC with a 384-bit tweakey.

As shown in Figure 9, Deoxys-BC-384 employs three tweakey components TK_0 , TK_1 , and TK_2 to generate the subtweakeys. Let STK^r be the round subtweakey. Then, for the i -th cell of the subtweakey after r rounds, we have

$$STK_i^r = TK_{0,h^r[i]} \oplus LFSR_2^r(TK_{1,h^r[i]}) \oplus LFSR_3^r(TK_{2,h^r[i]}).$$

Here, h is a byte permutation defined as

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 6 & 11 & 12 & 5 & 10 & 15 & 0 & 9 & 14 & 3 & 4 & 13 & 2 & 7 & 8 \end{pmatrix}.$$

The notation h^r means applying permutation h r times. $LFSR_2$ and $LFSR_3$ are byte-oriented LFSR functions, and $LFSR_2^r$, $LFSR_3^r$ denote applying the corresponding LFSR r times. Accordingly, knowing $STK_{h^r[i]}^r$ over 3 rounds yields 3 independent equations in the variables $TK_{0,i}$, $TK_{1,i}$, and $TK_{2,i}$, which uniquely determine the master tweakey cells $TK_{0,i}$, $TK_{1,i}$, and $TK_{2,i}$.

Low-complexity 5.5-round distinguishers. Varying the positions of input active 1-bytes and output target 1-bytes gives $16 \times 16 = 256$ distinguishers in total. An example distinguisher is provided below.

Distinguisher 1. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^6 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha_{0,0}^6, \dots, \alpha_{d-1,0}^6)$. As shown in Figure 10, it holds that $|\Gamma| \leq 2^{336}$. By choosing $d = 43$, we can distinguish the 6-round Deoxys-BC-384 from a random permutation with a success probability of $1 - 2^{336-43 \cdot 8} = 1 - 2^{-8}$. The memory complexity for constructing the distinguisher is $43 \times 8 \times 2^{336}/(8 \times 16) \approx 2^{337.43}$ block sizes, and the time complexity is $2^{328} \times 2^8 \times 44/(6 \times 16) + 336 \times 2^{336} \times 2/(6 \times 16) \approx 2^{338.90}$ 6-round encryptions.

Low-data 5.5-round distinguishers. Increasing the number of output target bytes allows us to reduce the data complexity. Accordingly, when the target bytes correspond to the four bytes lying on the anti-diagonal, we obtain $16 \times 4 = 64$ low-data distinguishers. An example distinguisher is provided below.

Distinguisher 2. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^6 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha_{0,0}^6 \parallel \alpha_{0,7}^6 \parallel \alpha_{0,10}^6 \parallel \alpha_{0,13}^6, \dots, \alpha_{d-1,0}^6 \parallel \alpha_{d-1,7}^6 \parallel \alpha_{d-1,10}^6 \parallel \alpha_{d-1,13}^6).$$

As shown in Figure 10, all possible values of β^5 amount to 2^{328} . Thus, $|\Gamma| \leq 2^{328} \times 2^{32} = 2^{360}$. By choosing $d = 12$, we can distinguish 6-round Deoxys-BC-384 from a random permutation with a success probability of $1 - 2^{360-12 \cdot 32} = 1 - 2^{-24}$. The memory complexity required to construct the distinguisher is $12 \times 32 \times 2^{360}/(8 \times 16) \approx 2^{361.59}$ block sizes, and the time complexity consists of $2^{328} \times 2^{32} \times 13/(6 \times 16) + 360 \times 2^{360} \times 2/(6 \times 16) \approx 2^{362.93}$ 6-round encryptions.

Low-data 10-round attack. As illustrated in Figure 11, we exploit a 5-round IP transition distinguisher to construct a 10-round IP attack distinguisher by adding one round before the distinguisher and 4 rounds afterwards. The attack proceeds as follows.

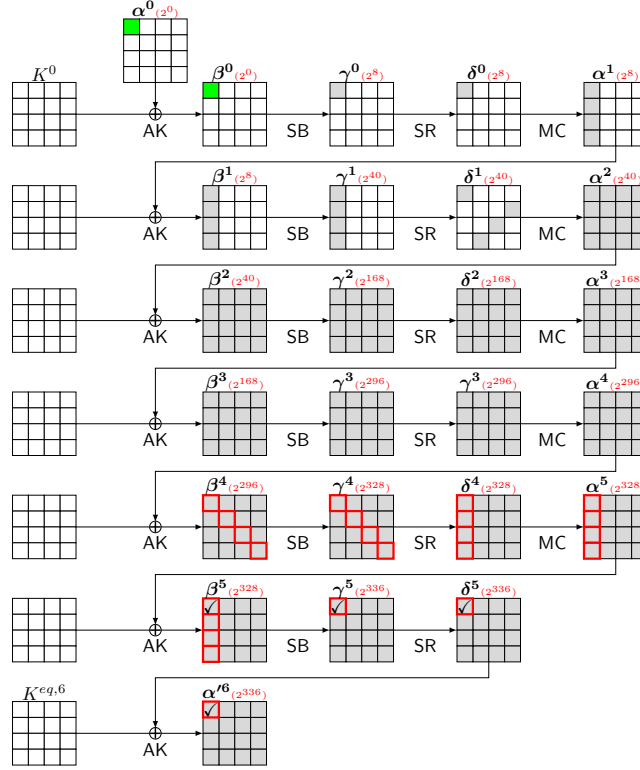


Fig. 10: The 5.5-round IP transition for Deoxys-BC-384

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, and 10-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0, K_{10}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) = 0 \end{cases} \right\}.$$

For each candidate key (K_0^0, K_5^0, K_{10}^0) , there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th and 1-st bytes.

- 2) Guess K_0^0 , K_5^0 and K_{10}^0 , then
 - a) Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d + 1)$ plaintexts yield d -differences that activate only the 0-th and 1-st bytes after one-round encryption, as illustrated by α^1 in Figure 11.
 - b) Obtain the corresponding $(d + 1)$ ciphertexts $(Y_0^{10}, \dots, Y_d^{10})$. Then we guess $K_0^{eq,7}$, $K_0^{eq,8}$, $K_7^{eq,8}$, $K_{10}^{eq,8}$, $K_{13}^{eq,8}$, as well as all bytes of $K^{eq,9}$ and $K^{eq,10}$. Afterwards, we check whether the d -differences $\beta_{*,0}^6$ belong to the set Γ ; any key candidate satisfying this condition is discarded.
- 3) Perform an exhaustive search over the remaining key material.

Choosing $d = 49$, the probability that a d -difference $\beta_{*,0}^6$ belongs to the set Γ equals $2^{368-49 \cdot 8} = 2^{-24}$. Since we guess 320 bits of key, $2^{320-24} = 2^{296}$ key candidates remain. By performing an exhaustive search over the remaining $2^{296+(384-320)} = 2^{360}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{24} . The time complexity is $50 \times 2^{320} \times (3 + 1 + 4 + 16 + 16)/(10 \times 16) + 2^{320} \times 368/(10 \times 16) + 2^{360} \approx 2^{360}$ 10-round encryptions. The memory complexity is $2^{368} \times 49 \times 8/128 \approx 2^{369.61}$ block sizes.

Low-complexity 10-round attack. Similar to the 10-round low-data attack, we exploit the 5-round IP transition distinguisher with one active input byte to mount a 10-round attack. Accordingly, there are at most 2^{328} possible d -differences $\beta_{*,0}^6$. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, 10-th, 15-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate

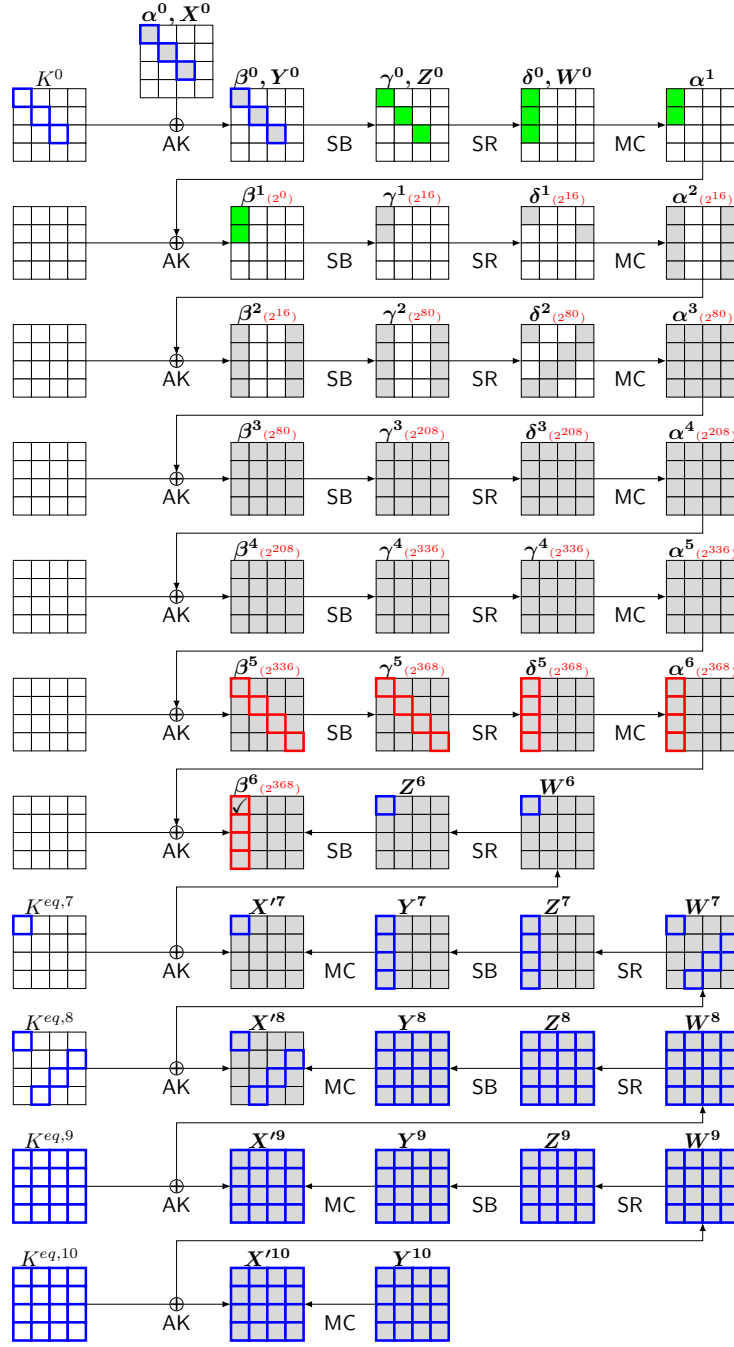


Fig. 11: The 10-round IP attack of Deoxys-BC-384

key $K = (K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus 2S(P_5^0 \oplus K_5^0) \oplus 3S(P_{10}^0 \oplus K_{10}^0) \oplus S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_1^1 \oplus K_0^0) \oplus 2S(P_5^1 \oplus K_5^0) \oplus 3S(P_{10}^1 \oplus K_{10}^0) \oplus 1S(P_{15}^1 \oplus K_{15}^0) = 0, \\ S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus 3S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_1^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) \oplus 3S(P_{15}^1 \oplus K_{15}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus 2S(P_{15}^0 \oplus K_{15}^0) \oplus \\ 3S(P_1^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) \oplus 2S(P_{15}^1 \oplus K_{15}^0) = 0 \end{cases} \right\}.$$

For each candidate key $(K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th byte.

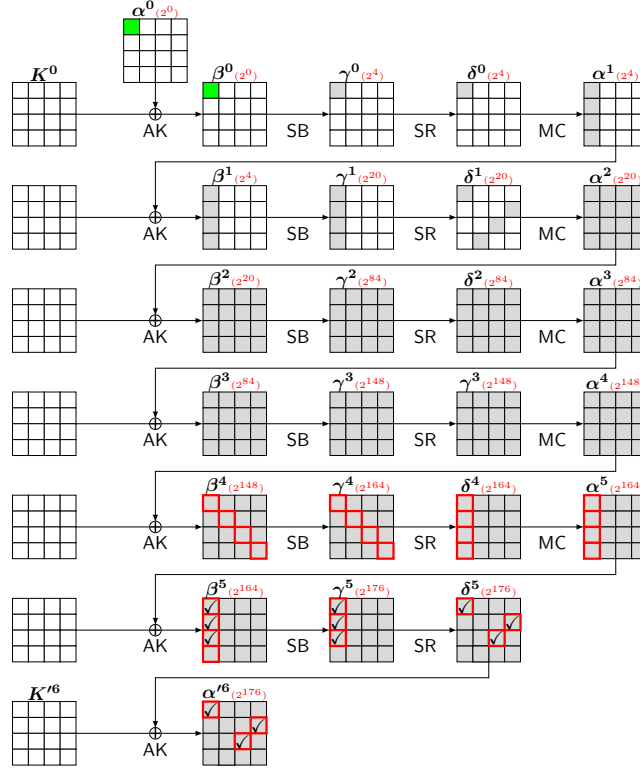


Fig. 12: The 5.5-round IP transition for Joltik-BC-192

2) Guess K_0^0 , K_5^0 , K_{10}^0 , and K_{15}^0 , then follow the same subsequent steps as the 10-round low-data attack.

Choosing $d = 49$, the probability that a d -difference $\beta_{*,0}^6$ belongs to the set Γ equals $2^{328-49 \cdot 8} = 2^{-64}$. Since we guess 328 bits of key, $2^{328-64} = 2^{264}$ key candidates remain. By performing an exhaustive search over the remaining $2^{264} + (384 - 328) = 2^{320}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{32} . The time complexity is $50 \times 2^{328} \times (4 + 1 + 4 + 16 + 16) / (10 \times 16) + 2^{328} \times 328 / (10 \times 16) + 2^{320} \approx 2^{331.89}$ 10-round encryptions. The memory complexity is $2^{328} \times 49 \times 8 / 128 \approx 2^{329.61}$ block sizes.

B. Joltik-BC-192

Joltik-BC is a lightweight tweakable block cipher proposed as the primitive of the Joltik authenticated encryption scheme [14]. Following the TWEAKEKEY framework, it adopts an AES-like round structure, which is similar to Deoxys-BC. Designed for lightweight hardware scenarios, Joltik-BC maintains a 64-bit internal state and supports two standard configurations with distinct tweak sizes, i.e., Joltik-BC-128 and Joltik-BC-192, which consist of 24 and 32 encryption rounds, respectively. The cipher utilizes a 4-bit S-box and an involutory MDS matrix for state updating. Moreover, the subtweakkey update rule in the tweakkey schedule of Joltik-BC is slightly different from that of Deoxys-BC. For the full definition and detailed parameters of Joltik-BC, we refer to the original submission [14].

Low-complexity 5.5-round distinguishers. Unlike Deoxys-BC, Joltik-BC utilizes a 4-bit S-box. With the input d -difference activating only one S-box, d is bounded by 15. For constructing long-round distinguishers, the set of all possible d -differences is large. Hence, the output target nibble cannot be fixed to a single nibble; otherwise, a valid distinguisher cannot be obtained. We thus should carefully choose the positions and the number of the active input nibbles and output target nibbles. This is the difference from constructing distinguishers for Deoxys-BC. We select the target cells corresponding to the three cells located on the anti-diagonal. In total, $16 \times 16 = 256$ distinguishers are acquired. An example distinguisher is provided below.

Distinguisher 3. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^6 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha_{0,0}^6 \| \alpha_{0,10}^6 \| \alpha_{0,13}^6 \| \dots \| \alpha_{d-1,0}^6 \| \alpha_{d-1,10}^6 \| \alpha_{d-1,13}^6)$. As shown in Figure 12, it holds that $|\Gamma| \leq 2^{176}$. By choosing $d = 15$, we can distinguish the 6-round Joltik-BC-192 without the final MC operation from a random permutation with a success probability of $1 - 2^{176-15 \cdot 12} = 1 - 2^{-4}$. The memory complexity for constructing the distinguisher is $15 \times 12 \times 2^{176} / (4 \times 16) \approx 2^{177.50}$ block sizes, and the time complexity is $2^{164} \times 2^{12} \times 16 / (6 \times 16) + 176 \times 2^{176} \times 2 / (6 \times 16) \approx 2^{177.94}$ 6-round encryptions.

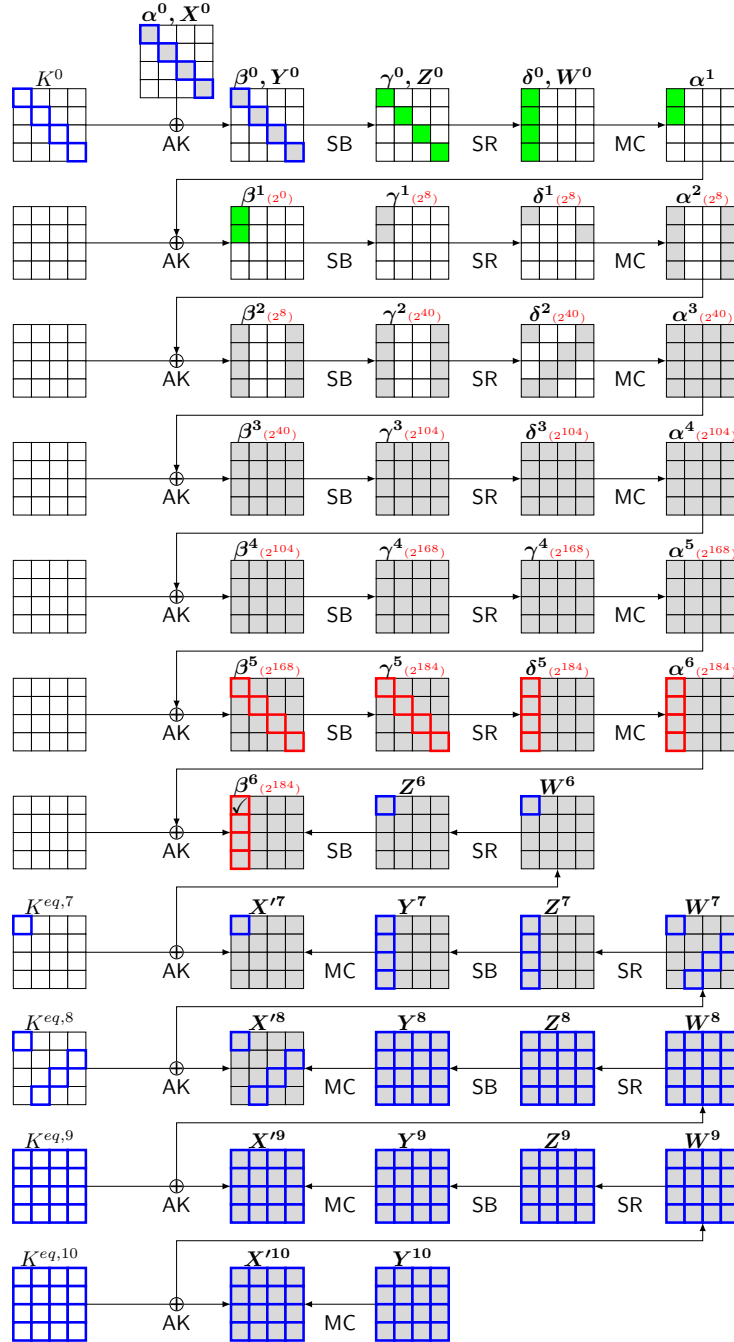


Fig. 13: The 10-round IP attack of Joltik-BC-192

Low-data 5.5-round distinguishers. When the target cells correspond to the four cells lying on the anti-diagonal, we obtain $16 \times 4 = 64$ low-data distinguishers. An example distinguisher is provided below.

Distinguisher 4. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α'^6 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha_{0,0}'^6 \| \alpha_{0,7}'^6 \| \alpha_{0,10}'^6 \| \alpha_{0,13}'^6, \dots, \alpha_{d-1,0}'^6 \| \alpha_{d-1,7}'^6 \| \alpha_{d-1,10}'^6 \| \alpha_{d-1,13}'^6).$$

Then, as shown in Figure 12, all possible values of β^5 amount to 2^{164} . Thus, $|\Gamma| \leq 2^{180}$. By choosing $d = 12$, we can distinguish 6-round Joltik-BC-192 from a random permutation with a success probability of $1 - 2^{180-12 \cdot 16} = 1 - 2^{-12}$. The memory complexity required to construct the distinguisher is $12 \times 16 \times 2^{180} / (4 \times 16) \approx 2^{181.59}$ block sizes, and the time complexity consists of $2^{164} \times 2^{16} \times 13 / (6 \times 16) + 180 \times 2^{180} \times 2 / (6 \times 16) \approx 2^{181.96}$ 6-round encryptions.

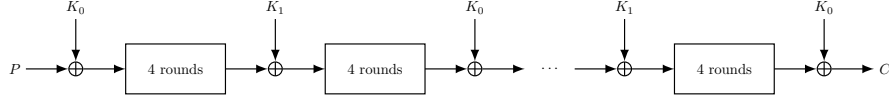


Fig. 14: Overview of LED-128

The 10-round attack. As illustrated in Figure 13, we exploit a 5-round IP transition distinguisher to construct a 10-round IP attack distinguisher by adding one round before the distinguisher and 4 rounds afterwards. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, 10-th, 15-th cells take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus 3S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) \oplus 3S(P_{15}^1 \oplus K_{15}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus 2S(P_{15}^0 \oplus K_{15}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) \oplus 2S(P_{15}^1 \oplus K_{15}^0) = 0 \end{cases} \right\}.$$

For each candidate key $(K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one round of encryption activates only the 0-th and 1-st cells.

- 2) Guess K_0^0, K_5^0, K_{10}^0 , and K_{15}^0 , then
 - a) Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d + 1)$ plaintexts yield d -differences that activate only the 0-th and 1-st bytes after one-round encryption, as illustrated by α^1 in Figure 13.
 - b) Obtain the corresponding $(d + 1)$ ciphertexts $(Y_0^{10}, \dots, Y_d^{10})$. Then we guess $K_0^{eq,7}, K_0^{eq,8}, K_7^{eq,8}, K_{10}^{eq,8}, K_{13}^{eq,8}$, as well as all bytes of $K^{eq,9}$ and $K^{eq,10}$. Afterwards, we check whether the d -differences $\beta_{*,0}^6$ belong to the set Γ ; any key candidate satisfying this condition is discarded.
- 3) Perform an exhaustive search over the remaining key material.

Choosing $d = 49$, the probability that a d -difference $\beta_{*,0}^6$ belongs to the set Γ equals $2^{184-49 \cdot 4} = 2^{-12}$. Since we guess 164 bits of key, $2^{164-12} = 2^{152}$ key candidates remain. By performing an exhaustive search over the remaining $2^{152+(192-28)}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{16} . The time complexity is $50 \times 2^{164} \times (4 + 1 + 4 + 16 + 16) / (10 \times 16) + 2^{164} \times 184 / (10 \times 16) + 2^{180} \approx 2^{180}$ 10-round encryptions. The memory complexity is $2^{184} \times 49 \times 4 / 64 \approx 2^{185.61}$ block sizes.

C. LED-128

LED is an AES-like lightweight block cipher proposed by Guo et al. at CHES 2011 [15]. It targets low-resource hardware implementations. LED-128 is a variant of LED with a 64-bit block size and a 128-bit key size. One full round consists of four sequential transformations: AddConstants (AC), SubCells (SC), ShiftRows (SR), and MixColumns (MC). As shown in Figure 14, LED does not adopt a traditional key schedule. Instead, subkey addition is performed only once every four rounds. A group of four consecutive rounds is defined as a *step*. Specifically, LED-128 iterates over 12 steps, corresponding to 48 rounds in total.

Within one step of LED, no key material is involved, so this step can be regarded as a super S-box. Leveraging this property, for a prescribed d -difference, there are at most 2^{64} possible inputs that yield this d -difference. This characteristic enables us to construct the following distinguishers.

Low-complexity 7.5-round distinguishers. Similar to Joltik-BC, LED-128 utilizes a 4-bit S-box. For constructing long-round distinguishers, the set of all possible d -differences is large. Hence, the output target nibble cannot be fixed to a single nibble; otherwise, a valid distinguisher cannot be obtained. We therefore select the target cells corresponding to the two cells located on the anti-diagonal. Since no key material is involved within a single step, we adopt the *large-Sbox strategy* to construct distinguishers. In total, $16 \times 24 = 384$ distinguishers are acquired. An example distinguisher is provided below.

Distinguisher 5. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d - 1$. Let α^8 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha_{0,0}^8 \parallel \alpha_{0,13}^8, \dots, \alpha_{d-1,0}^8 \parallel \alpha_{d-1,13}^8)$. As shown in Figure 15, it holds that $|\Gamma| \leq 2^{108}$. By choosing $d = 14$, we can distinguish the 8-round LED-128 from a random permutation with a success probability of $1 - 2^{108-14 \cdot 8} = 1 - 2^{-4}$. The memory complexity for constructing the distinguisher is $14 \times 8 \times 2^{108} / (4 \times 16) \approx 2^{108.81}$ block sizes, and the time complexity is $2^{100} \times 2^8 \times 15 / (8 \times 16) + 108 \times 2^{108} \times 2 / (8 \times 16) \approx 2^{108.86}$ 8-round encryptions.

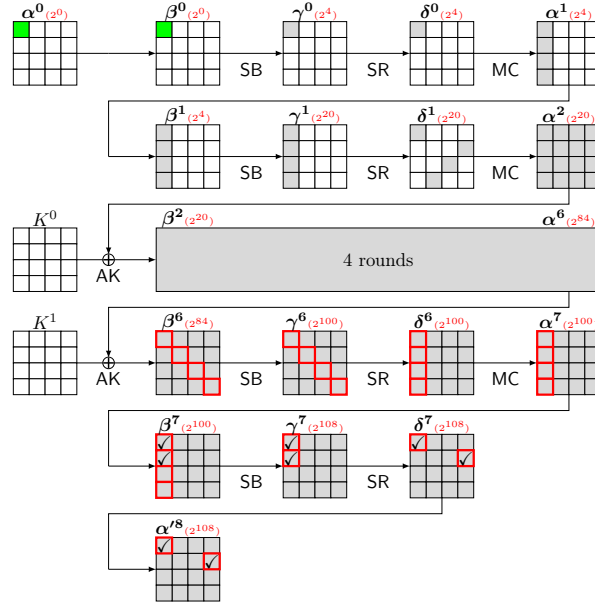


Fig. 15: The 7.5-round IP transition for LED-384

Low-data 7.5-round distinguishers. When the target cells correspond to the four nibbles lying on the anti-diagonal, by adopting the *large-Sbox strategy*, we obtain $16 \times 4 = 64$ low-data distinguishers. An example distinguisher is provided below.

Distinguisher 6. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^8 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha_{0,0}^8 \| \alpha_{0,7}^8 \| \alpha_{0,10}^8 \| \alpha_{0,13}^8, \dots, \alpha_{d-1,0}^8 \| \alpha_{d-1,7}^8 \| \alpha_{d-1,10}^8 \| \alpha_{d-1,13}^8).$$

Then, as illustrated in Figure 15, there are 2^{100} possible values of β^7 . Thus, $|\Gamma| \leq 2^{116}$. By choosing $d = 8$, we can distinguish 8-round LED-128 from a random permutation with a success probability of $1 - 2^{116-8 \cdot 16} = 1 - 2^{-12}$. The memory complexity required to construct the distinguisher is $8 \times 16 \times 2^{116} / (4 \times 16) = 2^{117}$ blocks, and the time complexity consists of approximately $2^{100} \times 2^{16} \times 9 / (8 \times 16) + 116 \times 2^{116} \times 2 / (8 \times 16) \approx 2^{116.92}$ 8-round encryptions.

D. ChiLow-32

ChiLow [3] is a family of low-latency tweakable block ciphers specifically designed for authenticated code encryption (ACE) on embedded microcontrollers. The full version of ChiLow runs with 8 rounds, using a 128-bit master key and a 64-bit tweak. ChiLow-32 is an instantiation of ChiLow. It contains a 32-bit tweakable block cipher D_{32} for decryption and a companion pseudorandom function D'_{32} for tag generation. The output of D'_{32} is truncated to τ bits, where $\tau \in \{8, 16\}$. Figure 16 outlines the complete decryption and tag computation procedure for ChiLow-32. The operations in Figure 16 are detailed as follows.

- Non-linear layer: $\mathbb{X}_n$ ($n = 32, 64, 128$) is extended from the classic χ function. Let $m = n/2$. For any input $x \in \mathbb{F}_2^n$, the output is computed as:

$$\mathbb{X}_n(x) = \left(\begin{array}{c} \chi_{m-1}(x_0, x_1, \dots, x_{m-2}) \\ \chi_{m+1}(x_{m-1}, x_m, \dots, x_{n-1}) \end{array} \right) \oplus \lambda(x).$$

Here $\lambda(x) \in \mathbb{F}_2^n$ is defined by:

$$\lambda_i(x) = \begin{cases} x_m \oplus x_{m-3}, & i = m-3, \\ x_{m-1} \oplus x_{m-2}, & i = m-2, \\ x_{m-3} \oplus x_m \oplus x_{m-1}, & i = m-1, \\ x_m \oplus x_{m-2}, & i = m, \\ 0, & \text{otherwise.} \end{cases}$$

- Linear layer: L_n ($n = 32, 64, 128$) and L'_{32} adopt the same construction. For input x and output y , each output bit satisfies: $y_i = x_{a \cdot i + b_0} \oplus x_{a \cdot i + b_1} \oplus x_{a \cdot i + b_2}$ ($0 \leq i \leq n-1$). The parameters of all linear layers are summarized in Table IV.

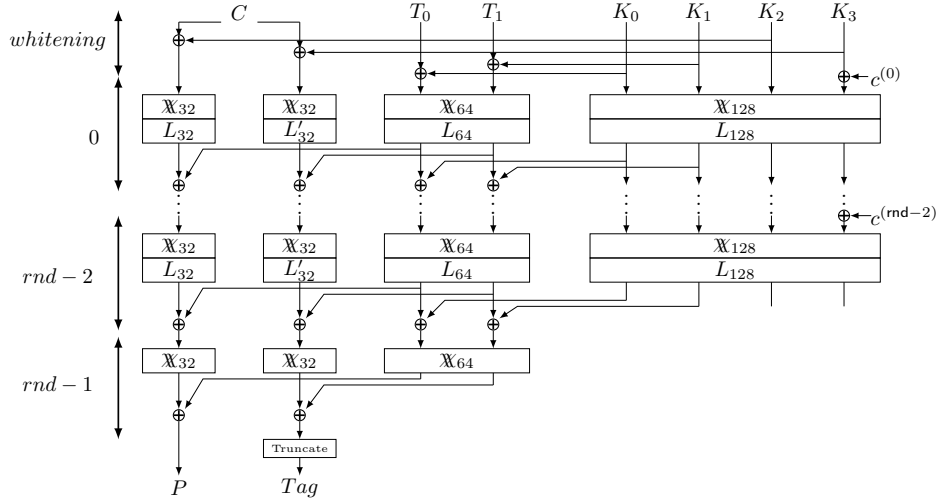


Fig. 16: Full ChiLow-32 decryption and tag computation.

TABLE IV: Parameters of linear layers in ChiLow

Layer	a	b_0	b_1	b_2
L_{32}	11	5	9	12
L'_{32}	11	1	26	30
L_{64}	3	1	26	50
L_{128}	17	7	11	14

Let the initial master key be $K = (K_0 \| K_1 \| K_2 \| K_3) \in \mathbb{F}_2^{128}$, and let the initial tweak be $T = (T_0 \| T_1) \in \mathbb{F}_2^{64}$. Let $K^{(r)} = (K_0^{(r)} \| K_1^{(r)} \| K_2^{(r)} \| K_3^{(r)}) \in \mathbb{F}_2^{128}$ and $T^{(r)} = (T_0^{(r)} \| T_1^{(r)}) \in \mathbb{F}_2^{64}$ denote the round master key and round tweak for round r , respectively. The initial tweak state is set as $T^{(0)} = T \oplus (K_0 \| K_1)$, and the initial key state is $K^{(0)} = K$. Let $rnd = 8$ be the total number of encryption rounds. For all $0 \leq i < rnd - 1$:

$$K^{(i+1)} = L_{128}(\mathbb{X}_{128}(K^{(i)} \oplus c^{(i)})),$$

$$T^{(i+1)} = L_{64}(\mathbb{X}_{64}(T^{(i)})) \oplus K_{0..63}^{(i+1)}.$$

Here $c^{(i)}$ denotes the round constant, which is only applied to bits 96 to 127 of $K^{(i)}$. For the final round, the tweak state is updated only by linear mapping: $T^{(rnd)} = L_{64}(T^{(rnd)-1})$.

To construct longer distinguishers, we employ the *subtlety-aware strategy* to analyze the detailed properties of $\mathbb{X}_{32}$.

Property 1. For the input 10-difference α with $\alpha_\ell = (\ell + 1)_{<26}$ for $0 \leq \ell \leq 9$, there exist 2^6 output d -differences; for the input 16-difference α with $\alpha_\ell = (\ell + 1)_{<26}$ for $0 \leq \ell \leq 15$, there exist 2^7 output d -differences. Here, the notation $x_{<l}$ denotes the left shift of the value x by l bits.

This property can be proven via exhaustive search by treating χ_{15} as an S-box.

Property 2. If we focus on the leftmost 13 bits of output d -differences, then for any input d -difference, there exist at most 2^{15} such output d -differences.

Proof. When χ_{15} is regarded as an S-box, any input d -difference yields at most 2^{15} distinct output d -differences. Furthermore, the linear function λ does not affect the leftmost 13 bits of the state. Consequently, for any input d -difference, the number of possible output d -differences restricted to the leftmost 13 bits is at most 2^{15} . $\square$

The 4.5-round distinguishers of D_{32} . By adopting the *subtlety-aware strategy*, a 4.5-round distinguisher is presented as follows.

Distinguisher 7. Let α^0 denote the input 10-difference, where $\alpha_\ell = (\ell + 1)_{<26}$ for $0 \leq \ell \leq 9$. Let β^4 denote the output 10-difference, and let Γ be the set of all possible output 10-differences targeting the leftmost 13 bits. As shown in Figure 17, it holds that $|\Gamma| \leq 2^{117}$. Then, we can distinguish the 5-round D_{32} from a random permutation with a success probability of $1 - 2^{117-10 \cdot 13} = 1 - 2^{-13}$. The memory complexity for constructing the distinguisher is $10 \times 13 \times 2^{117}/32 \approx 2^{119.02}$ block sizes, and the time complexity is $2^{102} \times 2^{15} \times 11/(2 \times 5) + 117 \times 2^{117} \times 2/(2 \times 5) \approx 2^{121.61}$ 5-round encryptions.



Fig. 17: The 4.5-round IP transition for ChiLow-32

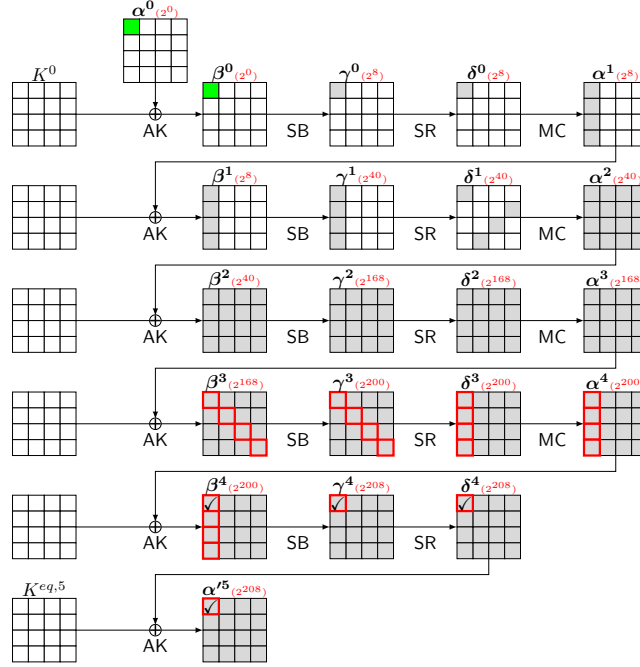


Fig. 18: The 4.5-round IP transition for Deoxys-BC-256

E. Deoxys-BC-256

Deoxys-BC-256 is a specific instance of the Deoxys-BC family with a 256-bit tweakkey.

Low-complexity 4.5-round distinguishers. Varying the positions of input active 1-bytes and output target 1-bytes gives $16 \times 16 = 256$ distinguishers in total. An example distinguisher is provided below.

Distinguisher 8. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α'^5 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha'_{0,0}, \dots, \alpha'_{d-1,0})$. As shown in Figure 18, it holds that $|\Gamma| \leq 2^{208}$. By choosing $d = 27$, we can distinguish the 5-round Deoxys-BC-256 from a random permutation with a success probability of $1 - 2^{208-27 \cdot 8} = 1 - 2^{-8}$. The memory complexity for constructing the distinguisher is $27 \times 8 \times 2^{208} / (8 \times 16) \approx 2^{208.76}$ block sizes, and the time complexity is $2^{200} \times 2^8 \times 28 / (5 \times 16) + 208 \times 2^{208} \times 2 / (5 \times 16) \approx 2^{210.47}$ 5-round encryptions.

Low-data 4.5-round distinguishers. When the target cells correspond to the four bytes lying on the anti-diagonal, we obtain $16 \times 4 = 64$ low-data distinguishers. An example distinguisher is provided below.

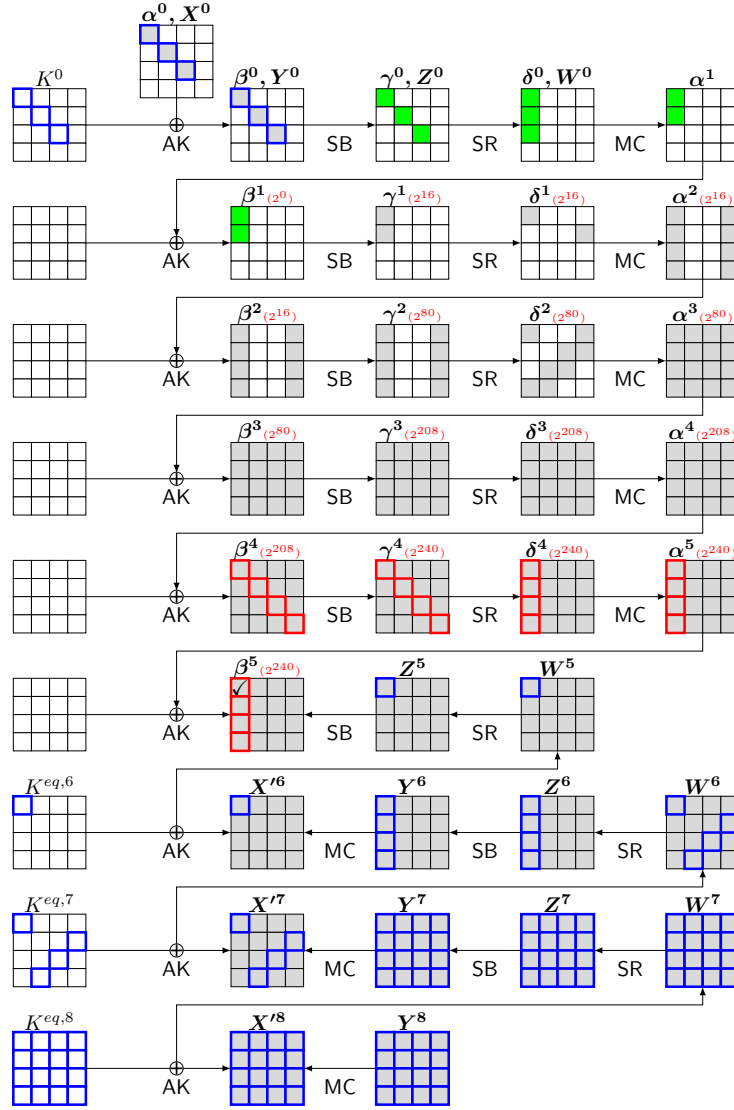


Fig. 19: The 8-round IP attack of Deoxys-BC-256

Distinguisher 9. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α'^5 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha'_{0,0}, \alpha'_{0,7}, \alpha'_{0,10}, \alpha'_{0,13}, \dots, \alpha'_{d-1,0}, \alpha'_{d-1,7}, \alpha'_{d-1,10}, \alpha'_{d-1,13}).$$

Then, as shown in Figure 18, all possible values of β^4 amount to 2^{200} . Thus, $|\Gamma| \leq 2^{232}$. By choosing $d = 8$, we can distinguish 5-round Deoxys-BC-256 from a random permutation with a success probability of $1 - 2^{232-8 \cdot 32} = 1 - 2^{-24}$. The memory complexity required to construct the distinguisher is $8 \times 32 \times 2^{232} / (8 \times 16) = 2^{233}$ block sizes, and the time complexity consists of $2^{200} \times 2^{32} \times 9 / (5 \times 16) + 232 \times 2^{232} \times 2 / (5 \times 16) \approx 2^{234.56}$ 5-round encryptions.

Low-data 8-round attack. As illustrated in Figure 19, we exploit a 4-round IP transition distinguisher to construct a 8-round IP attack distinguisher by adding one round before the distinguisher and 3 rounds afterwards. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, and 10-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate

key $K = (K_0^0, K_5^0, K_{10}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) = 0 \end{cases} \right\}.$$

For each candidate key (K_0^0, K_5^0, K_{10}^0) , there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th and 1-st bytes.

- 2) Guess K_0^0, K_5^0 and K_{10}^0 , then
 - a) Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d+1)$ plaintexts yield d -differences that activate only the 0-th and 1-st bytes after one-round encryption, as illustrated by α^1 in Figure 19.
 - b) Obtain the corresponding $(d+1)$ ciphertexts $(Y_0^8, \dots, Y_d^8)$. Then we guess $K_0^{eq,6}, K_0^{eq,7}, K_7^{eq,7}, K_{10}^{eq,7}, K_{13}^{eq,7}$, as well as all bytes of $K^{eq,8}$. Afterwards, we check whether the d -differences $\beta_{*,0}^5$ belong to the set Γ ; any key candidate satisfying this condition is discarded.
- 3) Perform an exhaustive search over the remaining key material.

Choosing $d = 33$, the probability that a d -difference $\beta_{*,0}^5$ belongs to the set Γ equals $2^{240-33 \cdot 8} = 2^{-24}$. Since we guess 192 bits of key, $2^{192-24} = 2^{168}$ key candidates remain. By performing an exhaustive search over the remaining $2^{168+(256-192)} = 2^{232}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{24} . The time complexity is $34 \times 2^{192} \times (3 + 1 + 4 + 16)/(8 \times 16) + 2^{192} \times 240/(8 \times 16) + 2^{232} \approx 2^{232}$ 8-round encryptions. The memory complexity is $2^{240} \times 33 \times 8/128 \approx 2^{241.04}$ block sizes.

Low-complexity 8-round attack. Similar to the low-data 8-round attack, we exploit the 4-round IP transition distinguisher with one active input byte to mount a 8-round attack. Accordingly, there are at most 2^{200} possible d -differences $\beta_{*,0}^5$. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, 10-th, 15-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus 2S(P_5^0 \oplus K_5^0) \oplus 3S(P_{10}^0 \oplus K_{10}^0) \oplus S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus 2S(P_5^1 \oplus K_5^0) \oplus 3S(P_{10}^1 \oplus K_{10}^0) \oplus 1S(P_{15}^1 \oplus K_{15}^0) = 0, \\ S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus 3S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) \oplus 3S(P_{15}^1 \oplus K_{15}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus 2S(P_{15}^0 \oplus K_{15}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) \oplus 2S(P_{15}^1 \oplus K_{15}^0) = 0 \end{cases} \right\}.$$

For each candidate key $(K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th byte.

- 2) Guess K_0^0, K_5^0, K_{10}^0 , and K_{15}^0 , then follow the same subsequent steps as the low-data 10-round attack.

Choosing $d = 33$, the probability that a d -difference $\beta_{*,0}^5$ belongs to the set Γ equals $2^{200-33 \cdot 8} = 2^{-64}$. Since we guess 200 bits of key, $2^{200-64} = 2^{136}$ key candidates remain. By performing an exhaustive search over the remaining $2^{136+(256-200)} = 2^{192}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{32} . The time complexity is $34 \times 2^{200} \times (4 + 1 + 4 + 16)/(8 \times 16) + 2^{200} \times 200/(8 \times 16) + 2^{192} \approx 2^{203.04}$ 8-round encryptions. The memory complexity is $2^{200} \times 33 \times 8/128 \approx 2^{201.04}$ block sizes.

F. Joltik-BC-128

Joltik-BC-128 is a specific instance of the Joltik-BC family with a 128-bit tweakkey.

Low-complexity 4.5-round distinguishers. Similar to Joltik-BC-192, when only one nibble is active in the input d -difference, the output target nibble cannot be set to one single nibble. We therefore select the target nibbles corresponding to the two nibbles located on the anti-diagonal. In total, $16 \times 24 = 384$ distinguishers are acquired. An example distinguisher is provided below.

Distinguisher 10. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^5 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha_{0,0}^5 \parallel \alpha_{0,13}^5, \dots, \alpha_{d-1,0}^5 \parallel \alpha_{d-1,13}^5)$. As shown in Figure 20, it holds that $|\Gamma| \leq 2^{108}$. By choosing $d = 14$, we can distinguish the 5-round Joltik-BC-128 from a random permutation with a success probability of $1 - 2^{108-14 \cdot 8} = 1 - 2^{-4}$. The memory

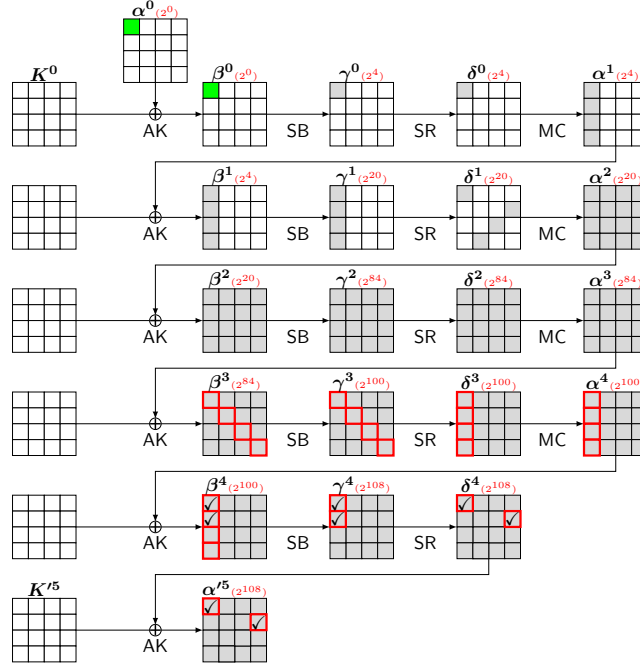


Fig. 20: The 4.5-round IP transition for Joltik-BC-128

complexity for constructing the distinguisher is $14 \times 8 \times 2^{108} / (4 \times 16) \approx 2^{108.81}$ block sizes, and the time complexity is $2^{100} \times 2^8 \times 15 / (5 \times 16) + 108 \times 2^{108} \times 2 / (5 \times 16) \approx 2^{109.53}$ 5-round encryptions.

Low-data 4.5-round distinguishers. When the target cells correspond to the four cells lying on the anti-diagonal, we obtain $16 \times 4 = 64$ low-data distinguishers. An example distinguisher is provided below.

Distinguisher 11. Let α^0 denote the input d -difference, where only the components $\alpha_{\ell,0}^0$ are non-zero for $0 \leq \ell \leq d-1$. Let α^{i5} denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha_{0,0}^{i5} \parallel \alpha_{0,7}^{i5} \parallel \alpha_{0,10}^{i5} \parallel \alpha_{0,13}^{i5}, \dots, \alpha_{d-1,0}^{i5} \parallel \alpha_{d-1,7}^{i5} \parallel \alpha_{d-1,10}^{i5} \parallel \alpha_{d-1,13}^{i5}).$$

Then, as shown in Figure 20, all possible values of β^4 amount to 2^{100} . Thus, $|\Gamma| \leq 2^{116}$. By choosing $d = 8$, we can distinguish 5-round Joltik-BC-128 from a random permutation with a success probability of $1 - 2^{116-8 \cdot 16} = 1 - 2^{-12}$. The memory complexity required to construct the distinguisher is $8 \times 16 \times 2^{116} / (4 \times 16) = 2^{117}$ block sizes, and the time complexity consists of $2^{100} \times 2^{16} \times 9 / (5 \times 16) + 116 \times 2^{116} \times 2 / (5 \times 16) \approx 2^{117.59}$ 5-round encryptions.

The 8-round attack. As illustrated in Figure 21, we exploit a 4-round IP transition distinguisher to construct a 8-round IP attack distinguisher by adding one round before the distinguisher and 3 rounds afterwards. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, 10-th, 15-th cells take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus 3S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) \oplus 3S(P_{15}^1 \oplus K_{15}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus 2S(P_{15}^0 \oplus K_{15}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) \oplus 2S(P_{15}^1 \oplus K_{15}^0) = 0 \end{cases} \right\}.$$

For each candidate key $(K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one round of encryption activates only the 0-th and 1-st cells.

- 2) Guess K_0^0, K_5^0, K_{10}^0 , and K_{15}^0 , then
 - a) Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d+1)$ plaintexts yield d -differences that activate only the 0-th and 1-st bytes after one-round encryption, as illustrated by α^1 in Figure 21.

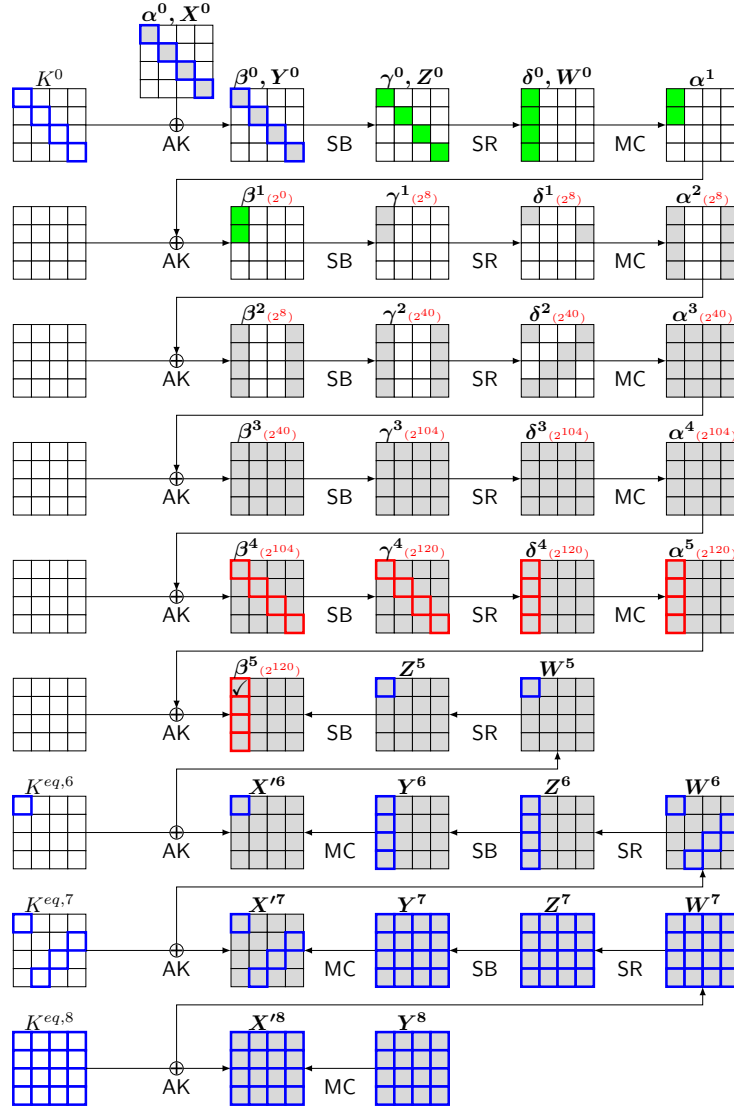


Fig. 21: The 8-round IP attack of Joltik-BC-128

- b) Obtain the corresponding $(d + 1)$ ciphertexts $(Y_0^8, \dots, Y_d^8)$. Then we guess $K_0^{eq,6}, K_0^{eq,7}, K_7^{eq,7}, K_{10}^{eq,7}, K_{13}^{eq,7}$, as well as all bytes of $K^{eq,8}$. Afterwards, we check whether the d -differences $\beta_{*,0}^5$ belong to the set Γ ; any key candidate satisfying this condition is discarded.

- 3) Perform an exhaustive search over the remaining key material.

Choosing $d = 33$, the probability that a d -difference $\beta_{*,0}^5$ belongs to the set Γ equals $2^{120-33 \cdot 4} = 2^{-12}$. Since we guess 100 bits of key, $2^{100-12} = 2^{88}$ key candidates remain. By performing an exhaustive search over the remaining $2^{88+(128-100)} = 2^{116}$ key candidates, we can recover the full secret key.

Complexity. The data complexity is 2^{16} . The time complexity is $34 \times 2^{100} \times (4 + 1 + 4 + 16) / (8 \times 16) + 2^{100} \times 120 / (8 \times 16) + 2^{116} \approx 2^{116}$ 10-round encryptions. The memory complexity is $2^{120} \times 33 \times 4 / 64 \approx 2^{121.04}$ block sizes.

G. AES-256

The 4.5-round low-complexity and low-data distinguishers for AES-256 are identical to those of Deoxys-BC-256 described in Section IV-E. However, due to the difference in key schedules, the complexities of the attacks differ.

Low-data 8-round attack. As illustrated in Figure 22, we exploit a 4-round IP transition distinguisher to construct a 8-round IP attack distinguisher by adding one round before the distinguisher and 3 rounds afterwards. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, and 10-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate

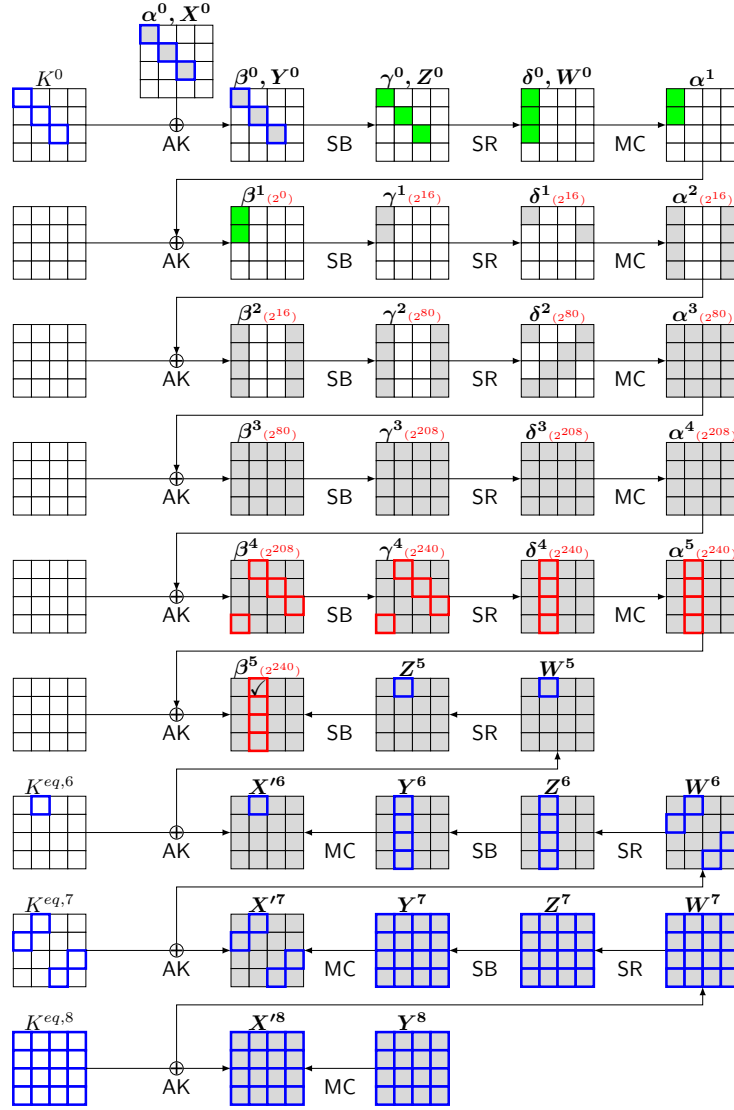


Fig. 22: The 8-round IP attack of AES-256

key $K = (K_0^0, K_5^0, K_{10}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) = 0 \end{cases} \right\}.$$

For each candidate key (K_0^0, K_5^0, K_{10}^0) , there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th and 1-st bytes.

2) Guess K_0^0, K_5^0 and K_{10}^0 , then

- Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d+1)$ plaintexts yield d -differences that activate only the 0-th and 1-st bytes after one-round encryption, as illustrated by α^1 in Figure 19.
- Obtain the corresponding $(d+1)$ ciphertexts $(Y_0^8, \dots, Y_d^8)$. Then we guess $K_4^{eq,6}, K_1^{eq,7}, K_4^{eq,7}, K_{11}^{eq,7}, K_{14}^{eq,7}$, as well as all bytes of $K^{eq,8}$. Afterwards, we check whether the d -differences $\beta_{*,4}^5$ belong to the set Γ ; any key candidate satisfying this condition is discarded.

3) Perform an exhaustive search over the remaining key material.

Choosing $d = 33$, the probability that a d -difference $\beta_{*,4}^5$ belongs to the set Γ equals $2^{240-33 \cdot 8} = 2^{-24}$. When all bytes of $K^{eq,8}$ are known, all bytes of K^8 are known. From the key schedule, $K_4^6, K_5^6, K_6^6, K_7^6$ can be deduced. Thus, $K_4^{eq,6}$

can be derived. That is, we only need to guess 184 bits of the key instead of 192 bits. Since we guess 184 bits of key, $2^{184-24} = 2^{160}$ key candidates remain. By performing an exhaustive search over the remaining $2^{160+(256-184)} = 2^{232}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{24} . The time complexity is $34 \times 2^{184} \times (3+1+4+16)/(8 \times 16) + 2^{184} \times 240/(8 \times 16) + 2^{232} \approx 2^{232}$ 8-round encryptions. The memory complexity is $2^{240} \times 33 \times 8/128 \approx 2^{241.04}$ block sizes.

Low-complexity 8-round attack. Similar to the low-data 8-round attack, we exploit the 4-round IP transition distinguisher with one active input byte to mount a 8-round attack. Accordingly, there are at most 2^{200} possible d -differences $\beta_{*,0}^5$. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th, 5-th, 10-th, 15-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \left\{ P^1 \mid \begin{cases} S(P_0^0 \oplus K_0^0) \oplus 2S(P_5^0 \oplus K_5^0) \oplus 3S(P_{10}^0 \oplus K_{10}^0) \oplus S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus 2S(P_5^1 \oplus K_5^0) \oplus 3S(P_{10}^1 \oplus K_{10}^0) \oplus 1S(P_{15}^1 \oplus K_{15}^0) = 0, \\ S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 2S(P_{10}^0 \oplus K_{10}^0) \oplus 3S(P_{15}^0 \oplus K_{15}^0) \oplus \\ S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus 2S(P_{10}^1 \oplus K_{10}^0) \oplus 3S(P_{15}^1 \oplus K_{15}^0) = 0, \\ 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus S(P_{10}^0 \oplus K_{10}^0) \oplus 2S(P_{15}^0 \oplus K_{15}^0) \oplus \\ 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) \oplus S(P_{10}^1 \oplus K_{10}^0) \oplus 2S(P_{15}^1 \oplus K_{15}^0) = 0 \end{cases} \right\}.$$

For each candidate key $(K_0^0, K_5^0, K_{10}^0, K_{15}^0)$, there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th byte.

- 2) Guess K_0^0, K_5^0, K_{10}^0 , and K_{15}^0 , then follow the same subsequent steps as the low-data 10-round attack.

Choosing $d = 33$, the probability that a d -difference $\beta_{*,4}^5$ belongs to the set Γ equals $2^{200-33 \cdot 8} = 2^{-64}$. When all bytes of $K^{eq,8}$ are known, all bytes of K^8 are known. From the key schedule, $K_4^6, K_5^6, K_6^6, K_7^6$ can be deduced. Thus, $K_4^{eq,6}$ can be derived. That is, we need to guess 192 bits of key. Since we guess 192 bits of key, $2^{192-64} = 2^{128}$ key candidates remain. By performing an exhaustive search over the remaining $2^{128+(256-192)} = 2^{192}$ key candidates, we can recover the full key.

Complexity. The data complexity is 2^{32} . The time complexity is $34 \times 2^{192} \times (4+1+4+16)/(8 \times 16) + 2^{192} \times 200/(8 \times 16) + 2^{192} \approx 2^{195.20}$ 8-round encryptions. The memory complexity is $2^{200} \times 33 \times 8/128 \approx 2^{201.04}$ block sizes.

H. AES-192

The 7-round attack. As illustrated in Figure 23, we exploit a 3-round IP transition distinguisher to construct a 8-round IP attack distinguisher by adding one round before the distinguisher and 3 rounds afterwards. The attack proceeds as follows.

- 1) Prepare an input space $\mathbb{S}$, which consists of plaintexts whose 0-th and 5-th bytes take arbitrary values, while all remaining bits are fixed. Collect the corresponding ciphertexts. Fix a plaintext P^0 . Given the candidate key $K = (K_0^0, K_5^0)$, construct the plaintext-grouping table. The K -th row, indexed by K , is defined as

$$\mathcal{P}_K = \{P^1 | 3S(P_0^0 \oplus K_0^0) \oplus S(P_5^0 \oplus K_5^0) \oplus 3S(P_0^1 \oplus K_0^0) \oplus S(P_5^1 \oplus K_5^0) = 0\}.$$

For each candidate key (K_0^0, K_5^0) , there exist $2^8 - 1$ plaintexts P^1 such that the difference between P^0 and P^1 after one-round encryption activates only the 0-th, 1-st and 2-rd bytes.

- 2) Guess K_0^0 and K_5^0 , then
 - a) Let $X_0^0 = P^0$. By looking up the plaintext-grouping table, we select d distinct values and assign them to X_ℓ^0 for $1 \leq \ell \leq d$. Then these $(d+1)$ plaintexts yield d -differences that activate only the 0-th, 1-st and 2-rd bytes after one-round encryption, as illustrated by α^1 in Figure 23.
 - b) Obtain the corresponding $(d+1)$ ciphertexts $(Y_0^7, \dots, Y_d^7)$. Then we guess $K_4^{eq,5}, K_1^{eq,6}, K_4^{eq,6}, K_{11}^{eq,6}, K_{14}^{eq,6}$, as well as all bytes of $K^{eq,7}$. Afterwards, we check whether the d -differences $\beta_{*,12}^4$ belong to the set Γ ; any key candidate satisfying this condition is discarded.
- 3) Perform an exhaustive search over the remaining key material.

Choosing $d = 21$, the probability that a d -difference $\beta_{*,12}^4$ belongs to the set Γ equals $2^{152-21 \cdot 8} = 2^{-16}$. When all bytes of $K^{eq,7}$ are known, all bytes of K^7 are known. From the key schedule, $K_0^6, K_1^6, K_2^6, K_3^6, K_4^6, K_5^6, K_6^6, K_7^6$ and $K_{12}^5, K_{13}^5, K_{14}^5, K_{15}^5$ can be deduced. Thus, $K_{12}^{eq,5}, K_3^{eq,6}$, and $K_6^{eq,6}$ can be derived. That is, we need to guess 160 bits of key. Since we guess 160 bits of key, $2^{160-16} = 2^{144}$ key candidates remain. By performing an exhaustive search over the remaining 2^{192-16} key candidates, we can recover the full key.

Complexity. The data complexity is 2^{16} . The time complexity is $22 \times 2^{160} \times (2+1+4+16)/(7 \times 16) + 2^{160} \times 152/(7 \times 16) + 2^{176} \approx 2^{176}$ 7-round encryptions. The memory complexity is $2^{152} \times 21 \times 8/128 \approx 2^{152.39}$ block sizes.

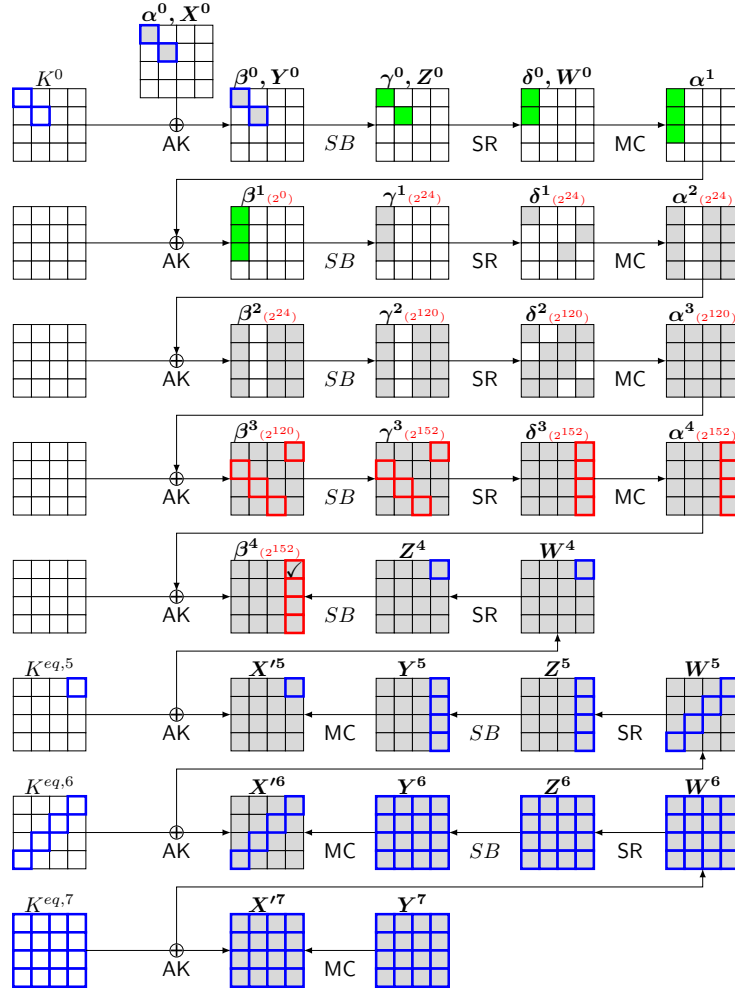


Fig. 23: The 7-round IP attack of AES-192

V. THE KP DISTINGUISHERS AND ATTACKS

In this section, we first propose our KP attack on full-round ChiLow-32. We then construct low-data KP distinguishers for Deoxys-BC-384, Deoxys-BC-256, AES-256, and AES-192.

A. ChiLow-32

In this part, we present a new full-round KP attack on the tweakable block cipher ChiLow-32. This attack only relies on the property that its nonlinear function consists of χ_{15} and χ_{17} in parallel, followed by a linear function, together with the properties of the linear layer. If χ_{15} is replaced by an arbitrary 15-bit S-box and χ_{17} by an arbitrary 17-bit S-box, the attack still holds.

The 3.5-round distinguishers. Let the leftmost 15 bits be the output target bits. A 3.5-round distinguisher is given below. Note that, since the nonlinear function of ChiLow-32 consists of χ_{15} and χ_{17} in parallel, we can equivalently represent one round of encryption as two table-lookup operations.

Distinguisher 12. Let α^0 denote the input d -difference, let β^3 denote the d -difference at round 4 without the last linear layer, and Γ be the set of all possible d -differences when targeting the leftmost 15 bits. As shown in Figure 24, $|\Gamma| \leq 2^{111}$. By choosing $d = 8$, we can distinguish the 4-round ChiLow-(32+ τ) from a random permutation with a success probability of $1 - 2^{111-8 \cdot 15} = 1 - 2^{-9}$. The memory complexity for constructing the distinguisher is $8 \times 15 \times 2^{111}/32 \approx 2^{112.91}$ block sizes, and the time complexity is $2^{96} \times 2^{15} \times 9/(2 \times 4) + 111 \times 2^{111} \times 2/(2 \times 4) \approx 2^{115.85}$ 4-round encryptions.

Low-complexity full-round attack. We utilize Distinguisher 12 to construct an full-round IP attack by appending 4.5 rounds after it. To reduce the computational complexity, we adopt the partition-guess-filter technique. We first employ a greedy algorithm to determine the ordering of target bit positions. Specifically, we iteratively select target bit positions: for the i -th bit position, given that the first $i - 1$ positions have already been fixed, we choose the position

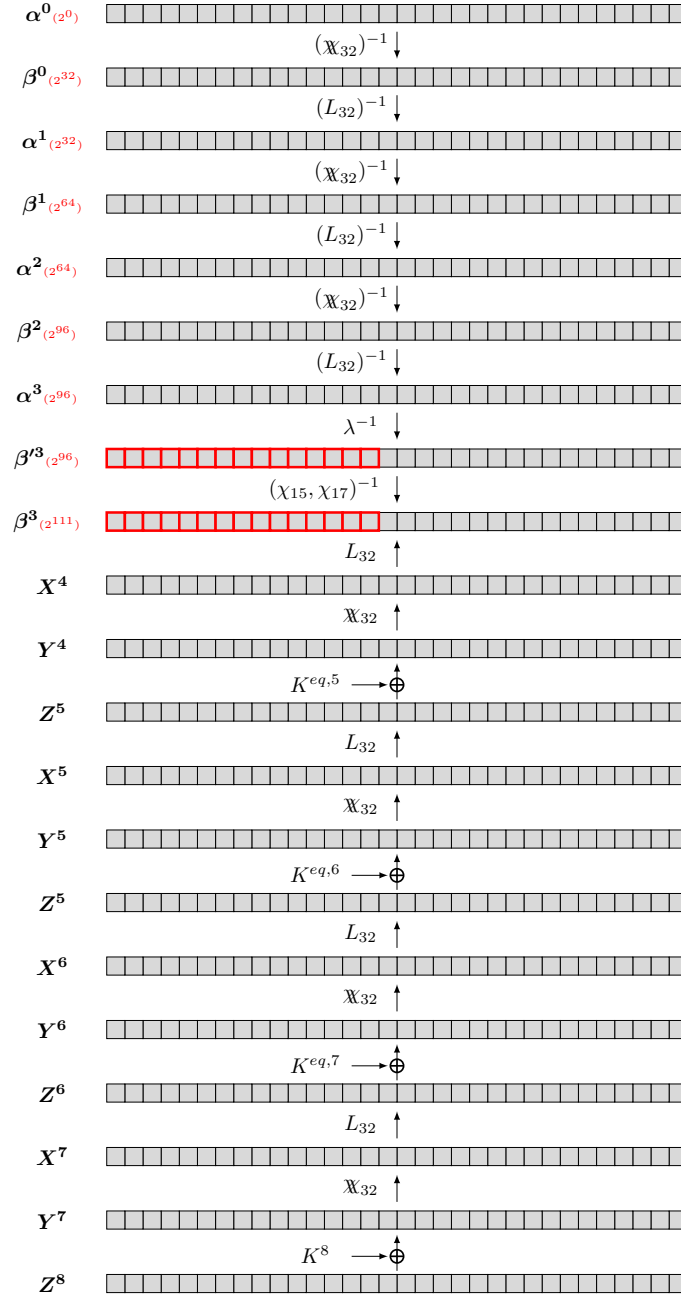


Fig. 24: Full-round KP IP attack on ChiLow-(32 + τ)

that requires the minimal number of additional guessed key bits. The results are summarized in Table V. The attack procedure is described as follows.

- 1) Choosing $d = 31$, given 32 known plaintext-ciphertext pairs, we adopt the partition-guess-filter technique to recover the key.
 - a) There are 2^{110} keys related to $\beta^3_{*,0}$, $\beta^3_{*,3}$, $\beta^3_{*,9}$, and $\beta^3_{*,12}$. We guess these keys and verify the corresponding d -differences. After filtering, $2^{110} \times 2^{111-31 \times 4} = 2^{97}$ candidate keys remain.
 - b) Guess an additional 8 bits of key material. There are 2^{97+8} keys related to $\beta^3_{*,0}$, $\beta^3_{*,3}$, $\beta^3_{*,9}$, $\beta^3_{*,12}$, $\beta^3_{*,6}$, $\beta^3_{*,8}$, $\beta^3_{*,11}$, and $\beta^3_{*,14}$. We guess these keys and verify the corresponding d -differences. After filtering, since $2^{105} \times 2^{-32 \times 4} < 1$, only one candidate key remains.
 - c) Guess an additional 10 bits of key material. There are 2^{10} keys related to the output d -difference of the distinguisher. After filtering, only one candidate key remains.
- 2) Since we have recovered the round keys of the last four rounds, we obtain 32 bits of the master key from the key schedule, and the full master key can be recovered via exhaustive search.

TABLE V: The guessed key bits for target bit positions

Bit Position	Additional Guessed Key Bits	Total Guessed Key Bits
9	104	104
12	3	107
3	2	109
0	1	110
6	2	112
14	3	115
11	2	117
8	1	118
5	1	119
2	1	120
1	6	126
13	1	127
4,7,10	1	128

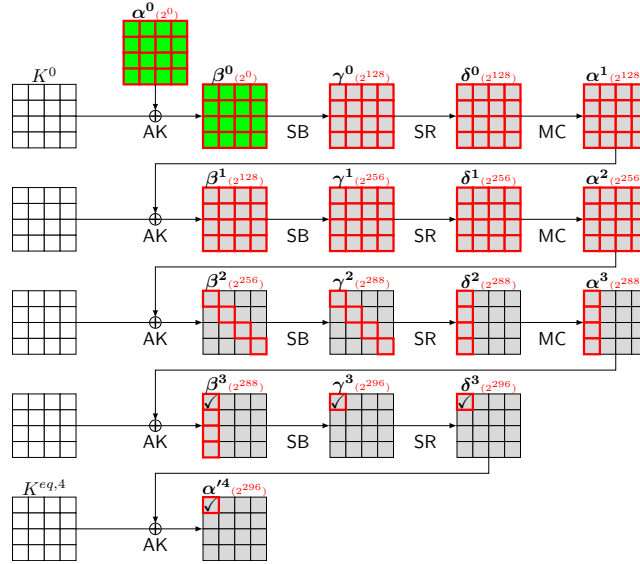


Fig. 25: The 3.5-round KP IP transition for Deoxys-BC-384

Complexity. The data complexity is 32. The time complexity is $32 \times 2^{110} \times 4/8 + 2^{110} \times 111/8 + 2^{96} \approx 2^{114.90}$ full-round encryption. The memory complexity is $31 \times 15 \times 2^{111}/32 \approx 2^{114.87}$ block sizes.

Low-data full-round attack. The attack procedure is described as follows.

- 1) Choosing $d = 12$, given 13 known plaintext-ciphertext pairs, we adopt the partition-guess-filter technique to recover the key.
 - a) As shown in Table V, there are 2^{120} keys related to $\beta_{*,0}^3, \beta_{*,2}^3, \beta_{*,3}^3, \beta_{*,5}^3, \beta_{*,6}^3, \beta_{*,8}^3, \beta_{*,9}^3, \beta_{*,11}^3, \beta_{*,14}^3$, and $\beta_{*,15}^3$. We guess these keys and verify the corresponding d -differences. After filtering, $2^{120} \times 2^{111-12 \times 10} = 2^{111}$ candidate keys remain.
 - b) Guess an additional 8 bits of key material. After filtering, since $2^{111+8} \times 2^{-12 \times 10} < 1$, only one candidate key remains.
- 2) Since we have recovered the round keys of the last four rounds, according to the key schedule, we obtain 32 bits of the master key, and the full master key can be retrieved via exhaustive search.

Complexity. The data complexity is 13. The time complexity is $13 \times 2^{120} \times 4/8 + 2^{120} \times 111/8 + 2^{96} \approx 2^{124.35}$ full-round encryption. The memory complexity is $12 \times 15 \times 2^{111}/32 \approx 2^{113.49}$ block sizes.

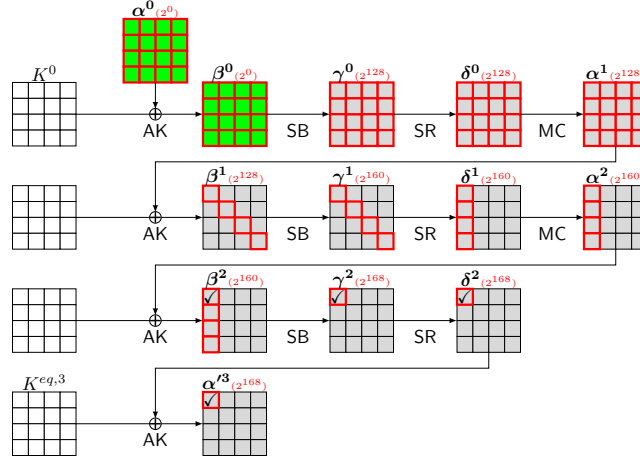


Fig. 26: The 2.5-round low-complexity KP IP transition for AES-256 and Deoxys-BC-256

B. Deoxys-BC-384

Low-complexity 3.5-round distinguishers. Varying the positions of output target 1-bytes gives 16 distinguishers in total. An example distinguisher is provided below.

Distinguisher 13. Let α^0 denote the input d -difference correspond to the $(d+1)$ plaintexts. Let α'^4 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha'_{0,0}, \dots, \alpha'_{d-1,0})$. As shown in Figure 25, it holds that $|\Gamma| \leq 2^{296}$. By choosing $d = 38$, we can distinguish the 4-round Deoxys-BC-384 from a random permutation with a success probability of $1 - 2^{296-38 \cdot 8} = 1 - 2^{-8}$. The memory complexity for constructing the distinguisher is $38 \times 8 \times 2^{296} / (8 \times 16) \approx 2^{297.25}$ block sizes, and the time complexity is $2^{288} \times 2^8 \times 39 / (4 \times 16) + 296 \times 2^{296} \times 2 / (4 \times 16) \approx 2^{293.30}$ 4-round encryptions.

Low-data 3.5-round distinguishers. When the target cells correspond to six bytes located on anti-diagonals, with three bytes on two distinct anti-diagonals respectively, we obtain 96 low-data distinguishers. An example distinguisher is provided below.

Distinguisher 14. Let α^0 denote the input d -difference correspond to the $(d+1)$ plaintexts. Let α'^3 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha'_{0,0} \parallel \alpha'_{0,1} \parallel \alpha'_{0,4} \parallel \alpha'_{0,7} \parallel \alpha'_{0,13} \parallel \alpha'_{0,14}, \dots, \alpha'_{d-1,0} \parallel \alpha'_{d-1,1} \parallel \alpha'_{d-1,4} \parallel \alpha'_{d-1,7} \parallel \alpha'_{d-1,13} \parallel \alpha'_{d-1,14}).$$

Then, similar to Figure 26, we can infer that $|\Gamma| \leq 2^{368}$. By choosing $d = 8$, i.e., any 9 plaintexts, we can distinguish 4-round Deoxys-BC-384 from a random permutation with a success probability of $1 - 2^{368-8 \cdot 48} = 1 - 2^{-16}$. The memory complexity required to construct the distinguisher is $8 \times 48 \times 2^{368} / (8 \times 16) \approx 2^{369.58}$ block sizes, and the time complexity consists of $2^{320} \times 2^{48} \times 9 \times 6 / (4 \times 16) + 368 \times 2^{368} \times 2 / (4 \times 16) = 2^{371.63}$ 4-round encryptions.

C. AES-256 and Deoxys-BC-256

Low-complexity 2.5-round distinguishers. Varying the positions of output target 1-bytes gives 16 distinguishers in total. An example distinguisher is provided below.

Distinguisher 15. Let α^0 denote the input d -difference correspond to the $(d+1)$ plaintexts. Let α'^3 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples $(\alpha'_{0,0}, \dots, \alpha'_{d-1,0})$. As shown in Figure 26, it holds that $|\Gamma| \leq 2^{168}$. By choosing $d = 22$, we can distinguish 3-round AES-192 from a random permutation with a success probability of $1 - 2^{168-22 \cdot 8} = 1 - 2^{-8}$. The memory complexity for constructing the distinguisher is $22 \times 8 \times 2^{168} / (8 \times 16) \approx 2^{168.46}$ block sizes, and the time complexity is $2^{160} \times 2^8 \times 23 / (3 \times 16) + 168 \times 2^{168} \times 2 / (3 \times 16) \approx 2^{170.90}$ 3-round encryptions.

Low-data 2.5-round distinguishers. When the target cells correspond to six bytes located on anti-diagonals, with three bytes on two distinct anti-diagonals respectively, we obtain 96 low-data distinguishers. An example distinguisher is provided below.

Distinguisher 16. Let α^0 denote the input d -difference correspond to the $(d+1)$ plaintexts. Let α'^3 denote the output d -difference excluding the final MC operation, and let Γ be the set of all possible tuples

$$(\alpha'_{0,0} \parallel \alpha'_{0,1} \parallel \alpha'_{0,4} \parallel \alpha'_{0,7} \parallel \alpha'_{0,13} \parallel \alpha'_{0,14}, \dots, \alpha'_{d-1,0} \parallel \alpha'_{d-1,1} \parallel \alpha'_{d-1,4} \parallel \alpha'_{d-1,7} \parallel \alpha'_{d-1,13} \parallel \alpha'_{d-1,14}).$$

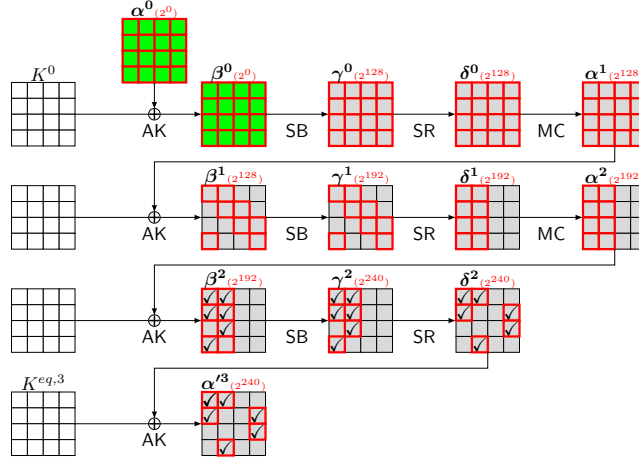


Fig. 27: The 2.5-round low-data KP IP transition for AES-256 and Deoxys-BC-256

Then, as shown in Figure 27, $|\Gamma| \leq 2^{240}$. By choosing $d = 6$, i.e., any 7 plaintexts, we can distinguish 3-round AES-256 and Deoxys-BC-256 from a random permutation with a success probability of $1 - 2^{240-6 \cdot 48} = 1 - 2^{-48}$. The memory complexity required to construct the distinguisher is $6 \times 48 \times 2^{240} / (8 \times 16) \approx 2^{241.17}$ block sizes, and the time complexity consists of $2^{192} \times 2^{48} \times 7 \times 6 / (3 \times 16) + 240 \times 2^{240} \times 2 / (3 \times 16) \approx 2^{243.44}$ 3-round encryptions.

D. AES-192

The low-complexity 2.5-round distinguishers are identical to those of AES-256 and Deoxys-BC-256 in Section V-C. **Low-data 2.5-round distinguishers.** When the target cells correspond to the three bytes lying on the anti-diagonal, we obtain 12 low-data distinguishers. An example distinguisher is provided below.

Distinguisher 17. Let α^0 denote the input d -difference correspond to the $(d+1)$ plaintexts. Let α'^3 denote the output d -difference excluding the final MC operation, and let Γ' be the set of all possible tuples

$$(\alpha'_{0,0} \parallel \alpha'_{0,7} \parallel \alpha'_{0,10}, \dots, \alpha'_{d-1,0} \parallel \alpha'_{d-1,7} \parallel \alpha'_{d-1,10}).$$

Then, as shown in Figure 26, all possible values of β^2 amount to 2^{160} . Thus, $|\Gamma'| \leq 2^{184}$. By choosing $d = 8$, i.e., any 9 plaintexts, we can distinguish 3-round AES-192 from a random permutation with a success probability of $1 - 2^{184-8 \cdot 24} = 1 - 2^{-8}$. The memory complexity required to construct the distinguisher is $8 \times 24 \times 2^{184} / (8 \times 16) \approx 2^{184.59}$ block sizes, and the time complexity consists of $2^{160} \times 2^{24} \times 9 \times 3 / (3 \times 16) + 184 \times 2^{184} \times 2 / (3 \times 16) = 2^{187.04}$ 3-round encryptions.

key sizes larger than their block sizes. These designs offer security higher than the block size.

VI. CONCLUSION

In this paper, we conduct an in-depth study on IP attack for block ciphers with key sizes larger than their block sizes, which thus provide security exceeding the block size. We formally define bound-estimation-based IP transition and propose a general framework for distinguisher construction, along with two strategies to reduce the complexity of IP transition construction. We further develop two dedicated key-recovery techniques: the plaintext-grouping technique and the partition-guess-filter technique. The former enables cryptanalysis over more rounds of the target ciphers, whereas the latter substantially reduces the overall attack complexity. These techniques enable the construction of the first framework for IP attack.

As applications, new distinguishers and attacks are derived for AES, Deoxys-BC, Joltik-BC, LED-128, and ChiLow-32. In particular, the proposed 10-round attacks on Deoxys-BC-384 and Joltik-BC-192 improve upon prior impossible differential attacks by one additional round. Furthermore, a novel full-round attack targeting ChiLow-32 under the KP setting is presented, achieving reductions in both data and overall complexity relative to state-of-the-art works. These results demonstrate that IP attack can construct longer-round distinguishers, enabling attacks over more rounds and applicability even under the KP setting.

REFERENCES

- [1] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. I, II-B
- [2] Jérémy Jean, Ivica Nikolić, Thomas Peyrin, and Yannick Seurin. Deoxys v1.41. CAESAR Competition submission, Round 3, 10 2016. I, IV-A

- [3] Yanis Belkheyar, Patrick Derbez, Shibam Ghosh, Gregor Leander, Silvia Mella, Léo Perrin, Shahram Rasoolzadeh, Lukas Stennes, Siwei Sun, Gilles Van Assche, and Damian Vizár. Chilow and chichi: New constructions for code encryption. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part I*, Lecture Notes in Computer Science, pages 212–243. Springer, 2025. I, I, III, IV-D
- [4] Tyge Tiessen. Polytopic cryptanalysis. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, Lecture Notes in Computer Science, pages 214–239. Springer, 2016. I, 1, 2, II-C, II-C, III-A, 1
- [5] National Bureau of Standards. Data encryption standard. Federal Information Processing Standards Publication (FIPS PUB 46), 1977. I
- [6] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, Lecture Notes in Computer Science, pages 2–21. Springer, 1990. I
- [7] 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
- [8] Lorenzo Grassi, Christian Rechberger, and Sondre Rønjom. A new structural-differential property of 5-round aes. In *Advances in Cryptology - EUROCRYPT 2017*, volume 10211 of *Lecture Notes in Computer Science*, pages 289–317. Springer, 2017. I
- [9] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseeth. Yoyo tricks with aes. In *Advances in Cryptology - ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 217–243. Springer, 2017. I
- [10] Navid Ghaedi Bardeh and Sondre Rønjom. The exchange attack: How to distinguish six rounds of aes with chosen plaintexts. In *Advances in Cryptology - ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 347–370. Springer, 2019. I
- [11] Xichao Hu, Yongqiang Li, Lin Jiao, Shizhu Tian, and Mingsheng Wang. Mind the propagation of states - new automatic search tool for impossible differentials and impossible polytopic transitions. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 415–445. Springer, 2020. I
- [12] Xichao Hu, Lin Jiao, Yongqiang Li, Shizhu Tian, Zhengbin Liu, Mingsheng Wang, and Dengguo Feng. Revisit the propagation of states: New construction theory and search method for impossible differentials and impossible polytopic transitions. *IEEE Trans. Inf. Theory*, 72(1):742–764, 2026. I
- [13] Ling Song, Qinggan Fu, Qianqian Yang, Yin Lv, and Lei Hu. Generalized impossible differential attacks on block ciphers: application to skinny and forkskinny. *Designs, Codes and Cryptography*, 93(7):2047–2082, 2025. I
- [14] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Joltik v1.1. Submission to the CAESAR Competition, 2015. I, IV-B, IV-B
- [15] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. LED: A light block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011. I, IV-C
- [16] Eli Biham and Nathan Keller. Cryptanalysis of reduced variants of rijndael. Official Public Comment for Round 2 of the Advanced Encryption Standard Development Effort, 2000. Available at <https://csrc.nist.gov/encryption/aes/round2/conf3/papers/35-ebiham.pdf>. I
- [17] Gaetan Leurent and Clara Pernot. New representations of the aes key schedule. In *Advances in Cryptology - EUROCRYPT 2021*, volume 12696 of *LNCS*, pages 54–84, 2021. I
- [18] Shuo Peng, Akram Khaledi, Zahra Ahmadian, Hosein Hadipour, Jiahui He, Kai Hu, Zhongfeng Niu, Shahram Rasoolzadeh, and Meiqin Wang. Cube and integral attacks on chilow-32. *IACR Trans. Symmetric Cryptol.*, 2026(1):253–292, 2026. I, III
- [19] 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. II
- [20] Achiya Bar-On, Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Improved key recovery attacks on reduced-round AES with practical data and memory complexities. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 185–212. Springer, 2018. II
- [21] Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved key recovery attacks on reduced-round AES in the single-key setting. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2013. II
- [22] Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved single-key attacks on 8-round AES-192 and AES-256. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 158–176. Springer, 2010. II
- [23] Eran Lambooi, Patrick Neumann, Michiel Verbauwheide, Shichang Wang, and Tianyu Zhang. Meet-in-the-middle attacks on full chilow. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part VI*, volume 16805 of *Lecture Notes in Computer Science*, pages 353–373. Springer, 2026. III
- [24] Hüseyin Demirci and A. Ali Selçuk. A meet-in-the-middle attack on 8-round aes. In *Fast Software Encryption*, volume 5086 of *LNCS*, pages 116–126, 2008. I
- [25] Joan Daemen. *Cipher and Hash Function Design Strategies Based on Linear and Differential Cryptanalysis*. PhD thesis, Katholieke Universiteit Leuven, 1995. I
- [26] Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, 2nd edition, 1998. II-D, II-D
- [27] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009. II-D, II-D
- [28] Jérémy Jean, Ivica Nikolić, and Thomas Peyrin. Tweaks and keys for block ciphers: the tweakey framework. In *Advances in Cryptology - ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014. IV-A

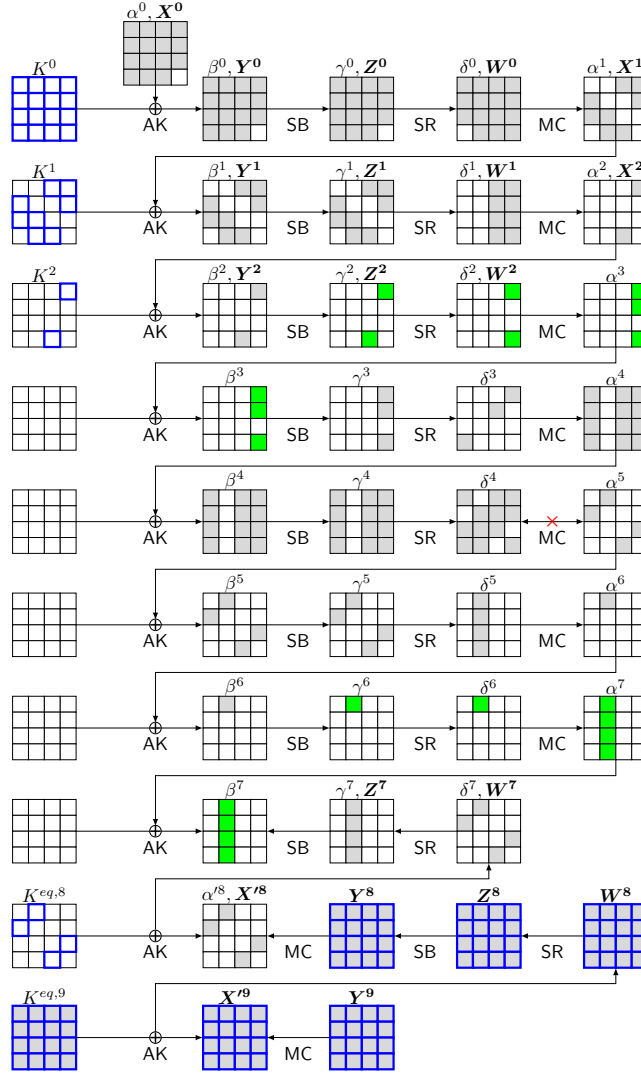


Fig. 28: The 9-round impossible differential attack of Deoxys-BC-384

APPENDIX

A. The key-recovery framework when $r_b = 0$

When $r_b = 0$, we just add r_f rounds after the distinguisher to mount the $(r_d + r_f)$ -round attack.

- 1) From the output target position,
 - a) Partition the output target positions of the distinguisher into Q groups, where each group corresponds to at least one distinct candidate key. Assume the i -th group corresponds to ι_i target-output bits ($0 \leq i \leq Q - 1$).
 - b) If $Q > 1$ and there exist ι_0 and L_d such that Equation (2) holds, let K_{out} denote the set of keys corresponding to the ι_0 bits. Otherwise, let K_{out} denote the set of keys corresponding to the output target position, and set $L_d = t/\tau + 1$. Then, we impose the constraint $d \geq L_d$.
- 2) Fix P , obtain the other d plaintexts according to the input d -differences, and obtain the corresponding ciphertexts.
- 3) For keys in K_{out} , sieve the candidate keys using the distinguisher.
- 4) After the above steps, there remain $p = 2^{t-d\tau}$ candidate keys for K_{out} . We then perform an exhaustive search over the remaining candidate keys.

Complexity. The data complexity is $d + 1$. The memory complexity is $d\tau|\Gamma|\phi_0^M$, where ϕ_0^M denotes memory-overhead fractions measured in units of one block-cipher operation. The time complexity stems from two components: Step 3 incurs time complexity $(d + 1) \cdot 2^{|K_{out}|}\phi_0^T$, and Step 4 contributes time complexity $2^{|K| - p \cdot |K_{out}|}\phi_1^T$. Herein, ϕ_0^T and ϕ_1^T denote time costs measured in units of one block-cipher encryption operation.

B. The impossible differential attacks on Deoxys-BC-384 and Joltik-BC-192

As illustrated in Figure 28, the 9-round attack of Deoxys-BC-384 is constructed by prepending 3 rounds and appending 2 rounds to the 4-round impossible differential. The detailed attack steps are given below.

- 1) Consider a plaintext structure where only the 15th byte is inactive. By preparing q such structures, we obtain $q \cdot 2^{239}$ plaintext-ciphertext pairs.
- 2) Perform partial encryption of the first round.
 - a) Guess $K_0^0, K_5^0, K_{10}^0, K_{15}^0$, verify the differences on α_0^1, α_3^1 , and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_1^0, K_6^0, K_{11}^0, K_{12}^0$, verify the differences on $\alpha_{14}^1, \alpha_{15}^1$, and discard invalid plaintext-ciphertext pairs.
 - c) For the remaining plaintext-ciphertext pairs, guess $K_2^0, K_7^0, K_8^0, K_{13}^0$, verify the differences on $\alpha_9^1, \alpha_{10}^1$, and discard invalid plaintext-ciphertext pairs.
 - d) For the remaining plaintext-ciphertext pairs, guess $K_3^0, K_4^0, K_9^0, K_{14}^0$, verify the differences on α_4^1, α_5^1 , and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $q2^{239} \times 2^{-6.8} \times 2^{16.8} \times (2 \cdot 4)/(9 \cdot 16) = \frac{1}{9}q2^{318}$ 9-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q2^{239} \times 2^{-8.8} = q2^{175}$.

- 3) Perform partial encryption of the second round.
 - a) For the remaining plaintext-ciphertext pairs, guess $K_1^1, K_6^1, K_{11}^1, K_{12}^1$, verify the differences on $\alpha_{13}^2, \alpha_{14}^2, \alpha_{15}^2$, and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_2^1, K_7^1, K_8^1, K_{13}^1$, verify the differences on $\alpha_8^1, \alpha_9^1, \alpha_{10}^1$, and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $q2^{175} \times 2^{-3.8} \times 2^{24.8} \times (2 \cdot 4)/(9 \cdot 16) = \frac{1}{9}q2^{343}$ 9-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q2^{175} \times 2^{-6.8} = q2^{127}$.

- 4) Perform partial encryption of the third round. For the remaining plaintext-ciphertext pairs, guess K_{11}^2, K_{12}^2 , verify the differences on α_{14}^3 , and discard invalid plaintext-ciphertext pairs. The time complexity of this step is $q2^{127} \times 2^{26.8} \times (2 \cdot 2)/(9 \cdot 16) = \frac{1}{9}q2^{333}$ 9-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q2^{127} \times 2^{-8} = q2^{119}$.

- 5) Perform partial decryption of the last round. Note that, since K_2^0, K_{13}^1 and K_{12}^2 are known, K_{13}^9 is known according to the tweak schedule.

- a) For the remaining plaintext-ciphertext pairs, guess $K_1^{eq,9}, K_4^{eq,9}, K_{11}^{eq,9}$ and $K_{14}^{eq,9}$, verify the differences on $\alpha_5^8, \alpha_6^8, \alpha_7^8$, and discard invalid plaintext-ciphertext pairs.
- b) For the remaining plaintext-ciphertext pairs, guess $K_2^{eq,9}, K_5^{eq,9}, K_8^{eq,9}$ and $K_{15}^{eq,9}$, verify the differences on $\alpha_8^8, \alpha_9^8, \alpha_{10}^8$, and discard invalid plaintext-ciphertext pairs.
- c) For the remaining plaintext-ciphertext pairs, guess $K_3^{eq,9}, K_6^{eq,9}, K_9^{eq,9}$ and $K_{12}^{eq,9}$, verify the differences on $\alpha_{12}^8, \alpha_{13}^8, \alpha_{15}^8$, and discard invalid plaintext-ciphertext pairs.
- d) Since

$$\begin{pmatrix} K_{12}^{eq,9} \\ K_{13}^{eq,9} \\ K_{14}^{eq,9} \\ K_{15}^{eq,9} \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} K_{12}^9 \\ K_{13}^9 \\ K_{14}^9 \\ K_{15}^9 \end{pmatrix}$$

and $K_{12}^{eq,9}, K_{14}^{eq,9}, K_{15}^{eq,9}$ and K_{13}^9 are known, $K_{13}^{eq,9}$ can be deduced, where M is the MDS matrix of AES. For the remaining plaintext-ciphertext pairs, guess $K_0^{eq,9}, K_7^{eq,9}$ and $K_{10}^{eq,9}$, verify the differences on $\alpha_0^8, \alpha_2^8, \alpha_3^8$, and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $q2^{119}2^{-9.8} \times 2^{41.8} \times (2 \cdot 4)/(9 \cdot 16) = \frac{1}{9}q2^{374}$ 9-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q2^{119} \times 2^{-12.8} = q2^{23}$.

- 6) Perform partial decryption of the second-last round. Enumerating the value of γ_4^6 , the values of $\beta_4^7, \beta_5^7, \beta_6^7, \beta_7^7$ are uniquely determined.
 - a) For the remaining plaintext-ciphertext pairs, guess $K_1^{eq,8}$, verify the differences on β_5^7 , and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_4^{eq,8}$, verify the differences on β_4^7 , and discard invalid plaintext-ciphertext pairs.
 - c) For the remaining plaintext-ciphertext pairs, guess $K_{11}^{eq,8}$, verify the differences on β_7^7 , and discard invalid plaintext-ciphertext pairs.
 - d) For the remaining plaintext-ciphertext pairs, guess $K_{14}^{eq,8}$, verify the differences on β_6^7 , and discard invalid plaintext-ciphertext pairs.

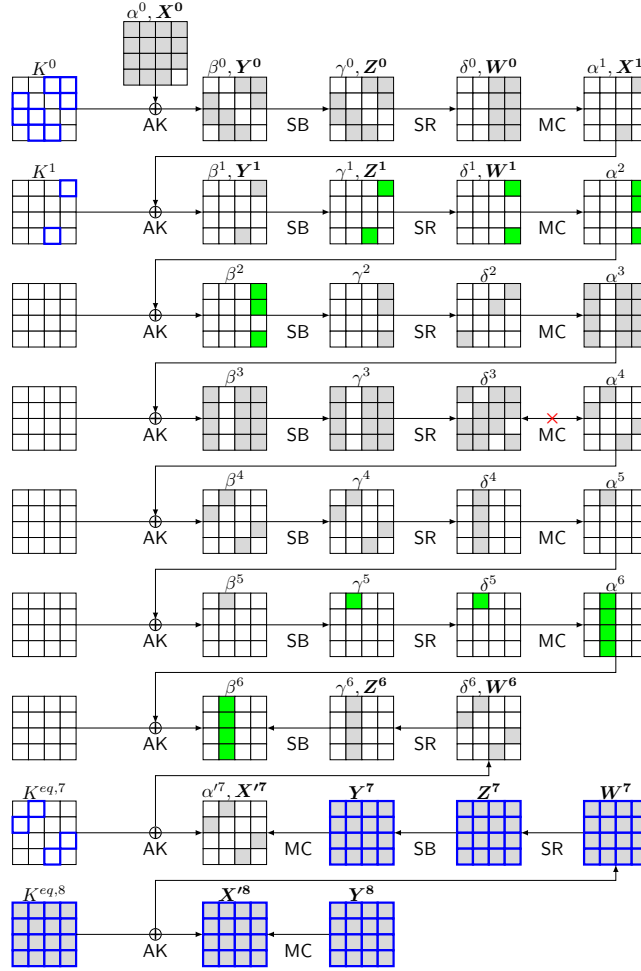


Fig. 29: The 8-round impossible differential attack of Deoxys-BC-256

The time complexity of this step is $2^8 \times q2^{23} \times 2^{-24} \times 2^{45.8} \times (2 \cdot 1)/(9 \cdot 16) = \frac{1}{9}q2^{264}$ 9-round encryptions, and the remaining number of plaintext-ciphertext pairs is $2^8 \times q2^{23} \times 2^{-4.8} = q2^{-1}$.

7) We perform an exhaustive search over all remaining key candidates to recover the correct key.

Choosing $q = 2^4$, the probability that the key survives before the exhaustive search is $p = (1 - 2^{-24})q \cdot 2^{23} = e^{-2^3} \approx 2^{-11.54}$. Thus, the complexity of exhaustive key search is less than $\frac{1}{9}q \cdot 2^{374}$. Overall, the data complexity is 2^{124} , the memory complexity is $4 \times 2^{2 \cdot 120 - 1 + 4} = 2^{245}$ block sizes, and the time complexity is $\frac{1}{9} \cdot 2^{374+4} \approx 2^{374.83}$.

The 9-round impossible differential attack on Joltik-BC-192 is very similar to that for Deoxys-BC-384. Set $q = 2^4$. The data complexity is 2^{64} , the memory complexity is $4 \times 2^{2 \cdot 60 - 1 + 4} = 2^{125}$ block sizes, and the time complexity is $\frac{1}{9} \cdot 2^{186+4} \approx 2^{186.83}$.

C. The impossible differential attacks on Deoxys-BC-256 and Joltik-BC-128

As illustrated in Figure 29, a 8-round attack of Deoxys-BC-256 is constructed by prepending 2 rounds and appending 2 rounds to the 4-round impossible differential. The detailed attack steps are given below.

- 1) Consider a plaintext structure where the 1, 2, 6, 7, 8, 11, 12, 13-th byte is active. By preparing q such structures, we obtain $q \cdot 2^{127}$ plaintext-ciphertext pairs.
- 2) Perform partial encryption of the first round.
 - a) For the remaining plaintext-ciphertext pairs, guess $K_1^0, K_6^0, K_{11}^0, K_{12}^0$, verify the differences on $\alpha_{13}^1, \alpha_{14}^1, \alpha_{15}^1$, and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_2^0, K_7^0, K_8^0, K_{13}^0$, verify the differences on $\alpha_8^1, \alpha_9^1, \alpha_{10}^1$, and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $q2^{127} \times 2^{-3.8} \times 2^{8.8} \times (2 \cdot 4)/(8 \cdot 16) = q2^{163}$ 8-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q2^{127} \times 2^{-6.8} = q2^{79}$.

- 3) Perform partial encryption of the third round. For the remaining plaintext-ciphertext pairs, guess K_{11}^1, K_{12}^1 , verify the differences on α_{14}^2 , and discard invalid plaintext-ciphertext pairs. The time complexity of this step is $q^{2^{79} \times 2^{10 \cdot 8} \times (2 \cdot 2)/(8 \cdot 16)} = q^{2^{154}}$ 8-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q^{2^{79} \times 2^{-8}} = q^{2^{71}}$.
- 4) Perform partial decryption of the last round. Note that, since K_{13}^0 and K_{12}^1 are known, K_{13}^8 is known according to the tweak schedule.
 - a) For the remaining plaintext-ciphertext pairs, guess $K_1^{eq,8}, K_4^{eq,8}, K_{11}^{eq,8}$ and $K_{14}^{eq,8}$, verify the differences on $\alpha_5'^7, \alpha_6'^7, \alpha_7'^7$, and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_2^{eq,8}, K_5^{eq,8}, K_8^{eq,8}$ and $K_{15}^{eq,8}$, verify the differences on $\alpha_8'^7, \alpha_9'^7, \alpha_{10}'^7$, and discard invalid plaintext-ciphertext pairs.
 - c) For the remaining plaintext-ciphertext pairs, guess $K_3^{eq,8}, K_6^{eq,8}, K_9^{eq,8}$ and $K_{12}^{eq,8}$, verify the differences on $\alpha_{12}'^7, \alpha_{13}'^7, \alpha_{15}'^7$, and discard invalid plaintext-ciphertext pairs.
 - d) Since

$$\begin{pmatrix} K_{12}^{eq,8} \\ K_{13}^{eq,8} \\ K_{14}^{eq,8} \\ K_{15}^{eq,8} \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} K_{12}^8 \\ K_{13}^8 \\ K_{14}^8 \\ K_{15}^8 \end{pmatrix}$$

and $K_{12}^{eq,8}, K_{14}^{eq,8}, K_{15}^{eq,8}$ and K_{13}^8 are known, $K_{13}^{eq,8}$ can be deduced, where M is the MDS matrix of AES. For the remaining plaintext-ciphertext pairs, guess $K_0^{eq,8}, K_7^{eq,8}$ and $K_{10}^{eq,8}$, verify the differences on $\alpha_0'^7, \alpha_2'^7, \alpha_3'^7$, and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $q^{2^{71} \times 2^{-9 \cdot 8} \times 2^{25 \cdot 8} \times (2 \cdot 4)/(8 \cdot 16)} = q^{2^{195}}$ 8-round encryptions, and the remaining number of plaintext-ciphertext pairs is $q^{2^{71} \times 2^{-12 \cdot 8}} = q^{2^{-25}}$.

- 5) Perform partial decryption of the second-last round. Enumerating the value of γ_4^5 , the values of $\beta_4^6, \beta_5^6, \beta_6^6, \beta_7^6$ are uniquely determined.
 - a) For the remaining plaintext-ciphertext pairs, guess $K_1^{eq,7}$, verify the differences on β_5^6 , and discard invalid plaintext-ciphertext pairs.
 - b) For the remaining plaintext-ciphertext pairs, guess $K_4^{eq,7}$, verify the differences on β_4^6 , and discard invalid plaintext-ciphertext pairs.
 - c) For the remaining plaintext-ciphertext pairs, guess $K_{11}^{eq,7}$, verify the differences on β_7^6 , and discard invalid plaintext-ciphertext pairs.
 - d) For the remaining plaintext-ciphertext pairs, guess $K_{14}^{eq,7}$, verify the differences on β_6^6 , and discard invalid plaintext-ciphertext pairs.

The time complexity of this step is $2^8 \times q^{2^{-25} \times 2^{-3 \cdot 8} \times 2^{29 \cdot 8} \times (2 \cdot 1)/(8 \cdot 16)} = q^{2^{185}}$ 8-round encryptions, and the remaining number of plaintext-ciphertext pairs is $2^8 \times q^{2^{-25} \times 2^{-4 \cdot 8}} = q^{2^{-49}}$.

- 6) We perform an exhaustive search over all remaining key candidates to recover the correct key.

Choosing $q = 2^{53}$, the probability that the key survives before the exhaustive search is $p = (1 - 2^{-24})^{q \cdot 2^{-25}} = e^{-2^4} \approx 2^{-23.08}$. Thus, the complexity of exhaustive key search is less than $q^{2^{195}}$. Overall, the data complexity is $2^{53+64} = 2^{117}$, the memory complexity is $4 \times 2^{2 \cdot 64 - 1 + 53} = 2^{182}$ block sizes, and the time complexity is $q^{2^{195}} = 2^{248}$.

The 8-round impossible differential attack on Joltik-BC-128 is very similar to that for Deoxys-BC-256. Set $q = 2^{28}$, the probability that the key survives before the exhaustive search is $p = (1 - 2^{-12})^{q \cdot 2^{-13}} = e^{-2^3} \approx 2^{-11.54}$. Thus, the complexity of exhaustive key search is less than $q^{2^{95}}$. The data complexity is 2^{60} , the memory complexity is $4 \times 2^{2 \cdot 32 - 1 + 4} = 2^{125}$ block sizes, and the time complexity is $2^{95+28} = 2^{123}$.