Guess-and-Determine Rebound: Applications to Key Collisions on AES (Full Version)

Lingyue Qin[1,2,3], Wenquan Bi[2], and Xiaoyang Dong[1,2,3(✉)]

[1] Tsinghua University, Beijing, P.R.China
{qinly,xiaoyangdong}@tsinghua.edu.cn
[2] Zhongguancun Laboratory, Beijing, P.R.China
biwq@mail.zgclab.edu.cn
[3] State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University, Beijing, P.R.China

Abstract. This paper introduces the *guess-and-determine rebound attack* that improves Dong *et al.*'s *triangulating rebound attack* in CRYPTO 2022 and Taiyama *et al.*'s key collision attack in ASIACRYPT 2024. The improvement comes from two aspects: The first improvement is to explore related-key differentials to suit for key collision attack, while Dong *et al.*'s *triangulating rebound attack* only considered single-key differentials on AES. To avoid the contradictions in the related-key differential, two tricks are proposed to identify valid trails for key collision attacks. The second improvement is to determine the range of Inbound phase flexibly with the guess-and-determine technique, to reduce the overall time complexity of the attack. By dividing the conflicts in the guess-and-determine steps into different types and handling them separately, the Inbound phase is significantly extended and ultimately leads to better or even practical key collision attacks.
Finally, we apply our method to the key collisions on AES, and improve the time complexities of all the theoretical key collision attacks on AES proposed by Taiyama *et al.* into practical ones, *i.e.*, from 2^{49} to our 2^{6} on 2-round AES-128, from 2^{61} to our 2^{21} for 5-round AES-192 and 6-round AES-256. Additionally, a new 3-round practical key collision attack on AES-128 is given, which is assumed to be impossible by Taiyama *et al.* All the practical attacks are implemented and some example pairs were found instantly on a standard PC. Besides, some quantum key collisions attacks and semi-free-start key collision attacks are proposed.

Keywords: Rebound Attack · Key Collision · Guess-and-Determine · Related-Key · Practical Attack · Quantum Attack

1 Introduction

Rebound attack [34] introduced by Mendel, Rechberger, Schläffer and Thomsen at FSE 2009, is a generic cryptanalysis tool on AES-like hash functions. The attack consists of an inbound phase and an outbound phase. In the inbound phase, the degrees of freedom are used to realize part of the differential characteristic deterministically. The remainder of the characteristic in the outbound phase is fulfilled in a probabilistic manner. To penetrate more rounds, at ASIACRYPT 2009, Lamberger *et al.* [31] proposed to connect two inbound phases by leveraging the degrees of freedom of the key. Gilbert and Peyrin [22] and Lamberger *et al.* [31] extended the inbound phase by treating two consecutive AES-like rounds as the Super-Sbox [9]. At ASIACRYPT 2010, Sasaki *et al.* [41] reduced the memory cost by exploiting the differential property of the non-full-active Super-Sbox. The memory cost of the rebound attack was further improved sequentially by Naya-Plasencia's advanced merging list algorithm [38] and Dinur *et al.*'s dissection technique [13]. At CRYPTO 2022, Dong *et al.* [14] introduced the triangulating rebound attack to penetrate more rounds in the inbound phase with the help of the triangulation algorithm [29]. The rebound attack has become a basic cryptanalysis tool to evaluate hash functions against collision attacks or distinguishing attacks [26,27,35,12,30,17,33], as well as the key collision attack on AES [43].

Quantum attacks has made significant progress in block ciphers [28,32,4,42] and hash functions [6,24,19]. At EUROCRYPT 2020, Hosoyamada and Sasaki [24] first converted the rebound attack [34] into a quantum one, and showed that, under their respective bounds of generic algorithms, quantum attacks can penetrate more rounds than classical attacks. At ASIACRYPT 2020, Dong *et al.* [15] reduced the requirement of qRAM in the quantum rebound attack by exploiting the non-full-active Super-Sbox technique [41]. At CRYPTO 2021, Hosoyamada and Sasaki [25] introduced quantum collision attacks on reduced SHA-2. At ASIACRYPT 2021, Dong *et al.* [16] studied quantum free-start collision attacks. At ToSC 2024, Chen *et al.* [7] proposed some chosen-prefix (quantum) collisions on AES-like hashing.

The Committing Attack and Key Collision. Recently, there has been a great deal of interest in the security of authenticated encryption with associated data (AEAD) in the key commitment frameworks [18,37,11,45,8]. The security in this framework ensures that a ciphertext chosen by an attacker does not decrypt into two different sets of key, nonce, and associated data. In USENIX Security 2022, Albertini *et al.* [1] revealed that the widely used AE schemes AES-GCM and ChaCha20-Poly1305 may suffer from the key committing attack. They introduced a simple countermeasure (named padding fix) by prepending a l-bit string of 0's, denoted as X, to the message M for each encryption, resulting in $\mathsf{Enc}(K, N, A, X\|M)$, and checking for the presence of X at the start of the message after decryption; decryption fails if X is not present. This countermeasure leads to the following open problem [18]:

> *"In particular, the padding fix with* AES-GCM *assumes an ideal cipher, and therefore raises the following interesting problem: Is it possible to find two keys K_1 and K_2 such that $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$ in less than 2^{64} trials. If the key size is larger than the block size, then such a pair of keys must exist. While there has been some work on the chosen-key*

setting [20] or using AES *in a hashing mode [40], we are not aware of any results on this specific problem."*

At ASIACRYPT 2024, Taiyama *et al.* [43] first answered this open question by introducing the key collision attack on AES based on the rebound attack. They found K_1 and K_2 such that $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$ for 2-round AES-128, 5-round AES-192, and 6-round AES-256 with 2^{49}, 2^{61}, and 2^{61} time complexities.

Our Contributions. In order to extend the attacked rounds by the rebound attack, Dong *et al.* introduced the triangulating rebound attack [14] and connected multiple inbound phases with the available degrees of freedom both from the key schedule and the encryption path. The core idea is to efficiently solve a nonlinear system of the byte equations of AES with the help of Khovratovich *et al.*'s triangulation algorithm [29] to fulfill the differential characteristics. However, the triangulation algorithm may fail to find good ways to solve the system when all variables appear in all or most equations simultaneously. Moreover, only single-key differentials of AES are explored in Dong *et al.*'s triangulating rebound attack [14], while the key collision attack should explore related-key differentials. As stated in [44, Section A.2], such techniques are not well-suited for solving key collision attacks:

> *"Besides, even when differential characteristics for key collision are identified, rebound attacks [34] and triangle attacks [29], which efficiently find the values which fulfill differential characteristics, are not well-suited for solving target-plaintext key collisions."*

We improve Dong *et al.*'s triangulating rebound attack [14] and Taiyama *et al.*'s key collision attack [43] with two strategies:

- First, we explore the related-key differential characteristics for our rebound attacks to adapt the key collision attacks on AES, while Dong *et al.*'s *triangulating rebound attack* only explored single-key differentials. The single-key differential characteristic allows to use all of degree of freedom of the key, while related-key differential has already fixed some key values due to fixed input/output differences of the active Sboxes in the key schedule. The consumed degrees of freedom in the key schedule may lead to contradictions with the value deduced from the encryption data path. In fact, we find the related-key differential trails on 2-round AES-128 and 6-round AES-256 used in Taiyama *et al.* [43] are invalid when searching the key collision $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$ (details are given in Section 4.1 and B.1).

  To avoid the contradictions in the related-key differentials of the key collision attacks, we introduce two tricks in the search model. The first one is to avoid activating Sboxes in round 0 of the key schedule, so that the available degrees of freedom from the key can be leveraged to connect the fixed bytes from the active Sboxes in the encryption path and the fixed plaintext P. The second trick is to assign the same difference to the active Sboxes at the same positions of the key schedule (KS) and the encryption path (EN). In this case, the probability of the two active Sboxes from the EN and KS only needs to be calculated once. This is the key factor that we can give a 3-round key collision attack on AES-128. Note that it has been proved in Taiyama *et al.*'s [44, Section B] that the 3-round key collision attack on AES-128 can hardly work:

> "*...the probability drops below* 2^{-128} *after 3 rounds. It means that in the fixed-target-plaintext scenario, no key collision pairs are guaranteed after 3 rounds for a given target plaintext, even when considering the entire 128-bit key space.*"

For our new related-key differential characteristic, if we use the same way of Taiyama *et al.* [44] to calculate its probability, it will be 2^{-131}, which is infeasible for a key collision attack. However, as the probability of two active Sboxes from the EN and KS only needs to be calculated once, the real probability is 2^{-125}, which is sufficient for a key collision attack .

– Second, we embed the guess-and-determine technique by Bouillaguet, Derbez, and Fouque [5] to solve the nonlinear system of the inbound phase to address problem that the triangulating rebound attack may not work. Moreover, we analyze the guess-and-determine (GD) steps in detail and find the conflicts (*e.g.* five conflicts marked by "?" in Table 5), which determine the complexity of the GD, could be divided into three types, *i.e.*, Type-I/II/III. Among them, Type-I conflicts could be moved to the outbound phase and Type-II conflicts could be solved with precomputation, which significantly reduces the complexity of the GD and the inbound phase.
  Compared to the key collision attacks in [44], our inbound phase covers more rounds including parts of both EN and KS, while the inbound phase in [44] only covers part of EN. For example, the inbound phase of the 6-round key collision attack on AES-256 in [44] only covers 2-round EN without KS (see Figure 14), while our inbound phase covers 4-round EN and 4-round KS (see Figure 15). Therefore, our attacks can achieve significant improvements than Taiyama *et al.*'s [44].

Based on the above two strategies, we build a heuristic method to find successful rebound attacks and key collision attacks, named the *guess-and-determine rebound attack*. The method includes two steps, the first step is to determine related-key differentials with restrictions on the degree of freedom and the tricks to avoid contradictions in the related-key differentials of the key collision attacks; the second step is to determine an efficient inbound phase via the GD and the methods to deal with the conflicts. Finally, a full rebound attack is determined.

As applications, we improve all the theoretical key collision attacks on AES proposed by Taiyama *et al.* [43] into practical ones. *We primarily focus on the key collision attack,* i.e.*, finding key pair* (K_1, K_2) *such that* $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$*, since this scenario has a practical impact on the key committing security of the widely used* AES-GCM.

– We improve the key collision attack on 2-round AES-128 from Taiyama *et al.*'s 2^{49} into the practical 2^6 time complexity.

- We first propose a new key collision attack on 3-round AES-128 with a practical time complexity of 2^{35}.
- We improve the key collision attack on 5-round AES-192 from Taiyama *et al.*'s 2^{61} into the practical 2^{21} time complexity.
- We improve the key collision attack on 6-round AES-256 from Taiyama *et al.*'s 2^{61} into the practical 2^{21} time complexity.

All our practical key collision attacks have been practically implemented and some key pairs such that $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$ are listed in Table 6 in Supplementary Material A. Futhermore, the quantum key collision attacks on 6-round AES-192 and 7-round AES-256 are given. Some improved semi-free-start collision attacks are also given for the reduced AES-DM. The results are summarized in Table 1. The verification codes for the practical attacks are given in

https://github.com/biwenquan/Guess-and-determine-Rebound

Comparison to the concurrent work by Ni *et al.* [39]. A related work recently appeared in eprint 2025/462 [39] that introduces key collision attacks on reduced AES and Kiasu-BC. In [39], the inbound phase covers 2-round or 2.5-round AES. Our inbound phase covers up to 4-round AES and up to 6-round AES's key schedule. For AES-128, we get the first 3-round practical key collision, while Ni *et al.* [39] only get a 2-round one. We get a 5-round semi-free-start collision with time complexity 2^{39}, while Ni *et al.*'s time complexity is 2^{54}; For AES-192, we get a 7-round practical semi-free-start collision with time complexity 2^{20}, while Ni *et al.*'s time complexity is 2^{56}; For AES-256, we get a 6-round practical key collision with time complexity 2^{21}, while Ni *et al.*'s time complexity is 2^{60}. The comparison is also given in Table 1.

2 Preliminaries

2.1 AES

AES [10] operates on a 4×4 column-major order array of bytes, whose round function contains four major transformations as illustrated in Figure 1: SubBytes (SB), ShiftRows (SR), MixColumns (MC), and AddRoundKey (AK). The MixColumns is to multiply each column of the state by an MDS matrix. AES has three variants called AES-128, AES-192, and AES-256 with key lengths of 128 bits, 192 bits, and 256 bits, respectively.

2.2 Key Collision Attacks

At ASIACRYPT 2024, Taiyama *et al.* introduced three variants of key collisions as Figure 2.

Definition 1 (Key Collision [43]). *It is two distinct keys that generate the same ciphertext for a single target plaintext.*

Table 1: A summary of the results

Target	Attack	Rounds	Time	C-Mem	qRAM	Setting	Ref.
AES-128	Key Collision	2/10	2^{49}	-	-	Classic	[43]
		2/10	Practical	2^{22}	-	Classic	[39]
		2/10	2^{6} Practical	-	-	Classic	Sect. 4.2
		3/10	2^{35} Practical	-	-	Classic	Sect. 4.3
	DM mode	5/10	2^{57}	-	-	Classic	[43]
	Semi-free-start	5/10	2^{54}	-	-	Classic	[39]
		5/10	2^{39}	-	-	Classic	Sect. D.1
AES-192	Key Collision	5/12	2^{61}	-	-	Classic	[43]
		5/12	Practical	2^{5}	-	Classic	[39]
		5/12	2^{21} Practical	-	-	Classic	Sect. C.1
		6/12	$2^{38.7}$		44	Quantum	Sect. C.2
	DM mode	7/12	2^{62}	-	-	Classic	[43]
	Semi-free-start	7/12	2^{56}	-	-	Classic	[39]
		7/12	2^{20} Practical	-	-	Classic	Sect. D.2
AES-256	Key Collision	6/14	2^{61}	-	-	Classic	[43]
		6/14	2^{60}	-	-	Classic	[39]
		6/14	2^{21} Practical	-	-	Classic	Sect. 5.1
		7/14	$2^{36.7}$		60	Quantum	Sect. 5.2

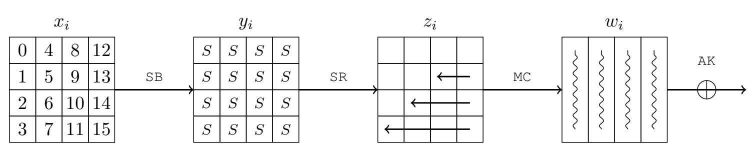

Fig. 1: The round function of AES

Identifying such a collision can be classified into two different problems depending on whether a single target plaintext is predetermined or not, illustrated in Figure 2. *Obviously, the most important and difficult case is fixed-target-plaintext key collision, i.e., finding key pair* (K_1, K_2) *such that* $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$. *This scenario has a direct impact on the key commitment security of* AES-GCM *and its padding fix variant [1]. Therefore, this paper focuses on this important case.*

The time complexity for solving these problems by generic attack (assuming that an underlying block cipher is an ideal cipher) depends on the size of the ciphertext. Specifically, for an n-bit ciphertext, such pairs can be found within a time complexity of $2^{n/2}$ in classical setting, owing to the birthday paradox. In quantum setting, the quantum version of parallel rho's algorithm [46,2,24] achieves a time-space tradeoff of time $\frac{2^{n/2}}{S}$ with S computers.

2.3 The Rebound Attack

The rebound attack was first introduced by Mendel *et al.* in [34], which consists of an inbound phase and an outbound phase as shown in Figure 3, where F is

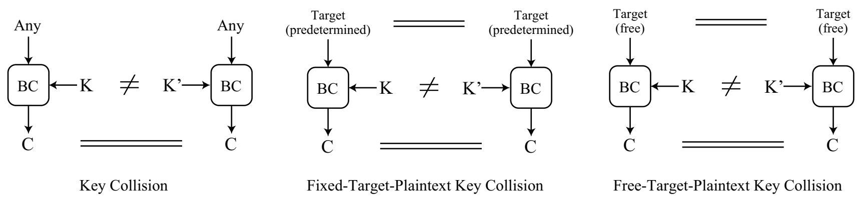

Fig. 2: Variants of key collisions

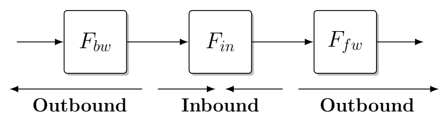

Fig. 3: The rebound attack

an internal block cipher or permutation which is split into three subparts, then $F = F_{fw} \circ F_{in} \circ F_{bw}$.

- **Inbound phase**. In the inbound phase, the attackers efficiently fulfill the low probability part in the middle of the differential trail with a meet-in-the-middle technique. The degree of freedom is the number of matched pairs in the inbound phase, which will act as the starting points for the outbound phase.
- **Outbound phase**. In the outbound phase, the matched values of the inbound phase, *i.e.*, starting points, are computed backward and forward through F_{bw} and F_{fw} to obtain a pair of values which satisfy the outbound differential trail in a brute-force fashion.

2.4 Triangulating Rebound Attack

At CRYPTO 2022, Dong *et al.* introduced the triangulating rebound attack [14]. The core idea is to connect multiple inbound phases by solving a nonlinear system of byte equations.

In Figure 4, we take the inbound phase of Dong *et al.*'s 7-round rebound attack on AES-128 as an example (see [14, Section 4.1]) to describe the triangulating rebound attack. There are two inbound phases named 'Inbound I' and 'Inbound II'. The triangulating rebound attack begins with the given differences of $(\Delta z_2, \Delta w_3, \Delta w_4, \Delta w_5)$, so that the input-output differences for the three SB layers in Round 3, 4, and 5 are determined. Based on the differential property of AES's Sbox, one can expect one pair of values for active bytes $(x_3[\blacksquare], x_4[\blacksquare], x_5[\blacksquare])$. To connect these values, 9 bytes of $k_4[\bullet]$ are directly determined by $k_4 = x_4 \oplus w_3$. The other 7 bytes of k_4 act as variables. Together with the known state w_3, we compute forward to get 6 nonlinear byte equations with the 6 known bytes $x_5[\blacksquare]$ for the 7 variables of k_4. There expect 2^8 solutions for the nonlinear system. Trivially, we may solve the system by exhaustive search and check if the 6 equations

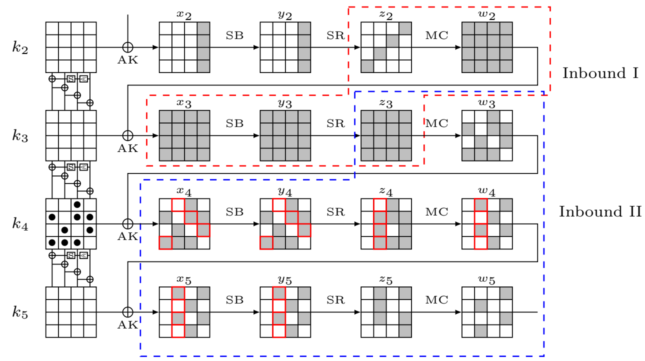

Fig. 4: Example of triangulating rebound attack in [14]

are satisfied, which needs 2^{56} time complexity to find all the solutions. Dong *et al.* figure out that the system can be solved by a triangulation algorithm efficiently in 2^8 time.

The triangulation algorithm was introduced by Khovratovich, Biryukov, and Nikolic [29] at CT-RSA 2009. The heart of the triangulation algorithm is to search for free variables. The formal process can be described as follows:

1. Given the system of equations with predefined values fixed as constants.
2. Label all variables and equations as unprocessed. Initially, all variables and equations are marked as unprocessed, meaning they have not yet been simplified or solved.
3. Identify a variable that appears in only one unprocessed equation. Label both the variable and the corresponding equation as processed. If there is no such variable — exit.
4. Repeat Step 3 if there are still unprocessed equations.
5. If all equations have been processed or no further simplification can be made, mark all remaining unprocessed variables as free.
6. Assign random values to free variables and compute the remaining variables.

Assume we have 7 byte-variables $s, t, u, v, x, y, z \in \mathbb{F}_2^8$ which are involved in the following byte-equations:

$$\begin{cases} F(x \oplus s) \oplus v = 0, \\ G(x \oplus u) \oplus s \oplus L(y \oplus z) = 0, \\ v \oplus G(u \oplus s) = 0, \\ H(z \oplus s \oplus v) \oplus t = 0, \\ u \oplus H(t \oplus x) = 0, \end{cases} \tag{1}$$

where F, G, H, and L are the bijective functions. After processing with the triangulation algorithm, we get

$$\begin{cases} L(y \oplus z) \oplus G(u \oplus x) \oplus s = 0, \\ z \oplus H^{-1}(t) \oplus v \oplus s = 0, \\ t \oplus H^{-1}(u) \oplus x = 0, \\ u \oplus G^{-1}(v) \oplus s = 0, \\ v \oplus F(x \oplus s) = 0. \end{cases} \tag{2}$$

Evidently, $x,\ s \in \mathbb{F}_2^8$ can be assigned randomly and deduce the other variables.

3 Guess-and-Determine Rebound Attack

3.1 The Weaknesses of Dong *et al.*'s Triangulating Rebound

Weakness I: Triangulation algorithm failed. The weakness of Dong *et al.*'s triangulating rebound [14] inherits from the triangulation algorithm [29]. The triangulation algorithm may fail to find good ways to solve the nonlinear system when all the variables appear in all or most equations simultaneously. For example, if the nonlinear system is the following Equation 3 ('S' is the application of Sbox), the triangulation algorithm terminates immediately without any processing, and the system will be solved by exhaustive search.

$$\begin{cases} x \oplus y \oplus S(y) \oplus z \oplus S(z) \oplus t \oplus S(t) = 0, \\ S(x) \oplus y \oplus S(y) \oplus z \oplus S(z) \oplus t \oplus S(t) = 0, \\ x \oplus S(x) \oplus 2y \oplus S(y) \oplus 3z \oplus 3S(z) \oplus 2t \oplus 3S(t) = 0. \end{cases} \tag{3}$$

However, the system can be simplified by the Gaussian elimination to be

$$\begin{cases} z \oplus S(z) \oplus S(t) \oplus S(y) = 0, \\ t \oplus S(x) \oplus y \oplus 2S(y) = 0, \\ x \oplus S(x) \oplus 2S(y) = 0. \end{cases} \tag{4}$$

The simplified system can be solved easily in 2^8 time by exhausting $y \in \mathbb{F}_2^8$. In fact, at CRYPTO 2011, Bouillaguet, Derbez and Fouque [5] have already proposed an efficient guess-and-determine method to solve the nonlinear system of related-key AES, which adopted the Gaussian elimination method to process the system. Therefore, we apply Bouillaguet *et al.*'s guess-and-determine tool [5] to solve AES's nonlinear system and improve the rebound attack.

Weakness II: Related-key differential unexplored on AES for triangulation rebound. The other weakness is that Dong *et al.*'s triangulating rebound attack [14] on AES only explores the single-key differential. Note that single-key differential allows full use of degree of freedom of the key, while related-key differential characteristic has already fixed some key values (lost some degrees of freedom from the key schedule) due to the fixed input/output differences of the active Sboxes in the key schedule. In addition, using related-key differential may induce unexpected conflicts in the attacks [3]. In fact, we find that the related-key differential characteristics on 2-round AES-128 and 6-round AES-256 used in Taiyama *et al.* [43] are invalid when searching the key collision $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$. When P is fixed, the value deduced from the active Sbox in the encryption path may conflict with the value deduced from the active Sbox in the key schedule. The details are given in Section 4.1 and B.1. Hence, considering related-key differential is not trivial, the consumed degrees of freedom in the key schedule may lead to the whole attack being invalid.

Those problems make Dong *et al.*'s triangulating rebound attack [14] not well-suited for the key-collision attack on AES, since this kind of attack is based on differences in the key schedule. This drawback has been spotted by Taiyama *et al.* [43] from ASIACRYPT 2024 that "*rebound attacks and triangle attacks are not well-suited for solving target-plaintext key collisions*" [44, Section A.2].

3.2 Guess-and-Determine Rebound Attack (GD rebound)

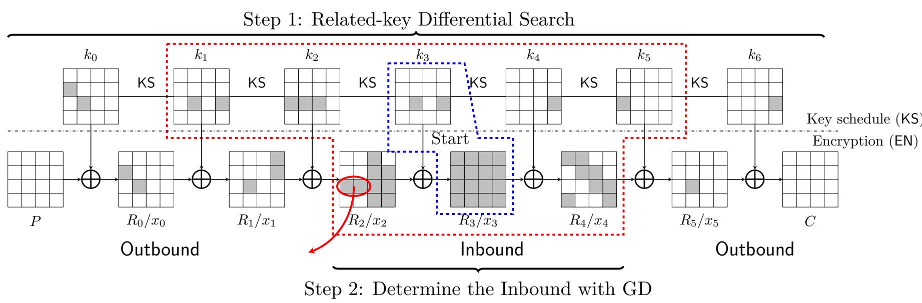

Fig. 5: Framework of guess-and-determine rebound

Our *guess-and-determine rebound attack* (abbreviated as "GD rebound") investigates the related-key differentials of AES to suit key collision attacks. Figure 5 shows the framework of our GD rebound, where the two critical steps are given as follows.

Step 1: Search for related-key differentials of AES by applying Gérault *et al.*'s model [21]. This step involves two sub-steps, *i.e.*, searching for the related-key truncated differential and searching for the instantiation of the truncated differential. The instantiation of the related-key differential characteristic (RKDC) should satisfy the following conditions:

- *Collision Condition:* There should be no active bytes in the states P and C for fixed or free-target-plaintext key collision.
- *Degree of Freedom (DoF):* Similar to Taiyama *et al.* [43], the differential characteristic should be constrained by the maximum DoF in each attack. A fixed-target-plaintext key collision can utilize the DoF of the key K, while the free-target-plaintext key collision can utilize the DoF of both the key K and plaintext P. Thus, the differential characteristic with probability 2^{-p} should meet the condition $p < |K|$ for fixed-target-plaintext key collision, or condition $p < n + |K|$ for free-target-plaintext key collision, where $|K|$ and n are the bit-length of the key and the plaintext.
- *Restriction on Differential in Round 0:* For key collision, the differences are all introduced by the key. Especially, for state x_0 in the round 0, we have $\Delta x_0 = \Delta k_0$. For the fixed-target attack, the value deduced from the active Sbox in the encryption path may conflict with the value deduced from the

active Sbox in the key schedule in the position of fixed P. We take the RKDC of 2-round AES-128 given in [43] as an example. As shown in Figure 6, there are $(\Delta x_0[12], \Delta\mathsf{SB}(x_0[12])) = (\texttt{0x69}, \texttt{0xef})$ and $(\Delta k_0[12], \Delta\mathsf{SB}(k_0[12])) = (\texttt{0x69}, \texttt{0x08})$. To fulfill the differential, the values of $x_0[12]$ and $k_0[12]$ must be $x_0[12] \in \{\texttt{0x1b}, \texttt{0x72}\}$ and $k_0[12] \in \{\texttt{0x60}, \texttt{0x08}\}$. With fixed $P[12] = 0$ and $P[12] = k_0[12] \oplus x_0[12] = 0$, the values of $x_0[12]$ and $k_0[12]$ cannot satisfy the differential. For details please refer to Section 4.1.

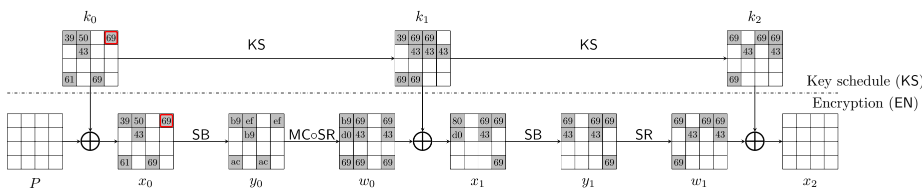

Fig. 6: The differential for 2-round AES-128 in [43]

We solve this incompatibility with two tricks:

1. The first way is to avoid activating Sbox in round 0 of the key schedule. For AES-128, the condition is satisfied by $\Delta k_0[j] = 0$ ($j \in [12, 13, 14, 15]$). One example is the new 2-round RKDC in Figure 8 of Section 4.2, that leads to a practical key collision attack on 2-round AES-128.
2. The second way is to set the output differences in the corresponding active Sbox in KS and EN path to be same. Then fix the corresponding state byte and key byte to the same value to keep $P = 0$. For example, in Figure 6, we can modify $\Delta\mathsf{SB}(x_0[12]) = \Delta\mathsf{SB}(k_0[12])$ and keep $x_0[12] = k_0[12]$.
   However, the degree of freedom and probability of the RKDC should be reconsidered. Because when $x_0[12] = k_0[12]$, once the value of $x_0[12]$ satisfies the active Sbox in the SB operation in encryption path EN, this value will instantly satisfy the corresponding active Sbox in the key schedule KS with probability 1. For the two active Sboxes of $x_0[12]$ and $k_0[12]$, the probability only needs to be calculated once[4]. At the same time, the degree of freedom should take into account the choice of $x_0[12] = k_0[12]$. This is the key factor that we can give a 3-round practical key collision attack on AES-128 in Section 4.3, which is believed impossible by Taiyama *et al.* [44, Section B].

Step 2: Determine the Inbound phase with guess-and determine. Given a related-key differential characteristic (RKDC), the key point of the GD rebound attack is to determine the Inbound phase. Figure 5 shows a 6-round RKDC,

[4] Similar features are also spotted by Nageler *et al.* when studying the joint differential characteristics [36].

where R_i represents the round i and only the state x_i before SB in round i is presented for short. Our strategy for determining the Inbound phase with guess-and-determine is as follows.

1. Select the starting round as the initial Inbound, *e.g.*, the starting round 3 in Figure 5 including the key schedule path KS and the encryption path EN. There are different choices for the starting round. Since the differences of the active Sboxes of the RKDC are fixed, we then fix all the values of the active Sboxes in KS and EN path by accessing DDT in the initial Inbound. The remaining part of the RKDC is the initial Outbound, which will be satisfied in a brute-force fashion. Suppose that the probability of the initial Outbound part is $2^{-p_{out}}$. If $2^{p_{out}} \geq 2^{n/2}$ (the rebound attack is already weaker than the birthday attack), add more rounds (or a partial round) of the KS or EN into the initial Inbound to get the new Inbound[5] marked by red dashed box in Figure 5. For fixed target-plaintext key collision, the Inbound phase usually includes the state P. Otherwise, a complexity of 2^n should be added to the Outbound phase to meet the fixed P, which already invalidates the attack. Suppose that the current probability of the Outbound part is $2^{-p_{out}}$.
2. Feed the known bytes (deduced by DDT) and unknown bytes of the Inbound into Buillaguet *et al.*'s guess-and-determine tool [5] to find an efficient GD for the Inbound. For example, Table 5 summarizes the steps of the GD for the Inbound on 7-round AES-256. However, in our cryptanalysis on AES, there exist *conflicts* during the GD, *e.g.*, five conflicts marked by "?" in Table 5. Trivially, these *conflicts* can be solved in a brute-force search. Suppose that the number of *conflicts* is c_{in}, the time complexity of the GD to find one starting point is $\mathcal{T}_{\mathsf{GD}} = 2^{8c_{in}}$.
   If there are too many *conflicts*, the overall time complexity may exceed the upper bound of a valid attack. In our cryptanalysis, we find that there are three types of *conflict*, which should be treated in different ways to speed up the full attack.
   - **Type I: Active sboxes falsely included in the Inbound.** In Figure 5, all the active Sboxes in the Inbound should be specified as known bytes by DDT. When the known bytes in the boundary of the Inbound are deduced again from GD, they will lead to conflicts. For example, two active bytes $x_2[2,6]$ included in the Inbound of Figure 5 are deduced again by GD, which will result in a 2-byte conflict acting as a filter of 2^{-16}. However, if we put the two bytes in the Outbound, they will be satisfied with probability of at least $2^{-7\times 2} = 2^{-14}$. Therefore, this type of conflicts should be solved in the Outbound phase to save time complexity.
   - **Type II: Conflict between KS and EN path.** Figure 7 shows the first 3 rounds of the Inbound in the 6-round attack on AES-256 in Section 5.1. After fixing the active bytes of $\{x_0, y_0, x_1, y_1, x_2, k_1\}$ (marked by V) by DDT, we deduce $k_2[2]$. Then with fixed $P = 0$ for key collision attack, $k_2[2]$ is again deduced through KS, *i.e.*, we get two equations about $k_2[2]$ in Equation 5.

[5] Note that in [43], only part of EN is selected as Inbound without the KS.

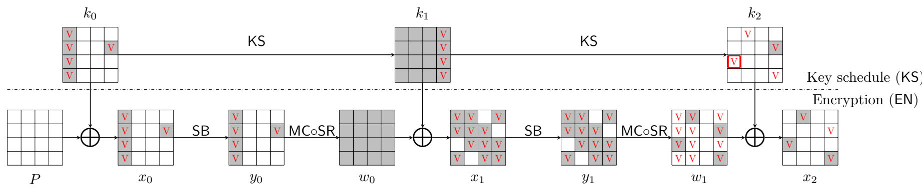

Fig. 7: Type II conflict between KS and EN path

$$\begin{cases} k_2[2] = y_1[0] \oplus y_1[5] \oplus 02 \cdot y_1[10] \oplus 03 \cdot y_1[15] \oplus x_2[2], \\ k_2[2] = x_0[2] \oplus P[2] \oplus \mathsf{SB}(k_1[15]). \end{cases} \tag{5}$$

where the bytes marked by red are known. The conflict can be solved in the brute-force fashion with time complexity 2^8. However, with a precomputation of Equation 6 as

$$y_1[0] \oplus y_1[5] \oplus 02 \cdot y_1[10] \oplus 03 \cdot y_1[15] \oplus x_2[2] \oplus x_0[2] \oplus P[2] \oplus \mathsf{SB}(k_1[15]) = 0, \tag{6}$$

on the known bytes $y_1[0,5,10,15]$, $x_2[2]$, $x_0[2]$ and $k_1[15]$, the 2^8 complexity can be saved. If any of the choices of the known bytes can not satisfy Equation 6, search a new differential characteristic. If satisfied, deduce the value of $k_2[2]$ without conflict. So this type of conflict does not affect the overall complexity.

Following the above example, we formalize the Type II conflict: Given the input/output differences of active Sboxes, one can derive a couple of input/output values by DDT of those active Sboxes. Given a differential characteristic, there are some constraints on those input/output values of the active Sboxes, like Equation 6, which are the so-called Type II conflicts. The steps to handle the Type II conflicts are:

(a) Given a differential characteristic, precompute all Type II conflicts, like Equation 6.
(b) Select the input/output values of the active Sboxes to directly satisfy all Type II conflicts.
(c) Perform the rebound attacks with these valid input/output values for those active Sboxes.
(d) If any input/output value of the active Sboxes does not satisfy the constraints, search a new differential characteristic.

– **Type III: Internal Conflict.** Conflicts that cannot be moved to the Outbound phase (conflicts are not on the boundary of the Inbound phase) or resolved by precomputation are called internal conflicts. This type of conflicts can only be solved in a brute-force fashion. For example, the conflicts marked by the underline in step 13 of 7-round attack AES-256 in Section 5.2. The nonlinear equations about these conflicts are too complicated to be precomputed. The number of type III conflicts will greatly affect the complexity.

Let the numbers of Type I/II/II conflicts be c_1, c_2, c_3, where $c_{in} = c_1 + c_2 + c_3$. Then, after addressing the conflicts in different ways, the time complexity of the GD to find one starting point is about $\mathcal{T}'_{\mathsf{GD}} = \mathcal{T}_{\mathsf{GD}}/2^{8(c_1+c_2)} = 2^{8c_3}$. The probability of the Outbound decreases to $2^{-p_{out}-(7 \text{ or } 6)\cdot c_1}$. The overall time complexity of the GD rebound will be

$$\mathcal{T} = 2^{8c_3} \cdot 2^{p_{out}+(7 \text{ or } 6)\cdot c_1}.$$

If $\mathcal{T} > 2^{n/2}$, add one more round (or a partial round) of the KS or EN path into the Inbound and update the probability of the Outbound phase. Run Buillaguet *et al.*'s guess-and-determine tool [5] to find a new GD for the new Inbound and analyze the conflicts. If $\mathcal{T} < 2^{n/2}$, we can still repeat the above steps to find a possible better attack.

Initially, with the short Inbound phase, the probability $2^{-p_{out}}$ of the Outbound phase is usually very low, leading to the complexity exceeding the birthday paradox. As the range of the Inbound phase increases, the probability of outbound will increase, but the number of conflicts could also increase. Our algorithm can find a balance between the time to solve the conflicts 2^{8c_3} and the time $2^{p_{out}}$ for the Outbound phase, leading to a better overall time complexity.

Summary of the GD Rebound Attack. After determining the related-key differential suitable for key collision (**Step 1**) and the Inbound phase (**Step 2**), we can conduct the full GD rebound attack as follows.

1. For the Inbound differential with s_1 active Sboxes of 2^{-7} probability and s_2 active Sboxes of 2^{-6} probability, we can determine $2^{(s_1+2s_2)-1}$ choices for the combinations of the known bytes in the Inbound.
2. In the GD steps of the Inbound, assuming that the number of guessed bytes is g, there are $2^{(s_1+2s_2)-1+8g}$ choices for the combinations of the known bytes and guessed bytes in the Inbound. Note that in the final Inbound phase, there are no Type I conflicts (removed to Outbound), only Type II and Type III conflicts, *i.e.*, $c_1 = 0$ and $c_{in} = c_2 + c_3$.
3. If $c_2 > 0$, precompute to solve the c_2 conflicts. Otherwise, skip this step.
4. Choosing $2^{8c_3+p_{out}}$ combinations of known and guessed bytes, run the GD steps to obtain $2^{p_{out}}$ starting points. Then, calculate whether the starting points satisfy the Outbound differential. One collision is expected.

   The time complexity of finding one starting point is $\mathcal{T}_{\mathsf{GD}} = 2^{8c_3}$, and the overall time complexity of the GD rebound is $\mathcal{T} = 2^{8c_3} \cdot 2^{p_{out}}$.

   Note that in **Step 1** to choose a RKDC, the degrees of freedom are already taken into account and there should be some key pairs that satisfy the full RKDC (thus leading to collisions). In the concrete attack, the total degree of freedom of the Inbound is $2^{(s_1+2s_2)-1+8g}$, and the consumed degree of freedom to precompute the c_2 Type II conflicts is 2^{8c_2}. Since the total probability of finding the final collision is $2^{-(8c_3+p_{out})}$, it is expected that $2^{(s_1+2s_2)-1+8g-8c_2} \geq 2^{(8c_3+p_{out})}$ according to the property of the RKDC.

4 Key Collision Attacks on Reduced AES-128

This section discusses the fixed-target-plaintext key collision on 2-round AES-128 in [43], and then gives practical key collision attacks on 2-/3-round AES-128.

4.1 The Invalid Key Collision on 2-round AES-128 in [43]

In [43], Taiyama *et al.* gave a fixed-target-plaintext key collision attack on 2-round AES-128. Their underlying differential characteristic is shown in Figure 6, which has a probability of 2^{-98}. In their attack, the round 0 in the EN path is the inbound phase with a probability of 2^{-42}, and the remaining parts including the key schedule are the outbound phase with a probability of 2^{-56}.

At the beginning of their attack, 2^{14} values of 4-byte $x_0[12, 13, 14, 15]$ are chosen. Then with fixed plaintext P, compute 2^{14} values of $k_0[12, 13, 14, 15]$. Since the input difference $\Delta k_0[12]$ and the output difference of $\Delta\mathsf{SB}(k_0[12])$ are fixed with a probability of 2^{-7}, the authors hope that there are $2^{7=(14-7)}$ values remaining. Focusing on the value of $x_0[12]$, since $\Delta x_0[12] = \texttt{0x69}$ and $\Delta\mathsf{SB}(x_0[12]) = \texttt{0xef}$, there are only two possible values of $x_0[12]$, *i.e.*, 0x1b and 0x72. For $k_0[12]$, since $\Delta k_0[12] = \texttt{0x69}$ and $\Delta\mathsf{SB}(k_0[12]) = \texttt{0x08}$, there are also only two possible values of $k_0[12]$, *i.e.*, 0x02 and 0x6b. So $P[12]$ is fixed according to $k_0[12] = P[12] \oplus x_0[12]$ for this differential.

- CASE-1: When $x_0[12]$ is fixed to 0x1b or 0x72 in all 2^{14} values of $x_0[12, 13, 14, 15]$, $P[12]$ should be fixed to corresponding values to satisfy the differential, *i.e.*,

$$(x_0[12], P[12]) \in \{(\texttt{0x1b}, \texttt{0x19}), (\texttt{0x1b}, \texttt{0x70}), (\texttt{0x72}, \texttt{0x70}), (\texttt{0x72}, \texttt{0x19})\}.$$

  In this case, all the 2^{14} values will remain.
- CASE-2: When $x_0[12]$ varies and $(x_0[12], P[12])$ is not among the value pairs in CASE-1, all the 2^{14} values of $x_0[12, 13, 14, 15]$ do not satisfy the differences in $\Delta k_0[12]$ and $\Delta\mathsf{SB}(k_0[12])$. No collision can be found.

As in the discussion above, the key collision attack for 2-round AES-128 in [43] is only valid for some plaintexts with fixed values in $P[12]$, and requires careful selection of $x_0[12]$. For other plaintexts, including $P = 0$, one cannot find a key pair that generates the same ciphertext.

4.2 The Practical Key Collision Attack on 2-round AES-128

We give a new key collision attack on 2-round AES-128 based on a new related-key differential characteristic as shown in Figure 8, whose probability is 2^{-107}. We choose the differential with $\Delta k_0[j] = 0$ ($j \in [12, 13, 14, 15]$) to avoid the restriction on the plaintext as in Section 4.1. Only the last round of the key schedule is the outbound phase. Since $\Delta k_1[13] = \texttt{0xf4}$ and $\Delta\mathsf{SB}(k_1[13]) = \texttt{0xdc}$, the probability is $2^{-p_{out}} = 2^{-6}$. The remaining parts are the inbound phase. The steps of the GD for the inbound phase are marked in Figure 9 and the detailed equations are listed in Table 2.

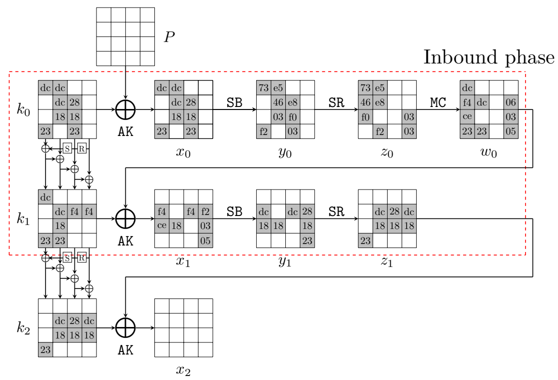

Fig. 8: The new related-key differential characteristic on 2-round AES-128

Guess-and-determine procedures of the inbound phase.

1. With the fixed differences in $\Delta x_0[0,3-6,9-11]$ and $\Delta y_0[0,3-6,9-11]$, we can deduce $x_0[0,3-6,9-11]$ and $y_0[0,3-6,9-11]$ (marked by $\boxed{1}$ in Figure 9) by accessing the DDT. Similarly, deduce $x_1[1,2,6,9,13,14,15]$ and $y_1[1,2,6,9,13,14,15]$ (marked by $\boxed{1}$).
   (a) In round 0, compute $k_0[0,3,4,5,6,9,10,11] = (x_0 \oplus P)[0,3,4,5,6,9,10,11]$ (marked by $\overleftarrow{1}$).
   (b) Compute forward to get $z_0[0,1,2,4,5,7,14,15]$ and $z_1[3,5,6,9,10,13,14]$ (marked by $\overrightarrow{1}$)
2. Guess $k_0[15]$ (marked by $\boxed{2}$), and compute forward to get $x_0[15]$, $y_0[15]$ and $z_0[3]$ (marked by $\overrightarrow{2}$). Then compute $w_0[0,1,2,3] = \mathsf{MC}(z_0[0,1,2,3])$. Since $x_1[1,2]$ are known, compute $k_1[1,2] = x_1[1,2] \oplus w_0[1,2]$ (marked by $\overleftarrow{2}$).
3. According to the key relations, we can deduce $k_1[5,6,9,10]$ and $k_0[2]$ (marked by $\boxed{3}$) with equations given in Table 2.
   (a) Compute forward to get $x_0[2]$, $y_0[2]$ and $z_0[10]$ (marked by $\overrightarrow{3}$).
   (b) Compute backward to get $w_0[6,9] = x_1[6,9] \oplus k_1[6,9]$ (marked by $\overleftarrow{3}$).
4. For column 1 over the MC operation in round 0, four values in the inputs and outputs are known, and we can deduce the other four values. That is, deduce $z_0[6]$ and $w_0[4,5,7]$ (marked by $\boxed{4}$) from $z_0[4,5,7]$ and $w_0[6]$.
   (a) Compute backward to get $k_0[14] = P[14] \oplus \mathsf{SB}^{-1}(z_0[6])$ (marked by $\overleftarrow{4}$).
   (b) Compute forward to get $x_1[5]$, $y_1[5]$ and $z_1[1]$ (marked by $\overrightarrow{4}$).
5. According to the key relations, deduce $k_0[1]$ and $k_1[14]$ (marked by $\boxed{5}$). Compute forward to get $z_0[13]$ (marked by $\overrightarrow{5}$) and compute backward to get $w_0[14]$ (marked by $\overleftarrow{5}$).
6. For column 3 of round 0, deduce $w_0[12,13,15]$ and $z_0[12]$ (marked by $\boxed{6}$) from $z_0[13,14,15]$ and $w_0[14]$.
   (a) Compute backward to get $k_0[12]$ (marked by $\overleftarrow{6}$).
   (b) Compute forward to get $k_1[13,15]$ (marked by $\overrightarrow{6}$).

7. According to the key relations, deduce $k_1[0, 3, 4, 7, 11]$ and $k_0[7, 13]$ (marked by $\boxed{7}$). Compute forward to get $z_0[9, 11]$ and $z_1[0, 4, 7, 11]$ (marked by $\overrightarrow{7}$).
8. For column 2 over the MC operation in round 0, deduce $z_0[8]$ and $w_0[8, 10, 11]$ (marked by $\boxed{8}$) from $z_0[9, 10, 11]$ and $w_0[9]$.
   (a) Compute backward to get $k_0[8]$ (marked by $\overleftarrow{8}$).
   (b) Compute forward to get $z_1[2, 15]$ (marked by $\overrightarrow{8}$).
9. According to the key relations, deduce $k_1[8, 12]$ (marked by $\boxed{9}$). Then we get all the states of the starting point.

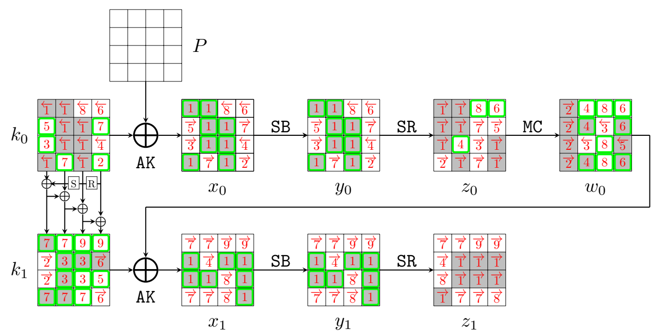

Fig. 9: Steps of the GD in the inbound phase for 2-round AES-128. *The green border bytes are the known or guessed bytes at the beginning of each step, which are used to deduce other bytes. E.g., in Step 2, $k_0[15]$ marked by 2 with a green border is guessed at the beginning of Step 2, then it is used to deduce $x_0[15]$, $y_0[15]$ and $z_0[3]$, etc.*

Degree of freedom and complexity.

– In step 1 of the above procedures, we deduce the values for active bytes from the input/output differences in the inbound phase. There are 15 active Sboxes with a total probability 2^{-101}, including $s_1 = 11$ active Sboxes with probability 2^{-7} and $s_2 = 4$ active Sboxes with probability 2^{-6}. Therefore, there are $2^{11+8}/2 = 2^{18}$ combinations for the 15 active bytes, *i.e.*, there are 2^{18} choices for the bytes marked by $\boxed{1}$ with a green border in Figure 9.
– Given one out of 2^{18} choices marked by $\boxed{1}$, one byte $k_0[15]$ (marked by a wavy line) is guessed in step 2. Therefore, there expect $2^{18+8} = 2^{26}$ states satisfying the inbound trial in total, which act as the starting points for the outbound phase.
– Since there is no conflict in the inbound phase, *i.e.*, $c_{in} = 0$, the time of the GD to find one starting point is $\mathcal{T}_{\mathsf{GD}} = 1$. Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-6}$, we need to collect 2^6 starting points to expect one collision. The overall time complexity is only $\mathcal{T} = 2^6$ and the

1.	$k_0[0,3,4,5,6,9,10,11] = (x_0 \oplus P)[0,3,4,5,6,9,10,11]$	
2.	$z_0[3] = \mathsf{SB}(P[15] \oplus k_0[15])$	$w_0[0,1,2,3] = \mathsf{MC}(z_0[0,1,2,3])$
	$k_1[1,2] = x_1[1,2] \oplus w_0[1,2]$	
3.	$k_1[5] = k_0[5] \oplus k_1[1]$	$k_1[6] = k_0[6] \oplus k_1[2]$
	$k_1[9] = k_0[9] \oplus k_1[5]$	$k_1[10] = k_0[10] \oplus k_1[6]$
	$k_0[2] = k_1[2] \oplus \mathsf{SB}(k_0[15])$	
4.	$w_0[4,5,7], z_0[6] = \mathtt{MC}(z_0[4,5,7], w_0[6])$	$k_0[14] = P[14] \oplus \mathsf{SB}^{-1}(z_0[6])$
5.	$k_0[1] = k_1[1] \oplus \mathsf{SB}(k_0[14])$	$k_1[14] = k_1[10] \oplus k_0[14]$
6.	$w_0[12,13,15], z_0[12] = \mathtt{MC}(z_0[13,14,15], w_0[14])$	$k_0[12] = P[12] \oplus \mathsf{SB}^{-1}(z_0[12])$
	$k_1[13] = w_0[13] \oplus x_1[13]$	$k_1[15] = w_0[15] \oplus x_1[15]$
7.	$k_1[3] = k_0[3] \oplus \mathsf{SB}(k_0[12])$	$k_1[11] = k_0[15] \oplus k_1[15]$
	$k_1[7] = k_1[11] \oplus k_0[11]$	$k_0[7] = k_1[7] \oplus k_1[3]$
	$k_0[13] = k_1[13] \oplus k_1[9]$	$k_1[0] = k_0[0] \oplus \mathsf{SB}(k_0[13]) \oplus const$
	$k_1[4] = k_0[4] \oplus k_1[0]$	
8.	$w_0[8,10,11], z_0[8] = \mathtt{MC}(z_0[9,10,11], w_0[9])$	$k_0[8] = P[8] \oplus \mathsf{SB}^{-1} z_0[8]$
9.	$k_1[8] = k_0[8] \oplus k_1[4]$	$k_1[12] = k_0[12] \oplus k_1[8]$

Table 2: Equations in the GD steps for 2-round AES-128. Blue bytes are guessed

memory complexity is negligible. We could find the key collisions in seconds on a desktop equipped with Intel Core i7-13700F @2.1 GHz 396 and 16G RAM, and some examples are listed in Table 6.

4.3 The Practical Key Collision Attack on 3-round AES-128

We give a new key collision attack on 3-round AES-128 based on a new related-key differential characteristic as shown in Figure 10. There is one active $k_0[15]$ in the first round key, i.e. $\Delta k_0[15] = \mathtt{0xcc}$, which brings the same difference to $x_0[15]$. Applying the observation in Section 3.2, to prevent the restriction on P, we set $\Delta\mathsf{SB}(k_0[15]) = \Delta\mathsf{SB}(x_0[15]) = \mathtt{0x28}$, and keep $x_0[15] = k_0[15]$ in the attack, which makes $P[15] = 0$. So when we choose the value of $x_0[15]$ satisfying the difference over the active Sbox in the EN path, the value of $k_0[15]$ satisfies the difference over the active Sbox in the KS with probability 1. Therefore, although there are 19 active Sboxes in the differential, we only count the probability of 18 of them, which is 2^{-125}. We choose the first two rounds of the EN and KS as the inbound phase, with a probability of 2^{-90}. The remaining parts are the outbound phase, with a probability of $2^{-p_{out}} = 2^{-35}$. The steps of the GD for the inbound phase are marked in Figure 11 with equations listed in Table 3.

Guess-and-determine procedures of the inbound phase.

1. With the fixed differences in $\Delta x_0[0, 2-4, 7, 15]$ and $\Delta y_0[0, 2-4, 7, 15]$, we can deduce $x_0[0, 2-4, 7, 15]$ and $y_0[0, 2-4, 7, 15]$ (marked by $\boxed{1}$ in Figure 11) by accessing the DDT. Similarly, deduce $x_1[1, 3, 4, 6, 9, 12]$ and $y_1[1, 3, 4, 6, 9, 12]$ (marked by $\boxed{1}$).
   (a) In round 0, deduce $k_0[0, 2, 3, 4, 7, 15] = (x_0 \oplus P)[0, 2, 3, 4, 7, 15]$ (marked by $\overleftarrow{1}$). Compute forward to $z_0[0, 3, 4, 7, 10, 11]$ (marked by $\overrightarrow{1}$).

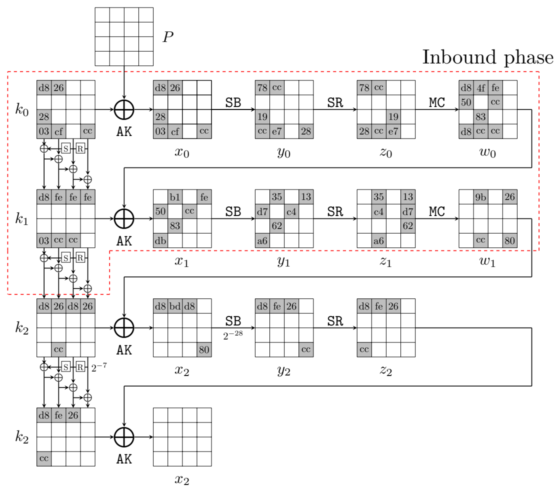

Fig. 10: The related-key differential characteristic on 3-round AES-128

(b) In round 1, since the differences $\Delta k_1[12]$ and $\Delta \mathsf{SB}(k_1[12])$ are known, deduce $k_1[12]$ (marked by $\boxed{1}$) by accessing the DDT. Compute backward to get $w_0[12]$ (marked by $\overleftarrow{1}$) and compute forward to get $z_1[4,5,7,12,13,14]$ (marked by $\overrightarrow{1}$) .

2. Guess $k_0[5,12]$ (marked by $\boxed{2}$). According to the key relations, deduce $k_1[2,3,7,8]$ (marked by $\boxed{2}$) as Table 3.
   (a) Compute forward to get $x_0[5,12]$, $y_0[5,12]$ and $z_0[1,12]$ (marked by $\overrightarrow{2}$).
   (b) Compute backward to get $w_0[3] = k_1[3] \oplus x_1[3]$ (marked by $\overleftarrow{2}$).
3. For column 0 over the MC operation of round 0, deduce $w_0[0,1,2]$ and $z_0[2]$ (marked by $\boxed{3}$) from $z_0[0,1,3]$ and $w_0[3]$.
   (a) Compute backward to get $x_0[10]$ and $k_0[10]$ (marked by $\overleftarrow{3}$).
   (b) Compute forward to get $k_1[1] = w_0[1] \oplus x_1[1]$ and $z_1[10]$ (marked by $\overrightarrow{3}$).
4. Guess $k_0[13]$ (marked by $\boxed{4}$). According to the key relations, deduce $k_0[8]$ and $k_1[0,4,5]$ (marked by $\boxed{4}$) as Table 3.
   (a) Compute forward to get $z_0[8,9]$ and $z_1[0]$ (marked by $\overrightarrow{4}$).
   (b) Compute backward to get $w_0[4]$ (marked by $\overleftarrow{4}$).
5. For column 2 over the MC operation of round 0, deduce $w_0[8,9,10,11]$ (marked by $\boxed{5}$) from $z_0[8,9,10,11]$. Compute forward to get $k_1[9] = w_0[9] \oplus x_1[9]$ and $z_1[8]$ (marked by $\overrightarrow{5}$).
6. According to the key relations, deduce $k_0[9]$ and $k_1[13]$ (marked by $\boxed{6}$). Compute forward to get $z_0[5]$ (marked by $\overrightarrow{6}$).
7. For column 1 over the MC operation of round 0, deduce $w_0[5,6,7]$ and $z_0[6]$ (marked by $\boxed{7}$) from $z_0[4,5,7]$ and $w_0[4]$.
   (a) Compute backward to get $x_0[14]$ and $k_0[14]$ (marked by $\overleftarrow{7}$).
   (b) Compute forward to get $k_1[6]$ and $z_1[1,11]$ (marked by $\overrightarrow{7}$).

8. According to the key relations, deduce $k_0[1,6]$ and $k_1[10,14]$ (marked by $\boxed{8}$). Compute forward to get $z_0[13,14]$ and $z_1[2]$ (marked by $\overrightarrow{8}$).
9. For column 3 over the MC operation of round 0, deduce $w_0[13,14,15]$ and $z_0[15]$ (marked by $\boxed{9}$) from $z_0[12,13,14]$ and $w_0[12]$.
   (a) Compute backward to get $x_0[11]$ and $k_0[11]$ (marked by $\overleftarrow{9}$).
   (b) Compute forward to get $z_1[6,9]$ and columns 1,2 of w_1 (marked by $\overrightarrow{9}$).
10. According to the key relations, deduce $k_1[11,15]$ (marked by $\boxed{10}$). Compute forward to get $z_1[3,15]$ and columns 0,3 of w_1 (marked by $\overrightarrow{10}$). Then we get all the states of the starting point.

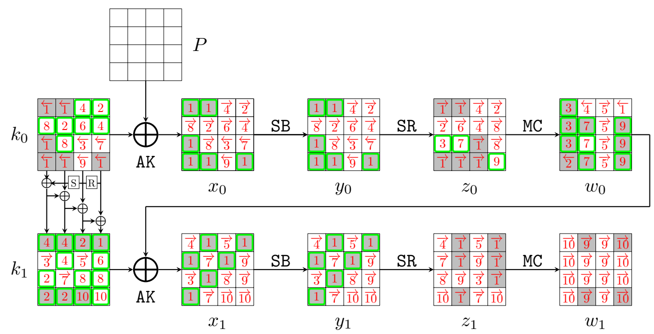

Fig. 11: Steps of the GD in the inbound phase for 3-round AES-128

1.	$k_0[0,2,3,4,7,15] = (x_0 \oplus P)[0,2,3,4,7,15]$	$w_0[12] = k_1[12] \oplus x_1[12]$
2.	$k_1[3] = k_0[3] \oplus \mathsf{SB}(k_0[12])$	$k_1[7] = k_0[7] \oplus k_1[3]$
	$k_1[8] = k_1[12] \oplus k_0[12]$	$k_1[2] = k_0[2] \oplus \mathsf{SB}(k_0[15])$
	$z_0[1] = \mathsf{SB}(k_0[5] \oplus P[5])$	
3.	$w_0[0,1,2], z_0[2] = \mathtt{MC}(z_0[0,1,3], w_0[3])$	$k_0[10] = P[10] \oplus \mathsf{SB}^{-1}(z_0[2])$
	$k_1[1] = w_0[1] \oplus x_1[1]$	
4.	$k_1[0] = k_0[0] \oplus \mathsf{SB}(k_0[13]) \oplus const$	$k_1[4] = k_0[4] \oplus k_1[0]$
	$k_0[8] = k_1[8] \oplus k_1[4]$	$k_1[5] = k_0[5] \oplus k_1[1]$
5.	$w_0[8,9,10,11] = \mathtt{MC}(z_0[8,9,10,11])$	$k_1[9] = w_0[9] \oplus x_1[9]$
6.	$k_0[9] = k_1[9] \oplus k_1[5]$	$k_1[13] = k_1[9] \oplus k_0[13]$
7.	$w_0[5,6,7], z_0[6] = \mathtt{MC}(z_0[4,5,7], w_0[4])$	$k_0[14] = P[14] \oplus \mathsf{SB}^{-1}(z_0[6])$
	$k_1[6] = w_0[6] \oplus x_1[6]$	
8.	$k_0[1] = k_1[1] \oplus \mathsf{SB}(k_0[14])$	$k_0[6] = k_1[6] \oplus k_1[2]$
	$k_1[10] = k_1[6] \oplus k_0[10]$	$k_1[14] = k_1[10] \oplus k_0[14]$
9.	$w_0[13,14,15], z_0[15] = \mathtt{MC}(z_0[12,13,14], w_0[12])$	$k_0[11] = P[11] \oplus \mathsf{SB}^{-1}(z_0[15])$
10.	$k_1[11] = k_0[11] \oplus k_1[7]$	$k_1[15] = k_1[11] \oplus k_0[15]$

Table 3: Equations in the GD steps for 3-round AES-128. Blue bytes are guessed.

Degree of freedom and complexity.

- In step 1, we deduce the values for active bytes from the input/output differences in the inbound phase. There are 13 active Sboxes with a total probability 2^{-90}, including $s_1 = 12$ active Sboxes with probability 2^{-7} and $s_2 = 1$ active Sboxes with probability 2^{-6}. Therefore, there are $2^{12+2}/2 = 2^{13}$ combinations for the 13 active bytes, *i.e.*, there are 2^{13} choices for the bytes marked by $\boxed{1}$ in Figure 11.
- Given one out of 2^{13} choices marked by $\boxed{1}$, three bytes $k_0[5, 12, 13]$ (marked by a wavy line) are guessed in step 2 and 4. Therefore, there expect $2^{13+24} = 2^{37}$ states satisfying the inbound trial in total, which act as the starting points for the outbound phase.
- Since there is no conflict in the inbound phase, *i.e.*, $c_{in} = 0$, the time of the GD to find one starting point is $\mathcal{T}_{\mathsf{GD}} = 1$. Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-35}$, we need to collect 2^{35} starting points to expect one collision. The overall time complexity is $\mathcal{T} = 2^{35}$ and the memory complexity is negligible, which is practical. We find key collisions in several hours on a desktop equipped with Intel Core i7-13700F @2.1 GHz and 16G RAM using one CPU core, and some examples are listed in Table 6.

5 Key Collision Attacks on Reduced AES-256

In this section, we give a practical key collision attack on 6-round AES-256 and a quantum attack on 7-round AES-256. We also discuss the fixed-target-plaintext key collision on 6-round AES-256 in [43] in Supplementary Material B.

5.1 Practical Key Collision Attack on 6-round AES-256

We find a new related-key differential characteristic on 6-round AES-256 with a probability of 2^{-214}, which is shown in Figure 15 in Supplementary Material B. Compared to the differential in Figure 14 in [43], the two differentials follow the same related-key truncated differential, but are different instantiations. The inbound phase covers the first four rounds and has 28 active Sboxes with a probability of 2^{-193}, including 6 active Sboxes in the key schedule. The probability of the outbound phase is $2^{-p_{out}} = 2^{-21}$. The steps of the GD are listed below and in Figure 12. The detailed equations are listed in Table 4.

Guess-and-determine procedures of the inbound phase.

1. Deduce the values of $x_0[0, 1, 2, 3, 13]$, $y_0[0, 1, 2, 3, 13]$, $x_1[0, 1, 3-6, 9-12, 14, 15]$, $y_1[0, 1, 3-6, 9-12, 14, 15]$, $x_2[2, 4, 15]$, $y_2[2, 4, 15]$, $x_3[1, 6]$ and $y_3[1, 6]$ with the fixed differences by accessing the DDT, which are all marked by $\boxed{1}$. Similarly, deduce the values of $k_1[12, 13, 14, 15]$ and $k_2[13]$ (marked by $\boxed{1}$) by accessing the DDT, according to the fixed $\Delta k_1[12, 13, 14, 15]$, $\Delta k_2[13]$, $\Delta\mathsf{SB}(k_1[12, 13, 14, 15])$ and $\Delta\mathsf{SB}(k_2[13])$.

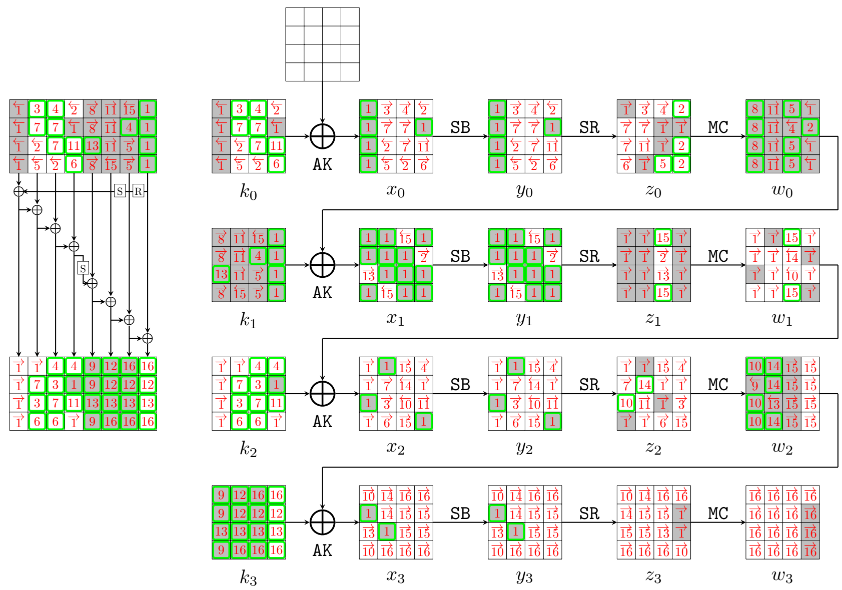

Fig. 12: Steps of the GD in the inbound phase for 6-round AES-256

(a) According to the known values, we have

$$y_1[0] \oplus y_1[5] \oplus 02 \cdot y_1[10] \oplus 03 \cdot y_1[15] \oplus x_2[2] \oplus x_0[2] \oplus P[2] \oplus \mathsf{SB}(k_1[15]) = 0, \tag{7}$$

which is a conflict of Type II. The bytes marked by red are known by accessing the DDT. We precompute the values of $y_1[0,5,10,15]$, $x_2[2]$, $x_0[2]$ and $k_1[15]$ to satisfy Equation 7 and solve the conflict. Note that the same conflict also exists in the differential in [43], and they can not fulfill Equation 7 for $P = 0$ (see details in Supplementary Material B.1).

(b) In round 0, compute backward to get $k_0[0,1,2,3,13]$ (marked by $\overleftarrow{1}$). Compute forward to $z_0[0,7,9,10,13]$ (marked by $\overrightarrow{1}$).

(c) In round 1, compute backward to $w_0[12,14,15]$ (marked by $\overleftarrow{1}$). Compute $\mathtt{MC} \circ \mathtt{SR}(y_1)$ and get columns $0,1,3$ of z_1 and w_1 (marked by $\overrightarrow{1}$).

(d) In round 2, compute $k_2[2,4,15]$ (marked by $\overrightarrow{1}$) from $w_1[2,4,15]$ and $x_2[2,4,15]$. According to the key relations, compute $k_2[0,1,2,3]$ (marked by $\overrightarrow{1}$) from $k_0[0,1,2,3]$ and $k_1[12,13,14,15]$. As step 1(a), the two values of $k_2[2]$ computed are equal of probability 1 after solving the conflict. Then deduce $x_2[0,1,3,13]$ and $z_2[0,3,4,7,9,10,13]$ (marked by $\overrightarrow{1}$).

(e) In round 3, compute forward to $z_3[13,14]$ (marked by $\overleftarrow{1}$).

2. For column 3 over the MC operation in round 0, compute $z_0[12,14,15]$ and $w_0[13]$ (marked by $\boxed{2}$) from $z_0[13]$ and $w_0[12,14,15]$.
   (a) Compute backward to get $k_0[6,11,12]$ (marked by $\overleftarrow{2}$).
   (b) Compute forward to $x_1[13]$ as well as $z_1[9]$ (marked by $\overrightarrow{2}$).

3. According to the key relations, deduce the key values $k_0[4], k_2[6,9]$ (marked by $\boxed{3}$). Compute forward to $z_0[4]$ and $z_2[14]$ (marked by $\overrightarrow{3}$).
4. Guess $k_0[8]$ and $k_1[9]$ (marked by $\boxed{4}$), and deduce $k_2[8,12]$ (marked by $\boxed{4}$) according to the key relations.
   (a) Compute forward to $z_0[8]$ (marked by $\overrightarrow{4}$) in round 0, and to $z_2[12]$ (marked by $\overrightarrow{4}$) in round 2.
   (b) Compute backward to $w_0[9]$ (marked by $\overleftarrow{4}$) in round 1.
5. For column 2 over the MC operation in round 0, compute $w_0[8,10,11]$ and $z_0[11]$ (marked by $\boxed{5}$) from $z_1[8,9,10]$ and $w_1[9]$.
   (a) Compute forward to $k_1[10,11]$ (marked by $\overrightarrow{5}$).
   (b) Compute backward to $k_0[7]$ (marked by $\overleftarrow{5}$) in round 0.
6. According to the key relations, compute $k_2[7,11]$ and $k_0[15]$ (marked by $\boxed{6}$), and compute forward to get $z_0[3]$ and $z_2[11]$ (marked by $\overrightarrow{6}$).
7. Guess $k_0[5]$ and $k_0[10]$ (marked by $\boxed{7}$), and deduce $k_2[5,10]$ and $k_0[9]$ (marked by $\boxed{7}$) according to the key relations. Then compute forward to $z_0[1,2,5]$ and $z_2[1]$ (marked by $\overrightarrow{7}$).
8. For column 0 over the MC operation in round 0, compute $w_0[0,1,2,3]$ (marked by $\boxed{8}$) from $z_0[0,1,2,3]$. Then deduce $k_1[0,1,3] = x_1[0,1,3] \oplus w_0[0,1,3]$ (marked by $\overrightarrow{8}$).
9. According to the key relations, we can deduce $k_3[0,1,3]$ (marked by $\boxed{9}$). Then compute backward to $w_2[1]$ (marked by $\overleftarrow{9}$).
10. For column 0 over the MC operation in round 2, compute $w_2[0,2,3]$ and $z_2[2]$ (marked by $\boxed{10}$) from $z_2[0,1,3]$ and $w_2[1]$.
    (a) Compute backward to $x_2[10]$ and $w_1[10]$ (marked by $\overleftarrow{10}$) in round 2.
    (b) Compute forward to $x_3[0,3]$ and $z_3[0,15]$ (marked by $\overrightarrow{10}$) in round 3.
11. Guess $k_0[14]$ (marked by $\boxed{11}$) and deduce $k_2[14]$ (marked by $\boxed{11}$). Compute forward to $z_0[6]$ and $z_2[6]$ (marked by $\overrightarrow{11}$), and deduce $w_0[4,5,6,7] = \mathtt{MC}(z_0[4,5,6,7])$ (marked by $\overrightarrow{11}$). Deduce $k_1[4,5,6] = x_1[4,5,6] \oplus w_0[4,5,6]$ (marked by $\overrightarrow{11}$).
12. According to the key relations, we deduce $k_3[4,5,9,13]$ (marked by $\boxed{12}$).
13. Guess $k_1[2]$ and deduce $k_3[2,6,10,14]$ (marked by $\boxed{13}$).
    (a) Compute forward to get $z_1[10]$ and $z_3[10]$ (marked by $\overrightarrow{13}$).
    (b) Compute backward to get $w_2[6]$ (marked by $\overleftarrow{13}$).
14. For column 1 over the MC operation in round 2, compute $w_2[4,5,7]$ and $z_2[5]$ (marked by $\boxed{14}$) from $z_2[4,6,7]$ and $w_2[6]$.
    (a) Compute backward in round 2 to $w_1[9]$ (marked by $\overleftarrow{14}$).
    (b) Compute forward in round 3 to $x_3[4,5]$ and $z_3[4,1]$ (marked by $\overrightarrow{14}$).
15. For column 2 over the MC operation in round 1, compute $z_1[8,11]$ and $w_1[8,11]$ (marked by $\boxed{15}$) from $z_1[9,10]$ and $w_1[9,10]$.
    (a) Compute backward in round 1 to $x_1[7,8]$ and $k_1[7,8]$ (marked by $\overleftarrow{15}$).
    (b) Compute forward in round 2 to $x_2[8,11]$ and $z_2[8,15]$ (marked by $\overrightarrow{15}$). Deduce columns 2 and 3 of w_2 and $z_3[2,5,6,9]$ (marked by $\overrightarrow{15}$).
16. According to the key relations, compute $k_3[7,8,11,12,15]$ (marked by $\boxed{16}$). Compute $w_3 = \mathsf{MC} \circ \mathsf{SR} \circ \mathsf{SB}(k_3 \oplus w_2)$. Deduce all states of the starting point.

1.	$k_0[0,1,2,3,13] = (x_0 \oplus P)[0,1,2,3,13]$	$w_1[0,1,2,3] = \mathtt{MC}(z_1[0,1,2,3])$
	$w_1[4,5,6,7] = \mathtt{MC}(z_1[4,5,6,7])$	$w_1[12,13,14,15] = \mathtt{MC}(z_1[12,13,14,15])$
	$k_2[2,4,15] = x_2[2,4,15] \oplus w_1[2,4,15]$	$k_2[0] = k_0[0] \oplus \mathtt{SB}(k_1[13]) \oplus const$
	$k_2[1,2,3] = k_0[1,2,3] \oplus \mathtt{SB}(k_1[14,15,12])$	$k_2[2] = w_1[2] \oplus x_2[2] \overset{?}{=} k_0[2] \oplus \mathsf{SB}(k_1[15])$
2.	$z_0[12,14,15], w_0[13] = \mathtt{MC}^{-1}(z_0[13], w_0[12,14,15])$	$k_0[6] = P[6] \oplus \mathsf{SB}^{-1}(z_0[14])$
	$k_0[11] = P[11] \oplus \mathsf{SB}^{-1}(z_0[15])$	$k_0[12] = P[12] \oplus \mathsf{SB}^{-1}(z_0[12])$
3.	$k_0[4] = k_2[4] \oplus k_2[0]$	$k_2[6] = k_0[6] \oplus k_2[2]$
	$k_2[9] = k_0[13] \oplus k_2[13]$	
4.	$k_2[8] = k_0[8] \oplus k_2[4]$	$k_2[12] = k_0[12] \oplus k_2[8]$
	$w_0[9] = k_1[9] \oplus x_1[9]$	
5.	$w_0[8,10,11], z_0[11] = \mathsf{MC}(z_0[8,9,10], w_0[9])$	$k_1[10,11] = w_0[10,11] \oplus x_1[10,11]$
	$k_0[7] = \mathsf{SB}^{-1}(z_0[11]) \oplus P[7]$	
6.	$k_2[7] = k_0[7] \oplus k_2[3]$	$k_2[11] = k_0[11] \oplus k_2[7]$
	$k_0[15] = k_2[15] \oplus k_2[11]$	
7.	$k_2[5] = k_0[5] \oplus k_2[1]$	$k_0[9] = k_2[5] \oplus k_2[9]$
	$k_2[10] = k_0[10] \oplus k_2[6]$	
8.	$w_0[0,1,2,3] = \mathsf{MC}(z_0[0,1,2,3])$	$k_1[0,1,3] = x_1[0,1,3] \oplus w_0[0,1,3]$
9.	$k_3[0,1,3] = k_1[0,1,3] \oplus \mathsf{SB}(k_2[12,13,15])$	
10.	$w_2[0,2,3], z_2[2] = \mathtt{MC}(z_2[0,1,3], w_2[1])$	
11.	$z_0[6] = \mathsf{SB}(k_0[14] \oplus x_0[14])$	$k_2[14] = k_0[14] \oplus k_2[10]$
	$w_0[4,5,6,7] = \mathsf{MC}(z_0[4,5,6,7])$	$k_1[4,5,6] = x_1[4,5,6] \oplus w_0[4,5,6]$
12.	$k_3[4] = k_1[4] \oplus k_3[0]$	$k_3[5] = k_1[5] \oplus k_3[1]$
	$k_3[9] = k_1[9] \oplus k_3[5]$	$k_3[13] = k_1[13] \oplus k_3[9]$
13.	$k_3[2] = k_1[2] \oplus \mathsf{SB}(k_2[14])$	$k_3[6] = k_1[6] \oplus k_3[2]$
	$k_3[10] = k_1[10] \oplus k_3[6]$	$k_3[14] = k_1[14] \oplus k_3[10]$
14.	$w_2[4,5,7], z_2[5] = \mathtt{MC}(z_2[4,6,7], w_2[6])$	
15.	$z_1[8,11], w_1[8,11] = \mathtt{MC}(z_1[9,10], w_1[9,10])$	$k_1[7,8] = \mathsf{SB}^{-1}(z_1[11,8]) \oplus w_0[7,8]$
	$z_2[8,15] = \mathsf{SB}(w_1[8,11] \oplus k_2[8,11])$	$w_2[8,9,10,11] = \mathsf{MC}(z_2[8,9,10,11])$
	$w_2[12,13,14,15] = \mathsf{MC}(z_2[12,13,14,15])$	
16.	$k_3[7] = k_1[7] \oplus k_3[3]$	$k_3[11] = k_1[11] \oplus k_3[7]$
	$k_3[15] = k_1[15] \oplus k_3[11]$	$k_3[8] = k_1[8] \oplus k_3[4]$
	$k_3[12] = k_1[12] \oplus k_3[8]$	$w_3 = \mathsf{MC} \circ \mathsf{SR} \circ \mathsf{SB}(k_3 \oplus w_2)$

Table 4: Equations in the guess-and-determine steps for 6-round AES-256. The blue bytes are guessed. The red equation is the conflict.

Degree of freedom and complexity.

- There are total 28 active Sboxes in the 4-round inbound phase, including $s_1 = 25$ active Sboxes with probability 2^{-7} and $s_2 = 3$ active Sboxes with probability 2^{-6}. There is $c_{in} = c_2 = 1$ conflict as Equation 7, and we fix the 7-byte values of $y_1[0,5,10,15]$, $x_2[2]$, $x_0[2]$ and $k_1[15]$ to satisfy Equation 7. Then, by accessing the DDT for the other 21 active Sboxes, we expect at least $2^{21}/2 = 2^{20}$ combinations for the 21 active Sboxes, *i.e.*, there are at least 2^{20} choices for the bytes marked by $\boxed{1}$ in Figure 12.
- Given one out of 2^{20} choices marked by $\boxed{1}$, six bytes $k_0[5,8,10,14]$ and $k_1[2,9]$ (marked by a wavy line) are guessed in step 4,7,11,13. Therefore,

there expect $2^{20+48} = 2^{68}$ states satisfying the inbound differential in total, which can act as the starting points for the outbound phase.

– Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-21}$, we have enough degrees of freedom to satisfy the outbound phase. The overall time complexity is $\mathcal{T} = 2^{21}$ and the memory complexity is negligible. We have practically implemented the attack and could find one key collision in several minutes. Some key pairs (K_1, K_2) are listed in Table 6 such that $\mathsf{AES\text{-}256}_{K_1}(0) = \mathsf{AES\text{-}256}_{K_2}(0)$, where AES-256 is a 6-round one.

5.2 Quantum Key Collision Attack on 7-round AES-256

We give a new quantum key collision attack on 7-round AES-256. The differential characteristic with a probability of 2^{-228} is shown in Figure 16 in Supplementary Material B. The inbound phase covers the first four rounds of the EN and KS path, which has 30 active Sboxes with a probability of 2^{-198}. The outbound phase has 5 active Sboxes, including 1 active Sboxes in the key schedule, with a probability of $2^{-p_{out}} = 2^{-30}$. In the GD procedure of inbound phase, there are $c_{in} = 5$ conflicts, where $c_1 = c_2 = 0$ and $c_3 = 5$. The guess-and-determine steps of the GD are listed as follows, as in Figure 13. The equations are listed in Table 5.

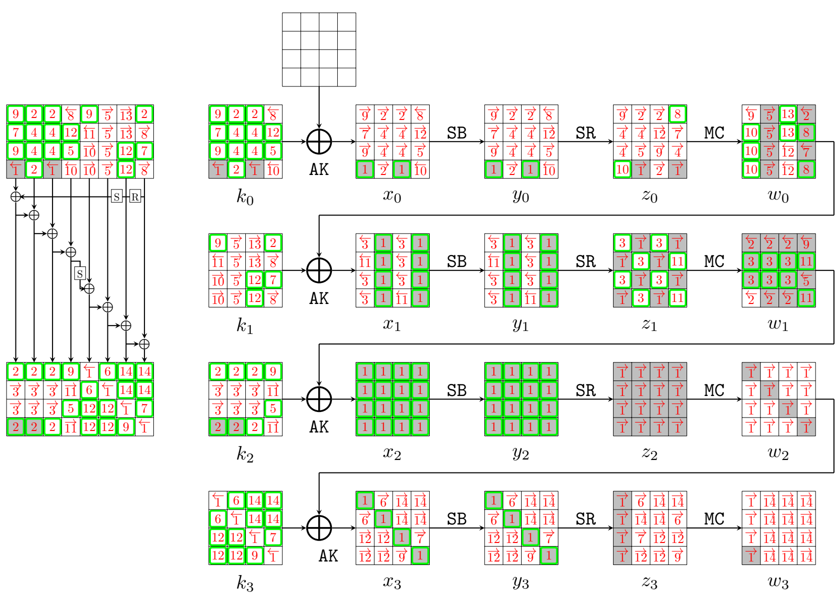

Fig. 13: Steps of the GD in the inbound phase for 7-round AES-256

Guess-and-determine procedures of the inbound phase.

1. Deduce the values of $x_0[3,11]$, $y_0[3,11]$, $x_1[4-7,12-15]$, $y_1[4-7,12-15]$, $x_2[0-15]$, $y_2[0-15]$, $x_3[0,5,10,15]$ and $y_3[0,5,10,15]$ with the fixed differences by accessing the `DDT`, which are all marked by $\boxed{1}$ in Figure 13.
   (a) In round 0, compute backward to get $k_0[3,11]$ (marked by $\overleftarrow{1}$), and compute forward to get $z_0[7,15]$ (marked by $\overrightarrow{1}$).
   (b) In round 1, compute forward to get $z_1[1,3,4,6,9,11,12,14]$ (marked by $\overrightarrow{1}$).
   (c) In round 2, compute forward to get the whole state $w_2 = \mathsf{MC} \circ \mathsf{SR}(y_2)$ (marked by $\overrightarrow{1}$).
   (d) In round 3, compute backward to deduce $k_3[0,5,10,15] = x_3[0,5,10,15] \oplus w_2[0,5,10,15]$ (marked by $\overleftarrow{1}$). Compute forward to get the $w_3[0,1,2,3]$ (marked by $\overrightarrow{1}$).
2. Guess $k_0[7]$, $k_1[12]$ and $k_2[0,4,8]$ (marked by $\boxed{2}$), then deduce the $k_0[4,8]$ and $k_2[3,7,11]$ (marked by $\boxed{2}$) according to the key relations.
   (a) In round 0, compute forward to get $z_0[4,8,11]$ (marked by $\overrightarrow{2}$).
   (b) In round 1, compute backward to get $w_0[12]$ (marked by $\overleftarrow{2}$).
   (c) In round 2, compute backward to get $w_1[0,3,4,7,8,11]$ (marked by $\overleftarrow{2}$).
3. For columns $0,1,2$ over the MC operation in round 1, compute $w_1[1,2,5,6,9,10]$ and $z_1[0,2,5,7,8,10]$ (marked by $\boxed{3}$) from $z_1[1,3,4,6,9,11]$ and $w_1[0,3,4,7,8,11]$.
   (a) Compute backward to get $x_1[0,2,3,8,9,10]$ (marked by $\overleftarrow{3}$).
   (b) Compute forward to get $k_2[1,2,5,6,9,10]$ (marked by $\overrightarrow{3}$).
4. According to the key relations, deduce $k_0[5,6,9,10]$ (marked by $\boxed{4}$). Compute forward then get $z_0[1,2,5,14]$ (marked by $\overrightarrow{4}$).
5. Guess $k_0[14]$ (marked by $\boxed{5}$) and deduce $k_2[14]$ (marked by $\boxed{5}$).
   (a) Compute forward to get $z_0[6]$ (marked by $\overrightarrow{5}$). Then compute $w_0[4,5,6,7]$ (marked by $\overrightarrow{5}$) from $z_0[4,5,6,7]$, and deduce $k_1[4,5,6,7]$ (marked by $\overrightarrow{5}$).
   (b) Compute backward to get $w_1[14]$ (marked by $\overleftarrow{5}$).
6. According to the key relations, deduce $k_3[1,4]$ (marked by $\boxed{6}$). Compute forward to get $x_3[1,4]$ and $z_3[4,13]$ (marked by $\overrightarrow{6}$).
7. Guess $k_0[1]$ (marked by $\boxed{7}$) and deduce $k_1[14]$ and $k_3[14]$ with the key relations (marked by $\boxed{7}$).
   (a) In round 0, compute forward get $x_0[1]$ and $z_0[13]$ (marked by $\overrightarrow{7}$).
   (b) In round 1, compute backward get $w_0[14]$ (marked by $\overleftarrow{7}$).
   (c) In round 3, compute forward get $x_3[14]$ and $z_3[6]$ (marked by $\overrightarrow{7}$).
8. For column 3 over the MC operation in round 0, compute $w_0[13,15]$ and $z_0[12]$ (marked by $\boxed{8}$) from $z_0[13,14,15]$ and $w_0[12,14]$. Since five values are known in the inputs/outputs over the MC operation, there is a conflict of Type III of 2^{-8} probability.
   (a) Compute forward to $k_1[13,15]$ (marked by $\overrightarrow{8}$).
   (b) Compute backward to $k_0[12]$ (marked by $\overleftarrow{8}$).
9. Deduce $k_0[0,2]$, $k_1[0]$, $k_2[12]$ and $k_3[11]$ (marked by $\boxed{9}$) by the key relations.
   (a) In round 0, compute forward to $x_0[0,2]$ and $z_0[0,10]$ (marked by $\overrightarrow{9}$).

(b) In round 1, compute backward to $w_0[0]$ (marked by $\overleftarrow{9}$).
(c) In round 2, compute backward to $w_1[12]$ (marked by $\overleftarrow{9}$).
(d) In round 3, compute forward to $x_3[11]$ and $z_3[15]$ (marked by $\overrightarrow{9}$).

10. For column 0 over the MC operation in round 0, compute $w_0[1, 2, 3]$ and $z_0[3]$ (marked by $\boxed{10}$) from $z_0[0, 1, 2]$ and $w_0[0]$.
    (a) Compute forward to $k_1[2, 3]$ (marked by $\overrightarrow{10}$).
    (b) Compute backward to $k_0[15]$ (marked by $\overleftarrow{10}$).
11. For column 3 over the MC operation in round 1, compute $w_1[13, 15]$ and $z_1[13, 15]$ (marked by $\boxed{11}$) from $z_1[12, 14]$ and $w_1[12, 14]$.
    (a) Compute backward to $x_1[1, 11]$ and $k_1[1]$ (marked by $\overleftarrow{11}$).
    (b) Compute forward to $k_2[13, 15]$ (marked by $\overrightarrow{11}$).
    (c) According to the key relations, we have $k_2[15] = k_0[15](\overleftarrow{10}) \oplus k_2[11](2)$ and $\mathsf{SB}(k_2[13]) \oplus k_1[1] = k_3[1](6)$, which are two conflicts of Type III with a total probability of 2^{-16}.
12. Deduce $k_0[13]$, $k_1[10, 11]$, $k_3[2, 3, 6, 7]$ (marked by $\boxed{12}$) by the key relations.
    (a) In round 0, compute forward to $z_0[9]$ (marked by $\overrightarrow{12}$).
    (b) In round 1, compute backward to $w_0[10, 11]$ (marked by $\overleftarrow{12}$).
    (c) In round 3, compute forward to $z_3[7, 10, 11, 14]$ (marked by $\overrightarrow{12}$).
13. For column 2 over the MC in round 0, compute $w_0[8, 9]$ (marked by $\boxed{13}$) from $z_0[8, 9, 10, 11]$ and $w_0[10, 11]$. Since six values are known in the inputs/outputs over the MC operation, there are two Type III conflicts with a total probability of 2^{-16}. Compute forward to $k_1[8, 9]$ (marked by $\overrightarrow{13}$).
14. According to the key relations, deduce $k_3[8, 9, 12, 13]$ (marked by $\boxed{14}$). Compute forward to $z_3[5, 8, 9, 12]$ (marked by $\overrightarrow{14}$) and deduce columns 1,2,3 of $w_3 = \mathsf{MC}(z_3)$ (marked by $\overrightarrow{14}$). So we deduce all states of the starting point.

Degree of freedom and complexity.

- In step 1, we deduce the values for active bytes from the input/output differences in the inbound phase. There are 30 active Sboxes, including $s_1 = 18$ Sboxes with probability 2^{-7} and $s_2 = 12$ Sboxes with probability 2^{-6}. Therefore, there are $2^{18+24}/2 = 2^{41}$ combinations for the 30 active bytes, *i.e.*, there are 2^{41} choices for the bytes marked by $\boxed{1}$ in Figure 13.
- Given one out of 2^{41} choices marked by $\boxed{1}$, seven bytes $k_0[1, 7, 14], k_1[12], k_2[0, 4, 8]$ (marked by a wavy line) are guessed in step 2/5/7. In step 8/11/13, there are $c_3 = 5$ conflicts with a total probability of 2^{-40} marked by underline. There expect $2^{41+56-40} = 2^{57}$ starting points satisfying the inbound path.
- The time of the GD to find one starting point is about $\mathcal{T}'_{\mathsf{GD}} = 2^{40}$. Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-30}$, we have to collect 2^{30} starting points to expect one collision and the degree of freedom is enough. The classical time complexity of the full key collision attack is about $\mathcal{T} = 2^{40+30} = 2^{70}$ and the time complexity is larger than the birthday bound 2^{64}.

1.	$k_0[3,11] = (x_0 \oplus P)[3,11]$	$w_2 = \mathsf{MC} \circ \mathsf{SR}(y_2)$
	$k_3[0,5,10,15] = (x_3 \oplus w_2)[0,5,10,15]$	$w_3[0,1,2,3] = \mathsf{MC}(y_3[0,5,10,15])$
2.	$k_2[3] = k_0[3] \oplus \mathsf{SB}(k_1[12])$	$k_2[7] = k_0[7] \oplus k_2[3]$
	$k_2[11] = k_0[11] \oplus k_2[7]$	$k_0[4] = k_2[4] \oplus k_2[0]$
	$k_0[8] = k_2[8] \oplus k_2[4]$	$w_1[0,4,8] = x_2[0,4,8] \oplus k_2[0,4,8]$
	$w_1[3,7,11] = x_2[3,7,11] \oplus k_2[3,7,11]$	
3.	$w_1[1,2], z_1[0,2] = \mathsf{MC}(z_1[1,3], w_1[0,3])$	$k_2[1,2] = x_2[1,2] \oplus w_1[1,2]$
	$w_1[5,6], z_1[5,7] = \mathsf{MC}(z_1[4,6], w_1[4,7])$	$k_2[5,6] = x_2[5,6] \oplus w_1[5,6]$
	$w_1[9,10], z_1[8,10] = \mathsf{MC}(z_1[9,11], w_1[8,11])$	$k_2[9,10] = x_2[9,10] \oplus w_1[9,10]$
4.	$k_0[5] = k_2[5] \oplus k_2[1]$	$k_0[6] = k_2[6] \oplus k_2[2]$
	$k_0[9] = k_2[9] \oplus k_2[5]$	$k_0[10] = k_2[10] \oplus k_2[6]$
5.	$k_2[14] = k_0[14] \oplus k_2[10]$	$z_0[6] = \mathsf{SB}(k_0[14] \oplus P[14])$
	$w_0[4,5,6,7] = \mathsf{MC}(z_0[4,5,6,7])$	$k_1[4,5,6,7] = x_1[4,5,6,7] \oplus w_0[4,5,6,7]$
6.	$k_3[1] = k_3[5] \oplus k_1[5]$	$k_3[4] = k_1[4] \oplus k_3[0]$
7.	$k_1[14] = \mathsf{SB}^{-1}(k_2[1] \oplus k_0[1])$	$k_3[14] = k_1[14] \oplus k_3[10]$
	$z_0[13] = \mathsf{SB}(k_0[1] \oplus P[1])$	
8.	$w_0[13,15], z_0[12] = \mathsf{MC}(z_0[13,14,15], w_0[12,14])$?	$k_1[13,15] = w_0[13,15] \oplus x_1[13,15]$
	$k_0[12] = P[12] \oplus \mathsf{SB}^{-1}(z_0[12])$	
9.	$k_0[0] = k_2[0] \oplus \mathsf{SB}(k_1[13]) \oplus const$	$k_0[2] = k_2[2] \oplus \mathsf{SB}(k_1[15])$
	$k_2[12] = k_0[12] \oplus k_2[8]$	$k_1[0] = k_3[0] \oplus \mathsf{SB}(k_2[12])$
	$k_3[11] = k_3[15] \oplus k_1[15]$	
10.	$w_0[1,2,3], z_0[3] = \mathsf{MC}(z_0[0,1,2], w_0[0])$	$k_1[2,3] = w_0[2,3] \oplus x_1[2,3]$
	$k_0[15] = P[15] \oplus \mathsf{SB}^{-1}(z_0[3])$	
11.	$w_1[13,15], z_1[13,15] = \mathsf{MC}(z_1[12,14], w_1[12,14])$	$k_1[1] = w_0[1] \oplus \mathsf{SB}^{-1}(z_1[13])$
	$k_2[13,15] = w_1[13,15] \oplus x_2[13,15]$	$k_2[15] \stackrel{?}{=} k_0[15] \oplus k_2[11]$
	$\mathsf{SB}(k_2[13]) \oplus k_1[1] \stackrel{?}{=} k_3[1]$	
12.	$k_0[13] = k_2[13] \oplus k_2[9]$	$k_3[2,3] = k_1[2,3] \oplus \mathsf{SB}(k_2[14,15])$
	$k_3[6,7] = k_1[6,7] \oplus k_3[2,3]$	$k_1[10,11] = k_3[6,7] \oplus k_3[10,11]$
13.	$w_0[8,9] = \mathsf{MC}(z_0[8,9,10,11], w_0[10,11])$?	$k_1[8,9] = w_0[8,9] \oplus x_1[8,9]$
14.	$k_3[8,9] = k_3[4,5] \oplus k_1[8,9]$	$k_3[12,13] = k_3[8,9] \oplus k_1[12,13]$

Table 5: Equations in the guess-and-determine steps for 7-round AES-256. The blue bytes are guessed. The red equations are conflicts.

Quantum attack on 7-round `AES-256`. Although a classical attack is invalid, we can give a valid quantum one. We select 2^{14} choices of bytes marked by [1] and traverse 2^{56} possible values of $k_0[1,7,14], k_1[12], k_2[0,4,8]$.

1. Deduce the pairs (m_i^0, m_i^1) $(i = 0, 1, ..., 29)$ for 30 active Sboxes by accessing the `DDT`, and store them in a qRAM L, whose size is about 60 bytes.
2. Given $|l_0, l_1, \cdots, l_{13}\rangle$ and $l_i \in \{0,1\}$, O_L is a quantum oracle that computes

$$O_L(|l_0, l_1, \cdots, l_{13}\rangle |0\rangle) = |l_0, l_1, \cdots, l_{13}\rangle |m_0^{l_0}, m_1^{l_1}, \cdots, m_{13}^{l_{13}}, m_{14}^0, \cdots, m_{29}^0\rangle \tag{8}$$

3. Define $F : \mathbb{F}_2^{14+56} \mapsto \mathbb{F}_2$ and its quantum oracle,

$$\begin{aligned} U_F : &|l_0, \cdots, l_{13}, k_0[1,7,14], k_1[12], k_2[0,4,8]\rangle |y\rangle \\ &\mapsto y \oplus F(l_0, \cdots, l_{13}, k_0[1,7,14], k_1[12], k_2[0,4,8]), \end{aligned} \tag{9}$$

Implementation of U_F:
(a) Access O_L to get $|m_0^{l_0}, m_1^{l_1}, \cdots, m_{13}^{l_{13}}, m_{14}^0, \cdots, m_{29}^0\rangle$.
(b) Fix the 30 bytes marked by [1] as $(m_0^{l_0}, m_1^{l_1}, \cdots, m_{13}^{l_{13}}, m_{14}^0, \cdots, m_{29}^0)$.
(c) Run Step 1-14 (or Table 5) with 7-byte $(k_0[1,7,14], k_1[12], k_2[0,4,8])$.
(d) Check if the 5 conflicts in Table 5 are satisfied with a probability of 2^{-40}. If so, set a 1-bit flag $\texttt{flag}_1$ as $\texttt{flag}_1 := 1$. Else, set $\texttt{flag}_1 := 0$
(e) Check if the outbound phase is satisfied with a probability of 2^{-30}. If so, set a 1-bit flag $\texttt{flag}_2$ as $\texttt{flag}_2 := 1$. Else, set $\texttt{flag}_2 := 0$
(f) Return 1 as the value of F if $\texttt{flag}_1 = \texttt{flag}_2 = 1$. Return 0 otherwise.
(g) Uncompute steps (a)-(e).

4. Run Grover's algorithm [23] on U_F to find the collision.

Quantum Complexity. Given a choice of bytes marked by [1] and a guess for the 7-byte $(k_0[1,7,14], k_1[12], k_2[0,4,8])$ and taking the uncomputation into account, the cost of U_F is about four 7-round AES-256. The probability of finding the collision is roughly $2^{-40-30} = 2^{-70}$. Therefore, the quantum time complexity is about

$$\frac{\pi}{4}\sqrt{2^{70}} \cdot 4 \approx 2^{36.7} \text{ 7-round AES-256.}$$

6 Key Collision Attacks on Reduced AES-192

We also give a practical key collision attack on 5-round AES-192 and a quantum key collision attack on 6-round AES-192 in the Supplementary Material C. Some practical key collisions on 5-round AES-192 are listed in Table 6.

7 Semi-Free-Start Collisions on Reduced AES-DM

The DM mode is $h_i = \mathsf{AES}_{m_i}(h_{i-1}) \oplus h_{i-1}$ shown in Figure 21 in Supplementary Material D, where the message block m_i acts as the key of the block cipher. The semi-free-start collision is to find two message blocks (m_i, m_i'), such that $h_i = \mathsf{AES}_{m_i}(h_{i-1}) \oplus h_{i-1} = h_i' = \mathsf{AES}_{m_i'}(h_{i-1}) \oplus h_{i-1}$. This is equivalent to $\mathsf{AES}_{m_i}(h_{i-1}) = \mathsf{AES}_{m_i'}(h_{i-1})$, *i.e.*, the free-target-plaintext key collision in Figure 2. At ASIACRYPT 2024, Taiyama *et al.* [43] introduced the semi-free-start collision attacks on 5-round AES-128-DM and 7-round AES-192-DM with time complexities of 2^{57} and 2^{62}, respectively. Based on our guess-and-determine rebound attack, we give improved attacks on 5-round AES-128-DM and 7-round AES-192-DM. In particular, the complexity of the 7-round attack on AES-192-DM is now 2^{20}, which has been practically implemented and some practical collisions are listed in Table 7. All these results are given in the Supplementary Material D.

8 Discussion and Conclusion

Discussion. This paper combines the guess-and-determine approach [5] with the rebound attack [34] to propose a novel framework to build collision attacks. The GD approach [5] itself cannot build a collision attack on AES. Note that in [5, Section 3.2], the authors comment on their GD approach:

> "*The main limitation of this approach is that it completely fails to take into account the differential properties of the S-box ... Therefore, this approach alone does not bring useful result when more than one plaintext is available. However, it can be used as a sub-component in a more complex technique.*"

The authors suggest their GD approach as a sub-component of a more complex technique when handling differentials. In our paper, we embed their GD approach into the rebound attack, called GD rebound, allowing the two tools to work together efficiently. Our GD rebound immediately and significantly improves Taiyama *et al.*'s key collision attack [43], demonstrating the power of combining these two cryptanalysis tools.

Conclusion. In this paper, we improve Dong *et al.*'s triangulating rebound attack by proposing the *guess-and-determine rebound* attack. Based on the new method, we significantly improve Taiyama *et al.*'s key collision attacks on AES and semi-free-start collision attacks on AES-DM. Most of our attacks are practical and the example collision pairs are given, including the 2-/3-round key collision attacks on AES-128, 5-round key collision attack and 7-round semi-free-start collision attack on AES-192, and 6-round key collision attack on AES-256. Besides, some quantum key collision attacks are proposed.

Acknowledgements. We thank the anonymous reviewers from CRYPTO 2025 for their insightful comments. This work is supported by the National Key R&D Program of China (2024YFA1013000), the Natural Science Foundation of China (62272257, 62302250), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Zhongguancun Laboratory.

References

1. Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3291–3308. USENIX Association, 2022.
2. Daniel J. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete. *SHARCS 2009 9: 105.*
3. Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir. Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 299–319. Springer, 2010.
4. Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *ASIACRYPT 2019, Proceedings, Part I*, pages 552–583.

5. Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque. Automatic search of attacks on round-reduced AES and applications. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 169–187. Springer, 2011.
6. André Chailloux, María Naya-Plasencia, and André Schrottenloher. An efficient quantum collision search algorithm and implications on symmetric cryptography. In *ASIACRYPT 2017, Proceedings, Part II*, pages 211–240, 2017.
7. Shiyao Chen, Xiaoyang Dong, Jian Guo, and Tianyu Zhang. Chosen-prefix collisions on aes-like hashing. *IACR Trans. Symmetric Cryptol.*, 2024(4):64–96, 2024.
8. Yu Long Chen, Antonio Flórez-Gutiérrez, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Minematsu, Nicky Mouha, Yusuke Naito, Ferdinand Sibleyras, and Yosuke Todo. Key committing security of AEZ and more. *IACR Trans. Symmetric Cryptol.*, 2023(4):452–488, 2023.
9. Joan Daemen and Vincent Rijmen. Understanding two-round differentials in AES. In *SCN 2006, Proceedings*, volume 4116, pages 78–94. Springer.
10. Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.
11. Patrick Derbez, Pierre-Alain Fouque, Takanori Isobe, Mostafizar Rahman, and André Schrottenloher. Key committing attacks against aes-based AEAD schemes. *IACR Trans. Symmetric Cryptol.*, 2024(1):135–157, 2024.
12. Patrick Derbez, Paul Huynh, Virginie Lallemand, María Naya-Plasencia, Léo Perrin, and André Schrottenloher. Cryptanalysis results on Spook - bringing full-round Shadow-512 to the light. In *CRYPTO 2020, Proceedings, Part III*, volume 12172, pages 359–388.
13. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In *CRYPTO 2012, Proceedings*, volume 7417, pages 719–740. Springer.
14. Xiaoyang Dong, Jian Guo, Shun Li, and Phuong Pham. Triangulating rebound attack on aes-like hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 94–124. Springer, 2022.
15. Xiaoyang Dong, Siwei Sun, Danping Shi, Fei Gao, Xiaoyun Wang, and Lei Hu. Quantum collision attacks on AES-like hashing with low quantum random access memories. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Proceedings, Part II*, volume 12492, pages 727–757.
16. Xiaoyang Dong, Zhiyu Zhang, Siwei Sun, Congming Wei, Xiaoyun Wang, and Lei Hu. Automatic classical and quantum rebound attacks on AES-like hashing by exploiting related-key differentials. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 241–271. Springer, 2021.
17. Alexandre Duc, Jian Guo, Thomas Peyrin, and Lei Wei. Unaligned rebound attack: Application to Keccak. In *FSE 2012, Revised Selected Papers*, volume 7549, pages 402–421.
18. Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symmetric Cryptol.*, 2017(1):449–473, 2017.

19. Antonio Flórez-Gutiérrez, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, André Schrottenloher, and Ferdinand Sibleyras. Internal symmetries and linear properties: Full-permutation distinguishers and improved collisions on gimli. *J. Cryptol.*, 34(4):45, 2021.
20. Pierre-Alain Fouque, Jérémy Jean, and Thomas Peyrin. Structural evaluation of AES and chosen-key distinguisher of 9-round AES-128. In *CRYPTO 2013, Proceedings, Part I*, volume 8042, pages 183–203.
21. David Gérault, Pascal Lafourcade, Marine Minier, and Christine Solnon. Computing AES related-key differential characteristics with constraint programming. *Artif. Intell.*, 278, 2020.
22. Henri Gilbert and Thomas Peyrin. Super-Sbox cryptanalysis: Improved attacks for AES-like permutations. In *FSE 2010, Seoul, Korea, February 7-10, 2010*, pages 365–383, 2010.
23. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219, 1996.
24. Akinori Hosoyamada and Yu Sasaki. Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Proceedings, Part II*, volume 12106, pages 249–279.
25. Akinori Hosoyamada and Yu Sasaki. Quantum collision attacks on reduced SHA-256 and SHA-512. In *CRYPTO 2021*, volume 12825, pages 616–646. Springer.
26. Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved rebound attack on the finalist Grøstl. In *FSE 2012, Washington, DC, USA, March 19-21, 2012*, pages 110–126, 2012.
27. Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Multiple limited-birthday distinguishers and applications. In *SAC 2013, Burnaby, BC, Canada, August 14-16, 2013*, pages 533–550, 2013.
28. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO 2016, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 207–237, 2016.
29. Dmitry Khovratovich, Alex Biryukov, and Ivica Nikolic. Speeding up collision search for byte-oriented hash functions. In *CT-RSA 2009, Proceedings*, volume 5473, pages 164–181.
30. Dmitry Khovratovich, Ivica Nikolic, and Christian Rechberger. Rotational rebound attacks on reduced Skein. *J. Cryptol.*, 27(3):452–479, 2014.
31. Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer. Rebound distinguishers: Results on the full Whirlpool compression function. In *ASIACRYPT 2009, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 126–143, 2009.
32. Gregor Leander and Alexander May. Grover Meets Simon - quantumly attacking the FX-construction. In *ASIACRYPT 2017, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, pages 161–178, 2017.
33. Krystian Matusiewicz, María Naya-Plasencia, Ivica Nikolic, Yu Sasaki, and Martin Schläffer. Rebound attack on the full LANE compression function. In *ASIACRYPT 2009, Proceedings*, volume 5912, pages 106–125.
34. Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In *FSE 2009, Leuven, Belgium, February 22-25, 2009*, pages 260–276, 2009.

35. Florian Mendel, Vincent Rijmen, and Martin Schläffer. Collision attack on 5 rounds of Grøstl. In *FSE 2014, London, UK, March 3-5, 2014*, pages 509–521, 2014.
36. Marcel Nageler, Felix Pallua, and Maria Eichlseder. Finding collisions for round-reduced romulus-h. *IACR Trans. Symmetric Cryptol.*, 2023(1):67–88, 2023.
37. Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Committing security of ascon: Cryptanalysis on primitive and proof on mode. *IACR Trans. Symmetric Cryptol.*, 2023(4):420–451, 2023.
38. María Naya-Plasencia. How to improve rebound attacks. In *CRYPTO 2011, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 188–205, 2011.
39. Jianqiang Ni, Yingxin Li, Fukang Liu, and Gaoli Wang. Practical key collision on AES and kiasu-bc. *IACR Cryptol. ePrint Arch.*, page 462, 2025.
40. Yu Sasaki. Meet-in-the-middle preimage attacks on AES hashing modes and an application to Whirlpool. In *FSE 2011, Revised Selected Papers*, pages 378–396.
41. Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta. Non-full-active Super-Sbox analysis: Applications to ECHO and grøstl. In *ASIACRYPT 2010, Singapore, December 5-9, 2010. Proceedings*, pages 38–55, 2010.
42. André Schrottenloher. Quantum linear key-recovery attacks using the QFT. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 258–291. Springer, 2023.
43. Kodai Taiyama, Kosei Sakamoto, Ryoma Ito, Kazuma Taka, and Takanori Isobe. Key collisions on AES and its applications. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part VII*, volume 15490 of *Lecture Notes in Computer Science*, pages 267–300. Springer, 2024.
44. Kodai Taiyama, Kosei Sakamoto, Ryoma Ito, Kazuma Taka, and Takanori Isobe. Key collisions on AES and its applications. *IACR Cryptol. ePrint Arch.*, page 1508, 2024.
45. Ryunouchi Takeuchi, Yosuke Todo, and Tetsu Iwata. Key recovery, universal forgery, and committing attacks against revised rocca: How finalization affects security. *IACR Trans. Symmetric Cryptol.*, 2024(2):85–117, 2024.
46. Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptol.*, 12(1):1–28, 1999.

Supplementary Material

A The Practical Key Collisions on AES

Some examples of the key collisions we found are given in Table 6 and 7.

B Key Collision Attacks and Differential Characteristics on Reduced AES-256

B.1 The Invalid Key Collision on 6-round AES-256 in [43]

In [43], Taiyama *et al.* gave a related-key differential characteristic for key collision attack on 6-round AES-256, which is listed in Figure 14. For $k_1[12]$, the differences $\Delta k_1[12] = \texttt{0x02}$ and $\Delta\mathsf{SB}(k_1[12]) = \texttt{0x48}$ are fixed. There is $\texttt{DDT}(\texttt{0x02}, \texttt{0x48}) = 0$ for the Sbox of AES, but [43] regards that the probability of the active Sbox is 2^{-7}. It is not a valid instantiation of the related-key truncated differential and one can not make a correct attack with this differential.

It seems that a simple way to correct this differential is to modify $\Delta\mathsf{SB}(k_1[12])$ to ensure $\texttt{DDT}(\texttt{0x02}, \Delta\mathsf{SB}(k_1[12])) > 0$ and $\texttt{DDT}(\Delta\mathsf{SB}(k_1[12]), \texttt{0x01}) > 0$. However, even if $\Delta\mathsf{SB}(k_1[12])$ is modified correctly, there is still another problem in this differential that makes the attack fail. We briefly recall the attack in [43], where the 0th and 1st rounds in the EN path is the inbound phase and the remaining parts including the KS are the outbound phase. The attack procedures are as follows.

1. Choose 2^{51} values of $x_0[0-15]$ and $y_0[0-15]$ that satisfy the differences in Δx_0 and Δy_0. Deduce 2^{51} values of $k_0[0-15]$.
2. Choose 2^{11} values of $x_1[12-15]$ and $y_1[12-15]$ that satisfy the differences in $\Delta x_1[12-15]$ and $\Delta y_1[12-15]$. Deduce 2^{51+11} values of $\{k_1[12-15], k_2[0-15]\}$. Since the differences in $\Delta k_1[12-15]$, $\Delta k_2[13]$, $\Delta\mathsf{SB}(k_1[12-15])$ and $\Delta\mathsf{SB}(k_2[13])$ are fixed, there is a filter of 2^{-35}. Then $2^{51+11-35} = 2^{27}$ values will remain.
3. Choose 2^{22} values of $x_1[4-11]$ and $y_1[4-11]$ satisfying the differences in $\Delta x_1[4-11]$ and $\Delta y_1[4-11]$. Deduce 2^{27+22} values of $k_1[4-11]$.
4. Choose 2^{12} values of $x_1[0-3]$ and $y_1[0-3]$ satisfying the differences in $\Delta x_1[0-3]$ and $\Delta y_1[0-3]$. Deduce 2^{49+12} starting points. Since the probability of the remaining parts is 2^{-61}, one collision is excepted.

The probability of the inbound phase is 2^{-118}. Since the probability of generating k_2 and k_3 in the KS is 2^{-35}, the probability of the outbound phase should be 2^{-96}. So the total probability is $2^{-118-94} = 2^{-214}$, not 2^{-179} in [43]. As above steps, the freedom seems enough to find one collision. But in fact, for some specific byte, the freedom is not enough. For $x_2[2]$, there is

$$\begin{aligned} x_2[2] &= w_1[2] \oplus k_2[2] \\ &= y_1[0] \oplus y_1[5] \oplus 02 \cdot y_1[10] \oplus 03 \cdot y_1[15] \oplus x_0[2] \oplus P[2] \oplus \mathsf{SB}(k_1[15]). \end{aligned} \tag{10}$$

Key Collisions on 2-round AES-128 **for** $P = 0$
K_1 : 0x377008630096ccb134256ba749694717
K_2 : 0xeb700840dc4ad4b1340d738449694717
C : 0xb6446d21185c641fb8919d7a9b317fa7
K_1 : 0x3781e1630096ccd49e256ba791906f40
K_2 : 0xeb81e140dc4ad4d49e0d738491906f40
C : 0x7a6a266776466cb326727890eed2a200
K_1 : 0x37a065630096cc2951256ba7f6d259cc
K_2 : 0xeba06540dc4ad429510d7384f6d259cc
C : 0x255739597d81544a17f77b3df757514c
K_1 : 0x37de86630096cc06b3256ba78dd0a420
K_2 : 0xebde8640dc4ad406b30d73848dd0a420
C : 0x9c5f0ecc32d2a7386a58974ab9065fbf
Key Collisions on 3-round AES-128 **for** $P = 0$
K_1 : 0x0f6eef4eea138a1b60057a26d30bedfa
K_2 : 0xd76ec74dcc138ad460057a26d30bed36
C : 0x87c494f5d33621b65ad032992b8f6def
K_1 : 0x0f06c74eeae0f2d494b699656837a236
K_2 : 0xd706ef4dcce0f21b94b699656837a2fa
C : 0xa10740d59630c5a0e1ac2462fb79349d
K_1 : 0x0f42ef4eea32361b5938c173b43fd7cc
K_2 : 0xd742c74dcc3236d45938c173b43fd700
C : 0x04a426d2376e704c409b8409cb6f02d1
K_1 : 0x0f80ef4eea6e6dd4304baa184553a000
K_2 : 0xd780c74dcc6e6d1b304baa184553a0cc
C : 0xfd982cfd198e8871b66de2da27ccf1df
Key Collisions on 5-round AES-192 **for** $P = 0$
K_1 : 0x44d96d845d5312c8f19c3600814ba03196f3705625a24502
K_2 : 0x6bf638da727c4780deb3475eae64d17996f3704025a24502
C : 0x4b49ed9c3ccc1a9dd3dcaa16f22165ce
K_1 : 0x44f66d845d5312c8f1b3365eae4ba031b1582bad72a214d8
K_2 : 0x6bd938da727c4780de9c47008164d179b1582bbb72a214d8
C : 0x1f5bd0d969371898991de10fdd46affa
K_1 : 0x44f638845d7c12c8f1b3365eae4ba031c13cd61fde0950b1
K_2 : 0x6bd96dda72534780de9c47008164d179c13cd609de0950b1
C : 0x58e3c7352e8f6e1179c3d04945dbb28b
K_1 : 0x44f66dda5d7c4780de9c4700ae4bd17978bb14bc68c2a9cc
K_2 : 0x6bd93884725312c8f1b3365e8164a03178bb14aa68c2a9cc
C : 0x3a68df0626730bb308140896a1d65d3f
Key Collisions on 6-round AES-256 **for** $P = 0$
K_1 : 0xcc642ac6ef0e7385009b145cd43c0606997c122e7ec132621604eedc0013e201
K_2 : 0xe8722dd2ef0e7385009b145cd4850606202ec2477c713660afd23eb50215e603
C : 0x3dea345ea340d0a3e4dd1a7c28d6babc
K_1 : 0xcc642ac6ef12733705870c5cd43c09b4d95601971575edaf2d01f1de0013e201
K_2 : 0xe8722dd2ef12733705870c5cd48509b46004d1fe17c5e9ad94d721b70215e603
C : 0x08a3d5e348d72cb664546861dd2a3476
K_1 : 0xcc642ac6ef1473a60d81055cd43c0325ab26014e42cd0291d70394660013e201
K_2 : 0xe8722dd2ef1473a60d81055cd48503251274d127407d06936ed5440f0215e603
C : 0xdda972f32bd01e98b335abc1c0269ee0
K_1 : 0xcc642ac6ef0a73e80f9f135cd43c0e6b417d1a45f0136ec90e1deffb0013e201
K_2 : 0xe8722dd2ef0a73e80f9f135cd4850e6bf82fca2cf2a36acbb7cb3f920215e603
C : 0xd75867d21905601fa92ea1685904989f

Table 6: Practical fixed-target-plaintext key collisions: $\mathsf{AES}_{K_1}(0) = \mathsf{AES}_{K_2}(0)$

Key Collisions on 7-round AES-192 for free plaintext	
P	: 0x64d66875c60b79e2e68073168f38cd68
K_1	: 0x70496db77bb5888702db85c405b090700753b5f50ff32437
K_2	: 0x70416db77bb57d8702db85c405b090700753b5f50ff3d137
C	: 0x909987f518b5eda72b0fd6912066b853
P	: 0x7be81382a1cfba2772f54963018004ea
K_1	: 0xfce8a3f277b64b1f0445dc5c038f0fea075046de0ef02470
K_2	: 0xfce0a3f277b6be1f0445dc5c038f0fea075046de0ef0d170
C	: 0xd2e325405187512f00966535636a9291
P	: 0xc5f81b710a1a8db5610a8c6181ee87d7
K_1	: 0x91677e1b74ae891a03476659001ff7ac07503b4909f524c6
K_2	: 0x916f7e1b74ae7c1a03476659001ff7ac07503b4909f5d1c6
C	: 0x2d76c7b5197ac86d6c9aa23144391cd6
P	: 0xe950ea4d0f00b6f09a42585cd5842ddc
K_1	: 0xe6589d0a684db227077e04640bfd58db0757dcf105f2243f
K_2	: 0xe6509d0a684d4727077e04640bfd58db0757dcf105f2d13f
C	: 0x7a8f4ed56bf1b669e9e70824d3812246
P	: 0x6acbfae07d2a0cc7231a412c1f57197b
K_1	: 0x10cbea5b7de4fd6b02a429bf041ada560553365a0bf724e0
K_2	: 0x10c3ea5b7de4086b02a429bf041ada560553365a0bf7d1e0
C	: 0xa679e50e98cc9ceb3f87e2fad0c4b011

Table 7: Practical free-target-plaintext key collisions on AES-192. Adding the feedback operation to C, we can get the semi-free start collision on AES-192-DM.

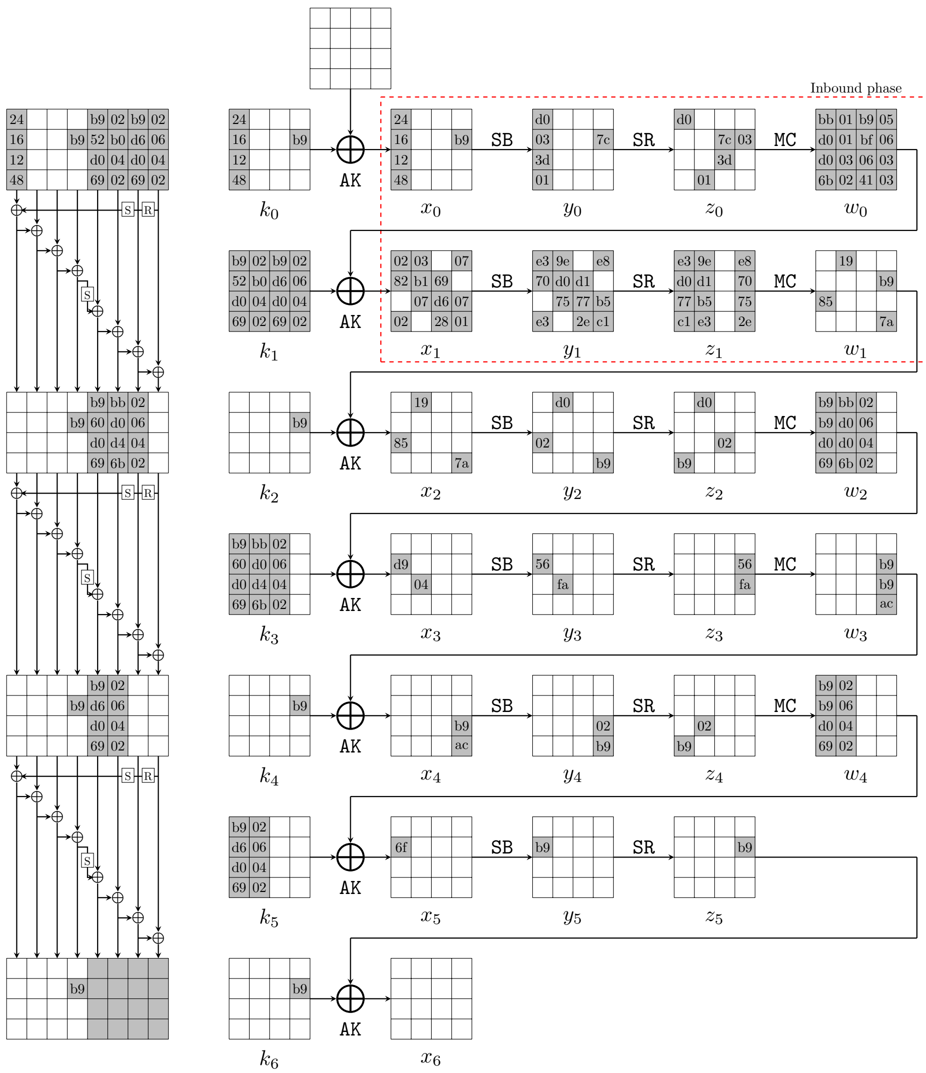

Fig. 14: The related-key differential characteristic on 6-round AES-256 in [43]

The values of $x_0[2]$ and $y_1[0,5,10,15]$ are directly chosen in step 1 to 4, and the value of $k_1[15]$ is filtered in step 2 to satisfy the differential. We list the values of $x_0[2]$, $y_1[0,5,10,15]$, $k_1[15]$ in Table 8, where there are 2^6 combinations in total. In fact, the value of $x_0[2]$ is fixed to one value in step 1. With 2^5 combinations of $x_0[2]$, $y_1[0,5,10,15]$, $k_1[15]$, there are 2^5 possible values of $x_2[2]$ with fixed $P[2]$. The probability of the active Sbox for $x_2[2]$ is 2^{-7}. When $P=0$ and $x_0[2]$ is fixed to `0x87` or `0x95`, all the 2^5 values of $x_2[2]$ don't match the differences unfortunately. So in step 4, after the filter there is no collision found.

State	$x_0[2]$	$y_1[0]$	$y_1[5]$	$y_1[10]$	$y_1[15]$	$k_1[15]$
Values	{0x87, 0x95}	{0x10, 0xf3}	{0xc3, 0x13}	{0x59, 0x2e}	{0xcd, 0x0c}	{0x5c, 0x5e}

Table 8: The possible values of states in Equation 10

B.2 Differential Characteristics on Reduced AES-256

We give the related-key differentials used in the key collision attacks on 6-round and 7-round AES-256 in Figure 15 and Figure 16.

C Key Collision Attacks on Reduced AES-192

In this section, we give a practical key collision attack on 5-round AES-192 applying the differential characteristic in [43]. We also give the first quantum key collision attack on 6-round AES-192.

C.1 The Practical Attack on 5-round AES-192

We reuse the differential characteristic for AES-192 with a probability of 2^{-186} in [43], which is shown in Figure 17. Our inbound phase covers the first three rounds of the EN and KS, which has 24 active Sboxes with a probability of 2^{-165}, including 1 active Sbox in the key schedule. The probability of the outbound phase is $2^{-p_{out}} = 2^{-21}$. The guess-and-determine steps of the GD are listed below, also in Figure 18. The detailed equations are listed in Table 9.

Guess-and-determine procedure of the inbound phase.

1. Since all 16 bytes Δx_0 and Δy_0 are known, we can deduce the full state of x_0 and y_0 (marked by $\boxed{1}$ in Figure 18) by accessing the DDT. With fixed P, we can deduce the whole state of k_0 (marked by $\overleftarrow{1}$) from x_0. Compute forward to $w_0 = \texttt{MC} \circ \texttt{SR}(y_0)$, marked by $\overrightarrow{1}$.

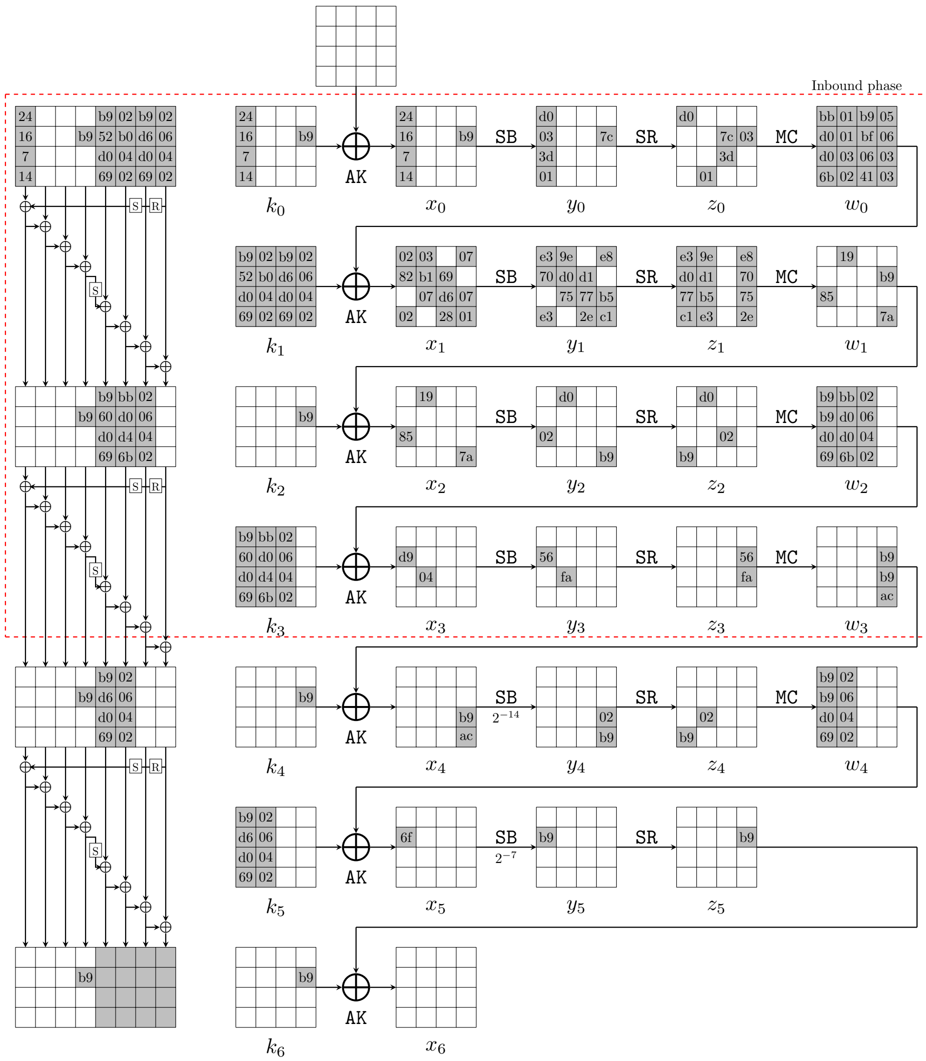

Fig. 15: The new related-key differential characteristic on 6-round AES-256

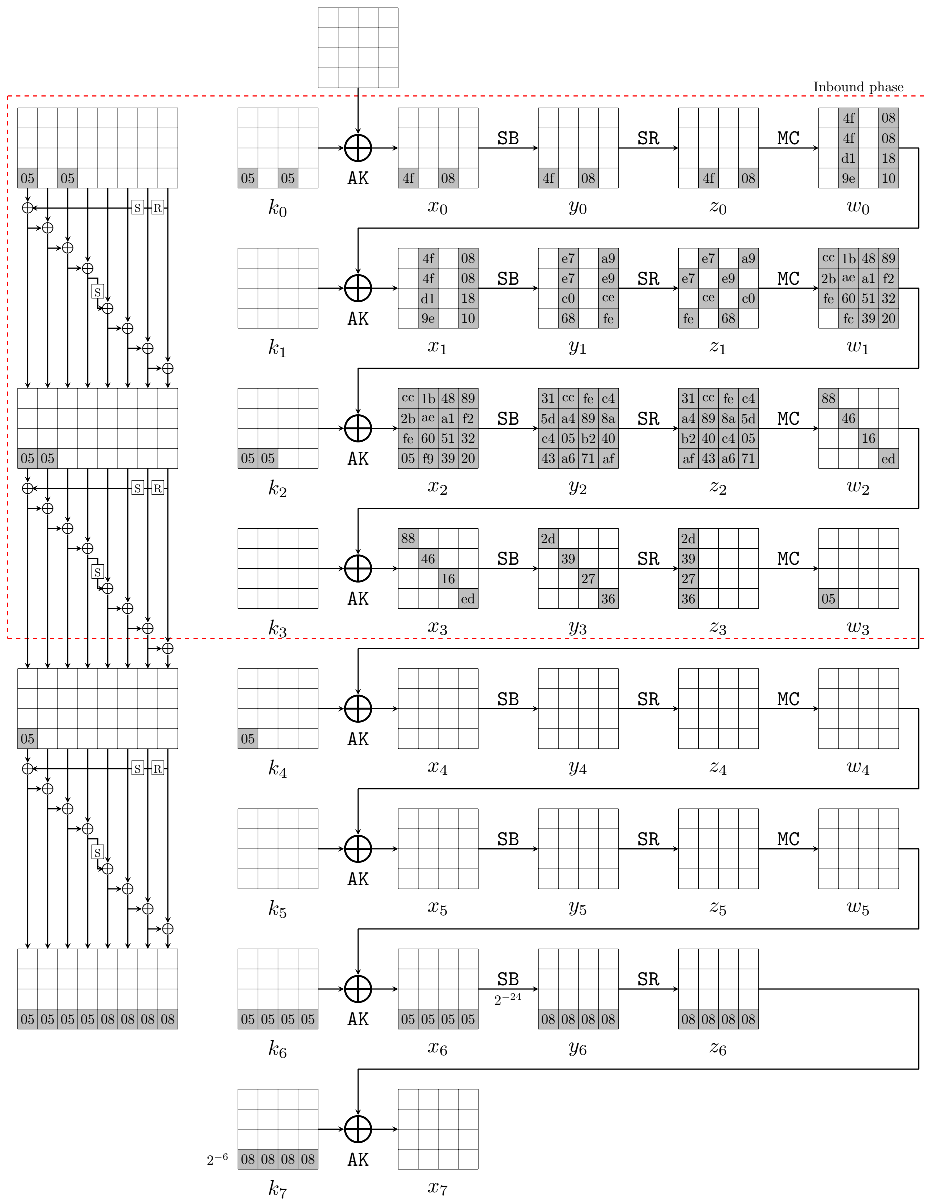

Fig. 16: The related-key differential trail on 7-round AES-256

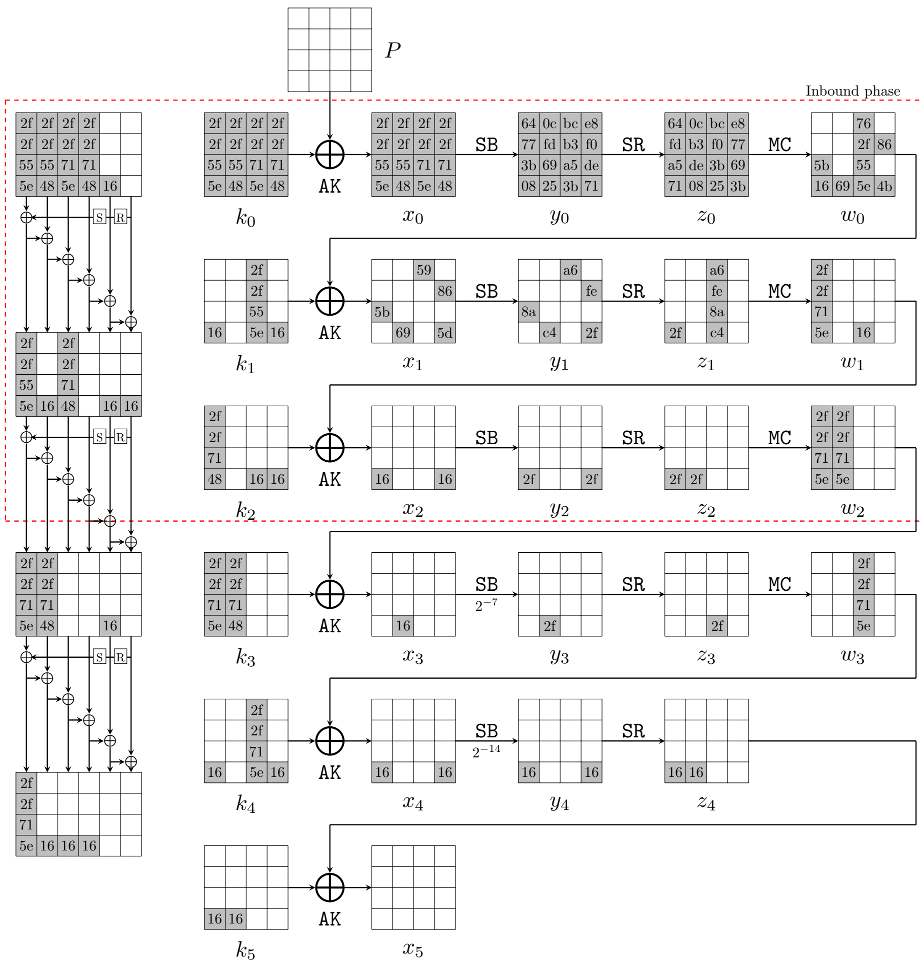

Fig. 17: The related-key differential characteristic on 5-round AES-192 in [43]

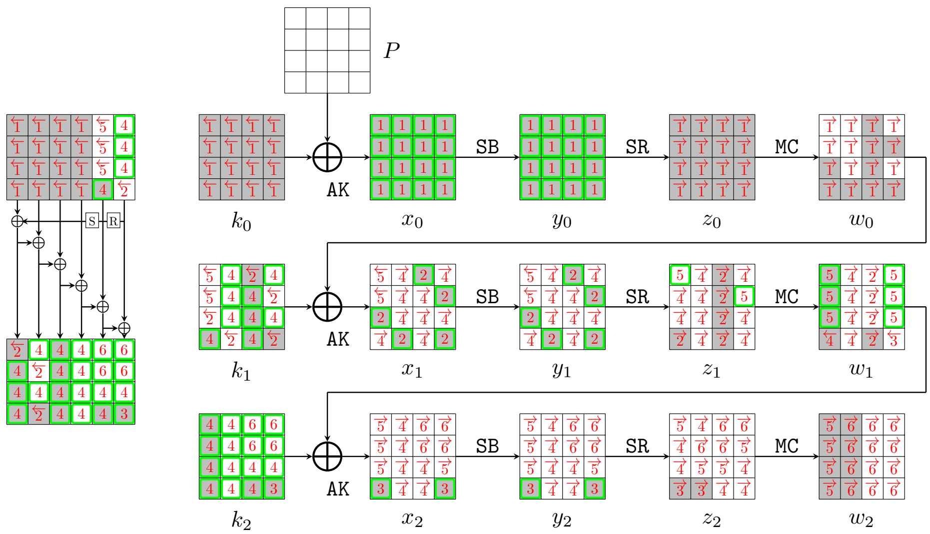

Fig. 18: Steps of the GD in the inbound phase for 5-round AES-192

2. Similarly, deduce $x_1[2, 7, 8, 13, 15]$ and $y_1[2, 7, 8, 13, 15]$ (marked by $\boxed{2}$) by accessing the DDT. Then compute $k_1[2, 7, 8, 13, 15] = (x_1 \oplus w_0)[2, 7, 8, 13, 15]$ (marked by $\overleftarrow{2}$). We can also deduce $z_1[3, 8, 9, 10, 11]$ (marked by $\overrightarrow{2}$) from $y_1[2, 7, 8, 13, 15]$, and compute $w_1[8, 9, 10, 11] = \mathtt{MC}(z_1[8, 9, 10, 11])$ (marked by $\overrightarrow{2}$).
3. Deduce $x_2[3, 15]$ and $y_2[3, 15]$ (marked by $\boxed{3}$) by accessing the DDT. Since $\Delta k_2[15]$ and $\Delta \mathsf{SB}(k_2[15])$ are known (see Figure 17), deduce $k_2[15]$ (marked by 3) by accessing the DDT. Compute backward to get $w_1[15] = k_2[15] \oplus x_2[15]$ (marked by $\overleftarrow{3}$).
4. According to the key relations, we can deduce $k_1[3, 4, 5, 6, 9, 10, 11, 12, 14]$ and $k_2[0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 14]$ (marked by $\boxed{4}$). The equations are given in Table 9.
   (a) Compute forward in round 1, we can deduce $z_1[1, 2, 4, 5, 6, 7, 12, 14, 15]$ and compute $w_1[4, 5, 6, 7] = \mathsf{MC}(z_1[4, 5, 6, 7])$ (marked by $\overrightarrow{4}$).
   (b) Compute backward in round 2 to get $w_1[3] = k_2[3] \oplus x_2[3]$ (marked by $\overleftarrow{4}$). Compute forward to deduce $z_2[1, 2, 4, 11, 14, 15]$ (marked by $\overrightarrow{4}$).
5. For column 0 and column 3 over the MC operation in round 1, four values in the inputs and outputs are known; we can deduce the other four values.
   (a) For column 0, deduce $z_1[0]$ and $w_1[0, 1, 2]$ (marked by $\boxed{5}$) from $z_1[1, 2, 3]$ and $w_1[3]$.
   (b) For column 3, deduce $z_1[13]$ and $w_1[12, 13, 14]$ (marked by $\boxed{5}$) from $z_1[12, 14, 15]$ and $w_1[15]$.
   (c) Compute backward to compute $k_1[0] = w_0[0] \oplus \mathsf{SB}^{-1}(z_1[0])$ and $k_1[1] = w_0[1] \oplus \mathsf{SB}^{-1}(z_1[13])$ (marked by $\overleftarrow{5}$).
   (d) Compute forward to get $z_2[0, 6, 10, 13]$ and $w_2[0, 1, 2, 3]$(marked by $\overrightarrow{5}$).

6. According to the key relations, we can deduce $k_2[8, 9, 12, 13]$ (marked by $\boxed{6}$). Compute forward to get columns 1, 2, and 3 of w_2 (marked by $\overrightarrow{6}$). So we totally determine the starting point.

Degree of freedom and complexity. There are totally 24 active Sboxes in the 3-round inbound phase, including 21 active Sboxes with probability $s_1 = 2^{-7}$ and 3 active Sboxes with probability $s_2 = 2^{-6}$. Therefore, by accessing the DDT, there expect $2^{21+6}/2 = 2^{26}$ combinations for the 24 active Sboxes, i.e., there are 2^{26} choices for the bytes marked by $\boxed{1}$, $\boxed{2}$, and $\boxed{3}$ in Figure 18. There is no conflict in the GD, i.e., $c_{in} = 0$. The probability of the outbound phase is $2^{-p_{out}} = 2^{-21}$. We have enough degrees of freedom to satisfy the outbound phase. Therefore, the total complexity of the 5-round key-collision attack on AES-192 is about $\mathcal{T} = 2^{21}$. We have practically implemented the attack and find some key pairs (K_1, K_2) in Table 6 such that $\text{AES-192}_{K_1}(0) = \text{AES-192}_{K_2}(0)$, where AES-192 is a 5-round one.

1.	$k_0 = x_0 \oplus P$	$w_0 = \mathsf{MC} \circ \mathsf{SR}(y_0)$
2.	$k_1[2, 7, 8, 13, 15] = (x_1 \oplus w_0)[2, 7, 8, 13, 15]$	$w_1[8, 9, 10, 11] = \mathsf{MC}(z_1[8, 9, 10, 11])$
3.	$w_1[15] = k_2[15] \oplus x_2[15]$	
4.	$k_1[5] = \mathtt{SB}^{-1}(k_1[8] \oplus k_0[0] \oplus const)$	$k_1[10] = k_0[2] \oplus \mathsf{SB}(k_1[7])$
	$k_1[14] = k_0[6] \oplus k_1[10]$	$k_2[2] = k_0[10] \oplus k_1[14]$
	$k_2[6] = k_0[14] \oplus k_2[2]$	$k_2[10] = k_1[2] \oplus k_2[6]$
	$k_1[12] = k_0[4] \oplus k_1[8]$	$k_2[0] = k_0[8] \oplus k_1[12]$
	$k_2[4] = k_0[12] \oplus k_2[0]$	$k_1[9] = k_1[13] \oplus k_0[5]$
	$k_1[6] = \mathsf{SB}^{-1}(k_1[9] \oplus k_0[1])$	$k_1[11] = k_1[15] \oplus k_0[7]$
	$k_1[4] = \mathsf{SB}^{-1}(k_1[11] \oplus k_0[3])$	$k_2[1] = k_0[9] \oplus k_1[13]$
	$k_2[5] = k_0[13] \oplus k_2[1]$	$k_2[3] = k_0[11] \oplus k_1[15]$
	$k_2[7] = k_0[15] \oplus k_2[3]$	$k_2[14] = k_1[6] \oplus k_2[10]$
	$k_2[11] = k_1[7] \oplus k_2[15]$	$k_1[3] = k_2[11] \oplus k_2[7]$
	$w_1[4, 5, 6, 7] = \mathsf{MC}(z_1[4, 5, 6, 7])$	$w_1[3] = k_2[3] \oplus x_2[3]$
5.	$z_1[0], w_1[0, 1, 2] = \mathtt{MC}(z_1[1, 2, 3], w_1[3])$	$k_1[0] = w_0[0] \oplus \mathsf{SB}^{-1}(z_1[0])$
	$z_1[13], w_1[12, 13, 14] = \mathtt{MC}(z_1[12, 14, 15], w_1[15])$	$k_1[1] = w_0[1] \oplus \mathsf{SB}^{-1}(z_1[13])$
6.	$k_2[8] = k_1[0] \oplus k_2[4]$	$k_2[9] = k_1[1] \oplus k_2[5]$
	$k_2[12] = k_1[4] \oplus k_2[8]$	$k_2[13] = k_1[5] \oplus k_2[9]$

Table 9: Equations in the guess-and-determine steps for 5-round AES-192.

C.2 The Quantum Key Collision Attack on 6-round AES-192

We find a new quantum key collision attack on 6-round AES-192. The differential characteristic for 6-round AES-192 is shown in Figure 19, with a probability of 2^{-184}. The inbound phase covers the first three rounds and has 22 active Sboxes, including 1 active Sbox in the key schedule. The probabilities of the inbound phase and outbound phase are 2^{-150} and 2^{-34}, respectively. In the GD of the inbound phase, there are 5 conflicts, which are all of Type III, i.e., $c_{in} = c_3 = 5$

and $c_1 = c_2 = 0$. The guess-and-determine steps of the GD in the inbound phase are given below and in Figure 20. The detail equations are given in Table 10.

Guess-and-determine procedure of the inbound phase.

1. With the fixed differences in $\Delta x_0[2,7]$ and $\Delta y_0[2,7]$, we can deduce $x_0[2,7]$ and $y_0[2,7]$ (marked by $\boxed{1}$) by accessing the $\mathtt{DDT}$. Similarly, deduce $x_1[7,8,9,11,15]$, $y_1[7,8,9,11,15]$, $x_2[0-8,10,12-15]$, and $y_2[0-8,10,12-15]$ (marked by $\boxed{1}$). Since the input difference and output difference of $k_1[7]$ are known, deduce $k_1[7]$ (marked by $\boxed{1}$) by accessing the $\mathtt{DDT}$.
   (a) In round 0, compute $k_0[2,7] = x_0[2,7] \oplus P[2,7]$ (marked by $\overleftarrow{1}$). Compute forward to get $z_0[10,11] = y_0[2,7]$, marked by $\overrightarrow{1}$.
   (b) Compute backward in round 1 to get $w_0[7] = k_1[7] \oplus x_1[7]$ (marked by $\overleftarrow{1}$). Compute forward to get $z_1[3,5,8,11,15]$ (marked by $\overrightarrow{1}$).
   (c) Compute forward in round 2 to get $z_2[0-4,6-14]$, marked by $\overrightarrow{1}$. Compute $w_2[0,1,2,3] = \mathsf{MC}(z_2[0,1,2,3])$ and $w_2[8,9,10,11] = \mathsf{MC}(z_2[8,9,10,11])$ (marked by $\overrightarrow{1}$).
2. According to the key relations, we can deduce $k_1[10]$ (marked by $\boxed{2}$). Guess $k_1[8,9]$ (marked by $\boxed{2}$). Compute $w_0[8,9]$ backward (marked by $\overleftarrow{2}$).
3. For column 2 over the MC operation in round 0, four values in the inputs and outputs are known, and we can deduce the other four values. That is, deduce $z_0[8,9]$ and $w_0[10,11]$ (marked by $\boxed{3}$) from $z_0[10,11]$ and $w_0[8,9]$.
   (a) Compute backward to compute $k_0[8,13] = P[8,13] \oplus \mathsf{SB}^{-1}(z_0[8,9])$ (marked by $\overleftarrow{3}$).
   (b) Compute forward to get $k_1[11] = w_0[11] \oplus x_1[11]$ and $x_1[10] = w_0[10] \oplus k_1[10]$ (marked by $\overrightarrow{3}$). Then deduce $z_1[2]$ (marked by $\overrightarrow{3}$).
4. Guess $k_2[0,1]$ (marked by $\boxed{4}$). According to the key relations, we can deduce $k_0[4]$, $k_1[12,15]$, and $k_2[5]$ (marked by $\boxed{4}$).
   (a) Compute $w_0[15]$ and $w_1[0,1,5]$ backward (marked by $\overleftarrow{4}$).
   (b) Compute $z_0[4]$ forward (marked by $\overrightarrow{4}$).
5. For column 0 over the MC operation in round 1, deduce $z_1[0,1]$ and $w_1[2,3]$ (marked by $\boxed{5}$) from $z_1[2,3]$ and $w_1[0,1]$.
   (a) Compute backward to compute $x_1[0,5] = \mathsf{SB}^{-1}(z_1[0,1])$ (marked by $\overleftarrow{5}$).
   (b) Compute forward to get $k_2[2,3] = w_1[2,3] \oplus x_2[2,3]$ (marked by $\overrightarrow{5}$).
6. Guess $k_1[13,14]$ (marked by $\boxed{6}$). According to the key relations, we can deduce $k_0[5,6,9,10,11]$ (marked by $\boxed{6}$). Compute forward to compute $z_0[1,2,5,14,15]$ (marked by $\overrightarrow{6}$).
7. Guess $k_0[12]$ (marked by $\boxed{7}$). According to the key relations, we can deduce $k_2[4]$ (marked by $\boxed{7}$).
   (a) In round 0, compute forward to compute $z_0[12]$ (marked by $\overrightarrow{7}$).
   (b) In round 2, compute backward to get $w_1[4]$ (marked by $\overleftarrow{7}$).
8. For column 3 over the MC operation in round 0, deduce $z_0[13]$ and $w_0[12,13,14]$ (marked by $\boxed{8}$) from $z_0[12,14,15]$ and $w_0[15]$.

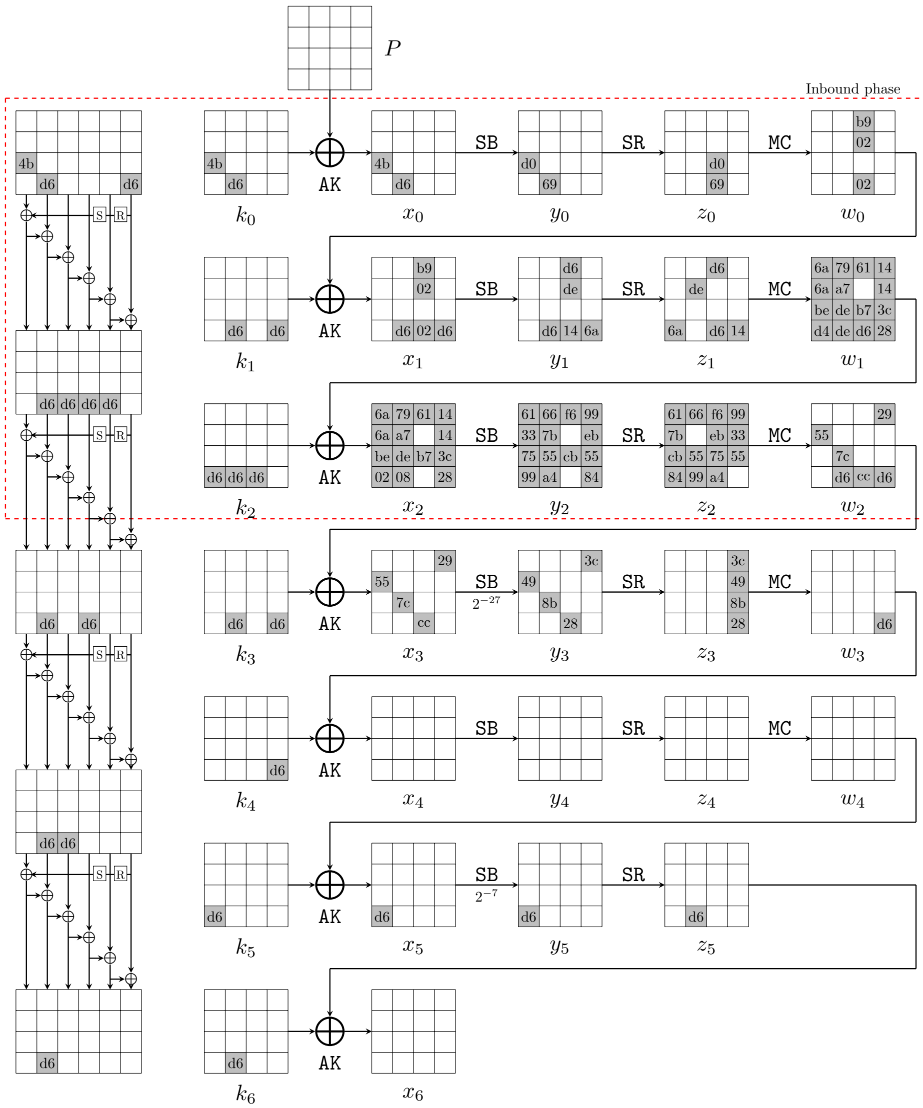

Fig. 19: The related-key differential characteristic on 6-round AES-192

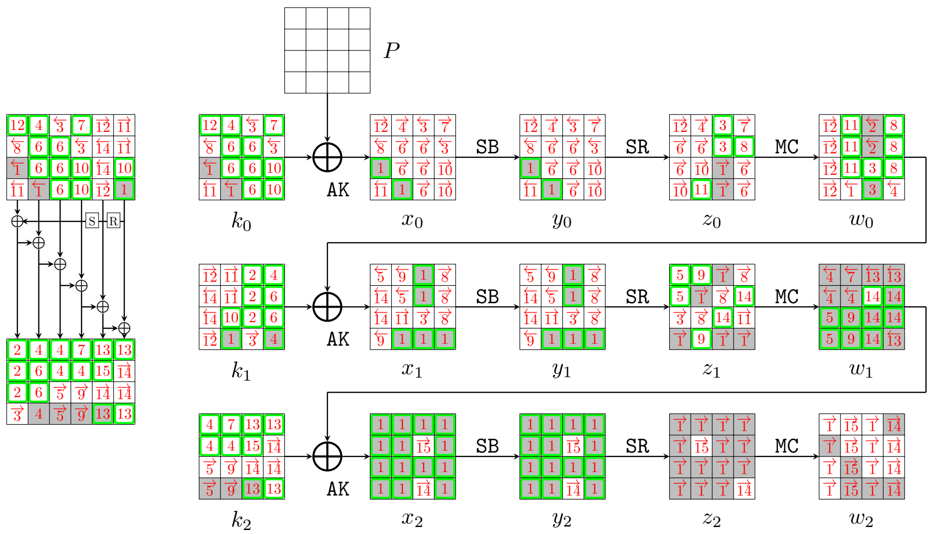

Fig. 20: Steps of the GD in the inbound phase for 6-round AES-192

   (a) Compute backward to compute $x_0[1]$ and $k_0[1]$ (marked by $\overleftarrow{8}$).
   (b) Compute forward to get $x_1[12, 13, 14]$ and $z_1[6, 9, 12]$ (marked by $\overrightarrow{8}$).

9. For column 1 over the MC operation in round 1, deduce $z_1[4, 7]$ and $w_1[6, 7]$ (marked by $\boxed{9}$) from $z_1[5, 6]$ and $w_1[4, 5]$.
   (a) Compute backward to compute $x_1[3, 4]$ (marked by $\overleftarrow{9}$).
   (b) Compute forward to get $k_2[6, 7]$ (marked by $\overrightarrow{9}$).
10. According to the key relations, we can deduce $k_0[14, 15]$ and $k_1[6]$ (marked by $\boxed{10}$). Compute forward to get $z_0[3, 6]$ (marked by $\overrightarrow{10}$).
11. For column 1 over the MC operation in round 0, deduce $z_0[7]$ and $w_0[4, 5, 6]$ (marked by $\boxed{11}$) from $z_0[4, 5, 6]$ and $w_0[7]$.
   (a) Compute backward to compute $x_0[3]$ and $k_0[3]$ (marked by $\overleftarrow{11}$).
   (b) Compute forward to get $k_1[4, 5]$ and $z_1[14]$ (marked by $\overrightarrow{11}$).
   (c) Since $k_0[3] \oplus \mathsf{SB}(k_1[4]) = k_1[11]$ and $k_1[11]$ (marked by $\overrightarrow{3}$) are deduced in Step 3, there is a conflict of Type III with probability of 2^{-8}.
12. According to the key relations, we can deduce $k_0[0]$ (marked by $\boxed{12}$). Compute forward to get $z_0[0]$ (marked by $\overrightarrow{12}$). For column 0 over the MC operation in round 0, deduce $w_0[0, 1, 2, 3]$ (marked by $\overrightarrow{12}$) from $z_0[0, 1, 2, 3]$. Then compute $k_1[0, 3]$ (marked by $\overrightarrow{12}$).
13. According to the key relations, we can deduce $k_2[8, 11, 12, 15]$ (marked by $\boxed{13}$). Compute backward to $w_1[8, 12, 15]$ (marked by $\overleftarrow{13}$).
14. For column 2 over the MC operation in round 1, deduce $z_1[10]$ and $w_1[9, 10, 11]$ (marked by $\boxed{14}$) from $z_1[8, 9, 11]$ and $w_1[8]$. For column 3 over the MC operation in round 1, since five values are known, there is a conflict of Type III

with a probability of 2^{-8}. Then deduce $z_1[13]$ and $w_1[13,14]$ (marked by $\boxed{14}$) from $z_1[12,14,15]$ and $w_1[12,15]$.

(a) Compute backward to compute $x_1[1,2]$ and $k_1[1,2]$ (marked by $\overleftarrow{14}$).

(b) Compute forward to get $k_2[10,13,14]$ and $x_2[11]$ (marked by $\overrightarrow{14}$). Then deduce $z_2[15]$ and $w_2[12,13,14,15]$ (marked by $\overrightarrow{14}$).

(c) Since $k_1[2] \oplus k_2[10] = k_2[6]$ (marked by $\overrightarrow{9}$) and $k_2[14] \oplus k_2[10] = k_1[6]$ (marked by $\boxed{10}$), there are two conflicts of Type III with a probability of 2^{-16}.

15. According to the key relations, we can deduce $k_2[9] = k_1[1] \oplus k_2[5] = k_2[13] \oplus k_1[5]$ (marked by $\boxed{15}$), which is a conflict of Type III with a probability of 2^{-8}. Compute forward to $x_2[9]$ and $w_2[4,5,6,7]$ (marked by $\overrightarrow{15}$).

Degree of Freedom and Complexity.

– In step 1, we deduce the values for active bytes from the input/output differences in the inbound phase. There are 22 active Sboxes, including $s_1 = 18$ Sboxes with probability 2^{-7} and $s_2 = 4$ Sboxes with probability 2^{-6}. Therefore, there are $2^{18+8}/2 = 2^{25}$ combinations for the 22 active bytes, i.e., there are 2^{25} choices for the bytes marked by $\boxed{1}$ in Figure 20.
– Given one out of 2^{25} choices marked by $\boxed{1}$, seven bytes $k_1[8,9], k_2[0,1], k_1[13,14], k_0[12]$ (marked by a wavy line) are guessed in step 2, 4, 6 and 7. In step 11, 14 and 15, there are 5 conflicts with a total probability of 2^{-40} marked by underline. Therefore, there expect $2^{25+56-40} = 2^{41}$ starting points satisfying the inbound differential.
– The time of the GD to find one starting point is about $\mathcal{T}'_{\mathsf{GD}} = 2^{40}$. Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-34}$, we have to collect 2^{34} starting points to expect one collision and the degree of freedom is enough. The classical time complexity of the full key collision attack is about $\mathcal{T} = 2^{40+34} = 2^{74}$ and the time complexity is larger than the birthday bound 2^{64}.

Quantum Attack on 6-round `AES-192`. Although a classical attack is invalid, we can give a valid quantum one. We select 2^{18} choices of bytes marked by $\boxed{1}$ and traverse 2^{56} possible values of $k_0[12], k_1[8,9,13,14], k_2[0,1]$.

1. Deduce the pairs (m_i^0, m_i^1) $(i = 0, 1, ..., 21)$ for 22 active Sboxes by accessing the `DDT`, and store them in a qRAM L, whose size is about 44 bytes.
2. Given $|l_0, l_1, \cdots, l_{17}\rangle$ and $l_i \in \{0,1\}$, O_L is a quantum oracle that computes

$$O_L(|l_0, l_1, \cdots, l_{17}\rangle |0\rangle) = |l_0, l_1, \cdots, l_{17}\rangle |m_0^{l_0}, m_1^{l_1}, \cdots, m_{17}^{l_{17}}, m_{18}^0, m_{19}^0, m_{20}^0, m_{21}^0\rangle \tag{11}$$

3. Define $F : \mathbb{F}_2^{18+56} \mapsto \mathbb{F}_2$ and its quantum oracle,

$$\begin{aligned} U_F : &|l_0, \cdots, l_{17}, k_0[12], k_1[8,9,13,14], k_2[0,1]\rangle |y\rangle \\ &\mapsto y \oplus F(l_0, \cdots, l_{17}, k_0[12], k_1[8,9,13,14], k_2[0,1]), \end{aligned} \tag{12}$$

Implementation of U_F:

1.	$k_0[2,7] = x_0[2,7] \oplus P[2,7]$	$w_0[7] = k_1[7] \oplus x_1[7]$
	$w_2[0,1,2,3] = \mathsf{MC}(z_2[0,1,2,3])$	$w_2[8,9,10,11] = \mathsf{MC}(z_2[8,9,10,11])$
2.	$k_1[10] = k_0[2] \oplus \mathsf{SB}(k_1[7])$	$w_0[8,9] = k_1[8,9] \oplus x_1[8,9]$
3.	$z_0[8,9], w_0[10,11] = \mathsf{MC}(z_0[10,11], w_0[8,9])$	$k_0[8,13] = P[8,13] \oplus \mathsf{SB}^{-1}(z_0[8,9])$
	$k_1[11] = w_0[11] \oplus x_1[11]$	$z_1[2] = \mathsf{SB}(w_0[10] \oplus k_1[10])$
4.	$k_1[15] = k_0[7] \oplus k_1[11]$	$k_1[12] = k_0[8] \oplus k_2[0]$
	$k_0[4] = k_1[12] \oplus k_1[8]$	$k_2[5] = k_0[13] \oplus k_2[1]$
5.	$z_1[0,1], w_1[2,3] = \mathsf{MC}(z_1[2,3], w_1[0,1])$	$x_1[0,5] = \mathsf{SB}^{-1}(z_1[0,1])$
	$k_2[2,3] = w_1[2,3] \oplus x_2[2,3]$	
6.	$k_0[11] = k_2[3] \oplus k_1[15]$	$k_0[5] = k_1[13] \oplus k_1[9]$
	$k_0[6] = k_1[14] \oplus k_1[10]$	$k_0[9] = k_2[1] \oplus k_1[13]$
	$k_0[10] = k_2[2] \oplus k_1[14]$	
7.	$k_2[4] = k_0[12] \oplus k_2[0]$	
8.	$z_0[13], w_0[12,13,14] = \mathsf{MC}(z_0[12,14,15], w_0[15])$	$k_0[1] = P[1] \oplus \mathsf{SB}^{-1}(z_0[12])$
9.	$z_1[4,7], w_1[6,7] = \mathsf{MC}(z_1[5,6], w_1[4,5])$	$k_2[6,7] = w_1[6,7] \oplus x_2[6,7])$
10.	$k_0[14] = k_2[6] \oplus k_2[2]$	$k_0[15] = k_2[7] \oplus k_2[3]$
	$k_1[6] = \mathsf{SB}^{-1}(k_1[9] \oplus k_0[1])$	
11.	$z_0[7], w_0[4,5,6] = \mathsf{MC}(z_0[4,5,6], w_1[7])$	$k_0[3] = P[3] \oplus \mathsf{SB}^{-1}(z_0[7])$
	$k_1[4,5] = w_0[4,5] \oplus x_1[4,5]$	$k_0[3] \oplus \mathsf{SB}(k_1[4]) \stackrel{?}{=} k_1[11]$
	$z_1[14] = \mathsf{SB}(k_1[6] \oplus w_0[6])$	
12.	$k_0[0] = k_1[8] \oplus \mathsf{SB}(k_1[5]) \oplus const$	$z_0[0] = \mathsf{SB}(P[0] \oplus k_0[0])$
	$w_0[0,1,2,3] = \mathsf{MC}(z_0[0,1,2,3])$	$k_1[0,3] = w_0[0,3] \oplus x_1[0,3]$
13.	$k_2[8,11] = k_1[0,3] \oplus k_2[4,7]$	$k_2[12,15] = k_1[4,7] \oplus k_2[8,11]$
	$w_1[8,12,15] = k_2[8,12,15] \oplus x_2[8,12,15]$	
14.	$z_1[10], w_1[9,10,11] = \mathsf{MC}(z_1[8,9,11], w_1[8])$	$z_1[13], w_1[13,14] = \mathsf{MC}(z_1[12,14,15], w_1[12,15])$?
	$k_1[1,2] = w_1[1,2] \oplus \mathsf{SB}^{-1}(z_1[10,13])$	$k_2[10,13,14] = w_1[10,13,14] \oplus x_2[10,13,14]$
	$k_1[2] \oplus k_2[10] \stackrel{?}{=} k_2[6]$	$k_2[14] \oplus k_2[10] \stackrel{?}{=} k_1[6]$
	$x_2[11] = w_1[11] \oplus k_2[11]$	$w_2[12,13,14,15] = \mathsf{MC}(z_2[12,13,14,15])$
15.	$k_2[9] = k_1[1] \oplus k_2[5] \stackrel{?}{=} k_2[13] \oplus k_1[5]$	$x_2[9] = w_1[9] \oplus k_2[9]$
	$w_2[4,5,6,7] = \mathsf{MC}(z_2[4,5,6,7])$	

Table 10: Equations in the guess-and-determine steps for 6-round AES-192. The blue bytes are guessed. The red equations are conflicts.

(a) Access O_L to get $|m_0^{l_0}, m_1^{l_1}, \cdots, m_{17}^{l_1}, m_{18}^0, m_{19}^0, m_{20}^0, m_{21}^0\rangle$.
(b) Fix the 22 bytes marked by $\boxed{1}$ as $(m_0^{l_0}, m_1^{l_1}, \cdots, m_{17}^{l_{17}}, m_{18}^0, m_{19}^0, m_{20}^0, m_{21}^0)$.
(c) Run Step 1-15 (or Table 10) with 7-byte $(k_0[12], k_1[8, 9, 13, 14], k_2[0, 1])$.
(d) Check if the 5 conflicts with a probability of 2^{-40} in Table 10 are satisfied. If so, set a 1-bit flag $\texttt{flag}_1$ as $\texttt{flag}_1 := 1$. Else, set $\texttt{flag}_1 := 0$
(e) Check if the outbound phase with a probability of 2^{-34} is satisfied. If so, set a 1-bit flag $\texttt{flag}_2$ as $\texttt{flag}_2 := 1$. Else, set $\texttt{flag}_2 := 0$
(f) Return 1 as the value of F if $\texttt{flag}_1 = \texttt{flag}_2 = 1$. Return 0 otherwise.
(g) Uncompute steps (a)-(e).

4. Run Grover's algorithm [23] on U_F to find the collision.

Quantum Complexity. Given a choice of bytes marked by $\boxed{1}$ and a guess for the 7-byte $(k_0[12], k_1[8, 9, 13, 14], k_2[0, 1])$ and taking the uncomputation into account, the cost of U_F is about four 6-round AES-192. The probability of finding the collision is roughly $2^{-40-34} = 2^{-74}$. Therefore, the quantum time complexity is about

$$\frac{\pi}{4}\sqrt{2^{74}} \cdot 4 \approx 2^{38.7} \text{ 6-round AES-192.}$$

D The Semi-Free-Start Collision Attacks on AES-DM

At ASIACRYPT 2024, Taiyama *et al.* [43] introduced the semi-free-start collision attacks on 5-round AES-128-DM and 7-round AES-192-DM with time complexities of 2^{57} and 2^{62}, respectively. This section improves the collision attack on 5-round AES-128-DM to a complexity of 2^{39}, and improves the collision attack on 7-round AES-192-DM into a practical one with 2^{20} time.

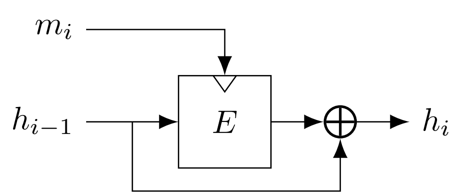

Fig. 21: Davies-Meyer (DM) mode

D.1 The Semi-Free-Start Collision Attack on 5-round AES-128-DM

We give a semi-free-start collision attack on 5-round AES-128-DM with the differential characteristic in [43]. The differential has a probability of 2^{-251} , which is shown in Figure 22. The inbound phase covers the whole KS and 3 rounds of the EN path, *i.e.*, round 1 to round 3. The inbound phase has 32 active Sboxes, including 1 active Sbox in the key schedule. The probabilities of the inbound phase and the outbound phase are 2^{-220} and 2^{-31}, respectively. In the

GD of the inbound phase, there is 1 conflict of Type III, *i.e.*, $c_{in} = c_3 = 1$ and $c_1 = c_2 = 0$. The guess and determination steps of the GD in the inbound phase are listed below, also in Figure 23. The detailed equations are listed in Table 11.

Guess-and-determine procedure of the inbound phase.

1. Deduce the values of $x_1[0,1,3,4,8-15]$, $y_1[0,1,3,4,8-15]$, $x_2[0-4,6-15]$, $y_2[0-4,6-15]$, $x_3[0,5,10,15]$ and $y_3[0,5,10,15]$ with the fixed differences by accessing the DDT, which are all marked by $\boxed{1}$. Since $\Delta k_2[13]$ and $\Delta\mathsf{SB}(k_2[13])$ are known (see Figure 22), deduce $k_2[13]$ (marked by $\boxed{1}$) by accessing the DDT.
   (a) In round 1, compute forward to get $z_1[0,2-9,12,13,15]$ and $w_1[4,5,6,7]$ (marked by $\overrightarrow{1}$).
   (b) In round 2, compute backward to $w_1[13]$ and deduce $k_2[4,6,7] = x_2[4,6,7] \oplus w_1[4,6,7]$ (marked by $\overleftarrow{1}$). Compute forward to get $z_2[0,2-15]$ and $w_2[4-15]$ (marked by $\overrightarrow{1}$).
   (c) In round 3, compute backward to deduce $k_3[5,10,15] = x_3[5,10,15] \oplus w_2[5,10,15]$ (marked by $\overleftarrow{1}$). Compute forward to get $w_3[0,1,2,3]$ (marked by $\overrightarrow{1}$).
2. For column 3 over the MC operation in round 1, compute $w_1[12,14,15]$ and $z_1[14]$ (marked by $\boxed{2}$) from $z_1[12,13,15]$ and $w_1[13]$.
   (a) Compute backward to get $x_1[6]$ (marked by $\overleftarrow{2}$).
   (b) Compute forward to get $k_2[12,14,15]$ (marked by $\overrightarrow{2}$).
3. According to the key relations, deduce $k_3[11,14]$ (marked by $\boxed{3}$). Compute forward to get $z_3[6,15]$ (marked by $\overrightarrow{3}$).
4. Guess $k_2[0]$ (marked by $\boxed{4}$) and deduce $k_1[4]$ and $k_3[0,4]$ (marked by $\boxed{4}$) according to the key relations. Then compute backward to $w_1[0]$ and $w_2[0]$ (marked by $\overleftarrow{4}$). Compute backward to $z_3[4]$ (marked by $\overrightarrow{4}$)
5. For column 0 over the MC operation in round 1, compute $w_1[1,2,3]$ and $z_1[1]$ (marked by $\boxed{5}$) from $z_1[0,2,3]$ and $w_1[0]$. For column 0 over the MC operation in round 2, compute $w_2[1,2,3]$ and $z_2[1]$ (marked by $\boxed{5}$) from $z_2[0,2,3]$ and $w_2[0]$.
   (a) In round 1, compute backward to get $x_1[5]$ (marked by $\overleftarrow{5}$). Compute forward to deduce $k_2[1,2,3]$ (marked by $\overrightarrow{5}$).
   (b) In round 2, compute forward to deduce $k_2[5]$ (marked by $\overleftarrow{5}$).
6. According to the key relations, deduce $k_1[1,2,5,6,7,10,11,14,15]$, $k_2[10,11]$ and $k_3[1,2,3,6,7]$ following the order in Table 11 (marked by $\boxed{6}$). Since the $k_3[1]$ can be computed twice from different key relations, there is a conflict of Type III with a probability of 2^{-8}. Compute backward to get $w_1[10,11]$ (marked by $\overleftarrow{6}$). Compute forward to get $z_3[7,10,11,13,14]$ (marked by $\overrightarrow{6}$).
7. For column 2 over the MC operation in round 1, compute $z_1[10,11]$ and $w_1[8,9]$ (marked by $\boxed{7}$) from $z_1[8,9]$ and $w_1[10,11]$.
   (a) Compute backward to get $x_1[2,7]$ (marked by $\overleftarrow{7}$).
   (b) Compute forward to get $k_2[8,9]$ (marked by $\overrightarrow{7}$).

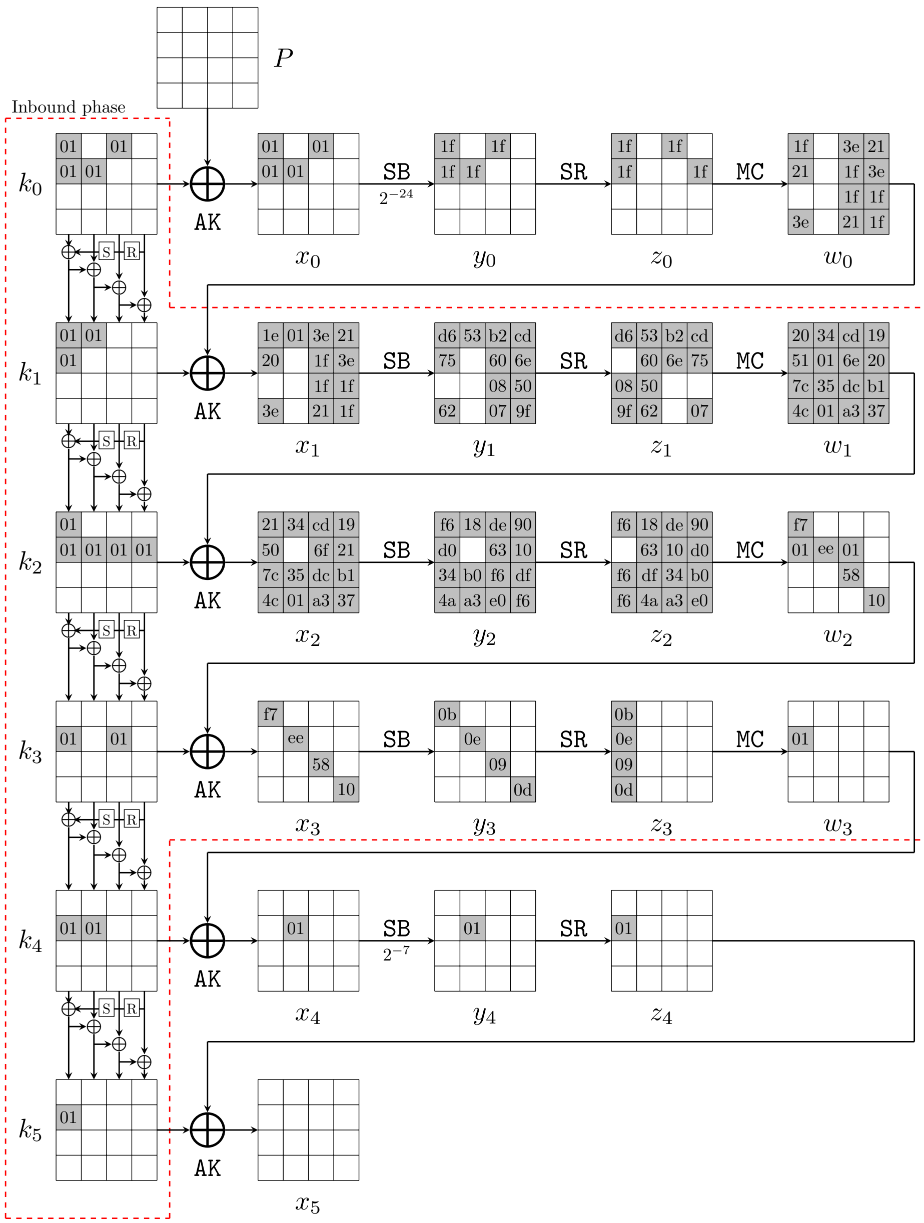

Fig. 22: The related-key differential characteristic on 5-round AES-128 in [43]

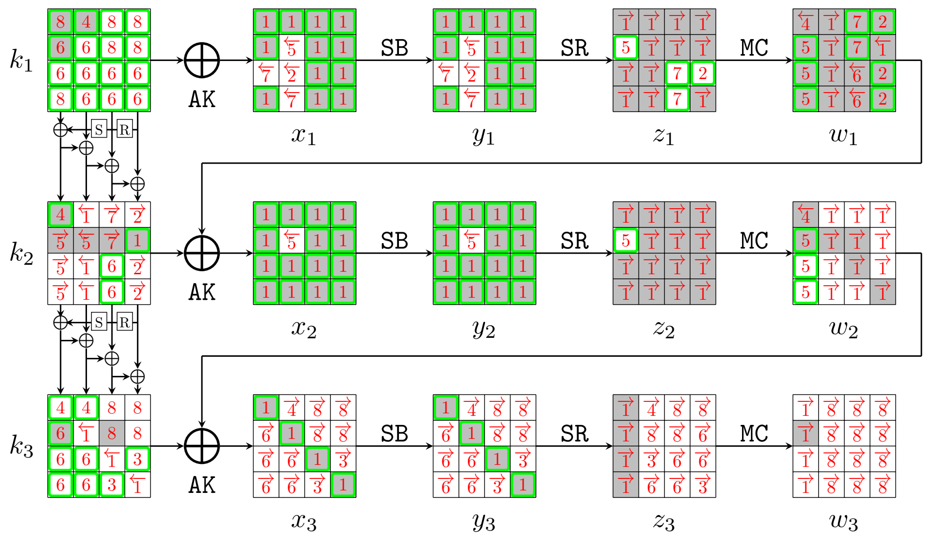

Fig. 23: Steps of the GD in the inbound phase for 5-round AES-128-DM

8. According to the key relations, deduce $k_1[0,3,8,9,12,13]$ and $k_3[8,9,12,13]$ (marked by $\boxed{8}$) following the order in Table 12. Compute forward to deduce the columns 1,2,3 of $w_3 = \mathsf{MC}(z_3)$ (marked by $\overrightarrow{8}$).

Degree of freedom and complexity. There are totally 32 active Sboxes in the inbound phase, including $s_1 = 28$ active Sboxes with probability 2^{-7} and $s_2 = 4$ active Sboxes with probability 2^{-6}. Therefore, by accessing the DDT, there expect $2^{28+8}/2 = 2^{35}$ combinations for the 32 active Sboxes, *i.e.*, there are 2^{35} choices for the bytes marked by $\boxed{1}$. Given one out of 2^{35} choices marked by $\boxed{1}$, 1 byte $k_2[0]$ (marked by a wavy line) is guessed in step 4. And in step 6, there is a conflict with a probability of 2^{-8}. Therefore, there expect $2^{35+8-8} = 2^{35}$ starting points satisfying the inbound differential. The time of the GD to find one starting point is about $\mathcal{T}'_{\mathsf{GD}} = 2^{8}$. Since the probability of the outbound phase is $2^{-p_{out}} = 2^{-31}$, we have to collect 2^{31} starting points to expect one collision and the degree of freedom is enough. The total complexity of the 5-round key-collision attack on AES-128-DM is about $\mathcal{T} = 2^{39}$.

D.2 The Practical Semi-Free-Start Collision Attack on 7-round AES-192-DM

We give a practical semi-free-start collision attack on 7-round AES-192-DM. We reuse the differential characteristic for AES-192 with a probability of 2^{-248} in [43], which is shown in Figure 24. The inbound phase covers the whole KS and 4 rounds of the EN path, *i.e.*, round 1 to round 4. The inbound phase has 33

1.	$k_2[4,6,7] = (x_2 \oplus w_1)[4,6,7]$	$k_3[5,10,15] = (x_3 \oplus w_2)[5,10,15]$
2.	$w_1[12,14,15], z_1[14] = \mathsf{MC}(z_1[12,13,15], w_1[13])$	$k_2[12,14,15] = (w_1 \oplus x_2)[12,14,15]$
3.	$k_3[14] = k_2[14] \oplus k_3[10]$	$k_3[11] = k_2[15] \oplus k_3[15]$
4.	$k_1[4] = k_2[4] \oplus k_2[0]$	$k_3[0] = k_2[0] \oplus \mathsf{SB}(k_2[13]) \oplus const$
	$k_3[4] = k_3[0] \oplus k_2[4]$	
5.	$w_1[1,2,3], z_1[1] = \mathsf{MC}(z_1[0,2,3], w_1[0])$	$k_2[1,2,3] = (w_1 \oplus x_2)[1,2,3]$
	$w_2[1,2,3], z_2[1] = \mathsf{MC}(z_2[0,2,3], w_2[0])$	$k_2[5] = w_1[5] \oplus \mathsf{SB}^{-1}(z_2[1])$
6.	$k_1[5] = k_2[5] \oplus k_2[1]$	$k_1[6] = k_2[6] \oplus k_2[2]$
	$k_1[7] = k_2[7] \oplus k_2[3]$	$k_3[1] = k_2[1] \oplus \mathsf{SB}(k_2[14]) \stackrel{?}{=} k_3[5] \oplus k_2[5]$
	$k_3[2] = k_2[2] \oplus \mathsf{SB}(k_2[15])$	$k_3[3] = k_2[3] \oplus \mathsf{SB}(k_2[12])$
	$k_3[6] = k_2[6] \oplus k_3[2]$	$k_3[7] = k_2[7] \oplus k_3[3]$
	$k_2[10] = k_3[6] \oplus k_3[10]$	$k_2[11] = k_3[7] \oplus k_3[11]$
	$k_1[14] = k_2[10] \oplus k_2[14]$	$k_1[15] = k_2[11] \oplus k_2[15]$
	$k_1[10] = k_2[10] \oplus k_2[6]$	$k_1[11] = k_2[11] \oplus k_2[7]$
	$k_1[1] = k_2[1] \oplus \mathsf{SB}(k_1[14])$	$k_1[2] = k_2[2] \oplus \mathsf{SB}(k_1[15])$
7.	$w_1[8,9], z_1[10,11] = \mathsf{MC}(z_1[8,9], w_1[10,11])$	$k_2[8,9] = (w_1 \oplus x_2)[8,9]$
8.	$k_1[8] = k_2[8] \oplus k_2[4]$	$k_1[9] = k_2[9] \oplus k_2[5]$
	$k_1[12] = k_2[8] \oplus k_2[12]$	$k_1[13] = k_2[9] \oplus k_2[13]$
	$k_1[0] = k_2[0] \oplus \mathsf{SB}(k_1[13]) \oplus const$	$k_1[3] = k_2[3] \oplus \mathsf{SB}(k_1[12])$
	$k_3[8] = k_2[8] \oplus k_3[4]$	$k_3[9] = k_2[9] \oplus k_3[5]$
	$k_3[12] = k_3[8] \oplus k_2[12]$	$k_3[13] = k_3[9] \oplus k_2[13]$

Table 11: Equations in the guess-and-determine steps for 5-round AES-128-DM. The blue bytes are guessed. The red equation is conflict.

active Sboxes, including 1 active Sbox in the key schedule. The probabilities of the inbound phase and outbound phase are 2^{-228} and 2^{-20}, respectively. There is no conflict in the GD of the inbound phase, i.e., $c_{in} = 0$. The guess-and-determine steps of the GD of the inbound phase are listed below, also in Figure 25. The detailed equations are listed in Table 12.

Guess-and-determine procedure of the inbound phase.

1. Deduce the values of $x_1[6,12,13]$, $y_1[6,12,13]$, $x_2[2,6,8{-}11,13{-}15]$, $y_2[2,6,8{-}11,13-15]$, $x_3[0-15]$, $y_3[0-15]$, $x_4[0,5,10,15]$ and $y_4[0,5,10,15]$ with the fixed differences by accessing the DDT, which are all marked by $\boxed{1}$. Since $\Delta k_1[6]$ and $\Delta\mathsf{SB}(k_1[6])$ are known (see Figure 24), deduce $k_1[6]$ (marked by $\boxed{1}$) by accessing the DDT.
   (a) In round 1, compute forward to get $z_1[9,12,14]$ (marked by $\overrightarrow{1}$).
   (b) In round 2, compute forward to get $z_2[2,3,5,6,8-10,14,15]$ (marked by $\overrightarrow{1}$).
   (c) In round 3, compute forward to get the whole state $w_3 = \mathsf{MC} \circ \mathsf{SR}(y_3)$ (marked by $\overrightarrow{1}$).
   (d) In round 4, compute backward to deduce $k_4[0,5,10,15] = x_4[0,5,10,15] \oplus w_3[0,5,10,15]$ (marked by $\overleftarrow{1}$). Compute forward to get the $w_4[0,1,2,3]$ (marked by $\overrightarrow{1}$).

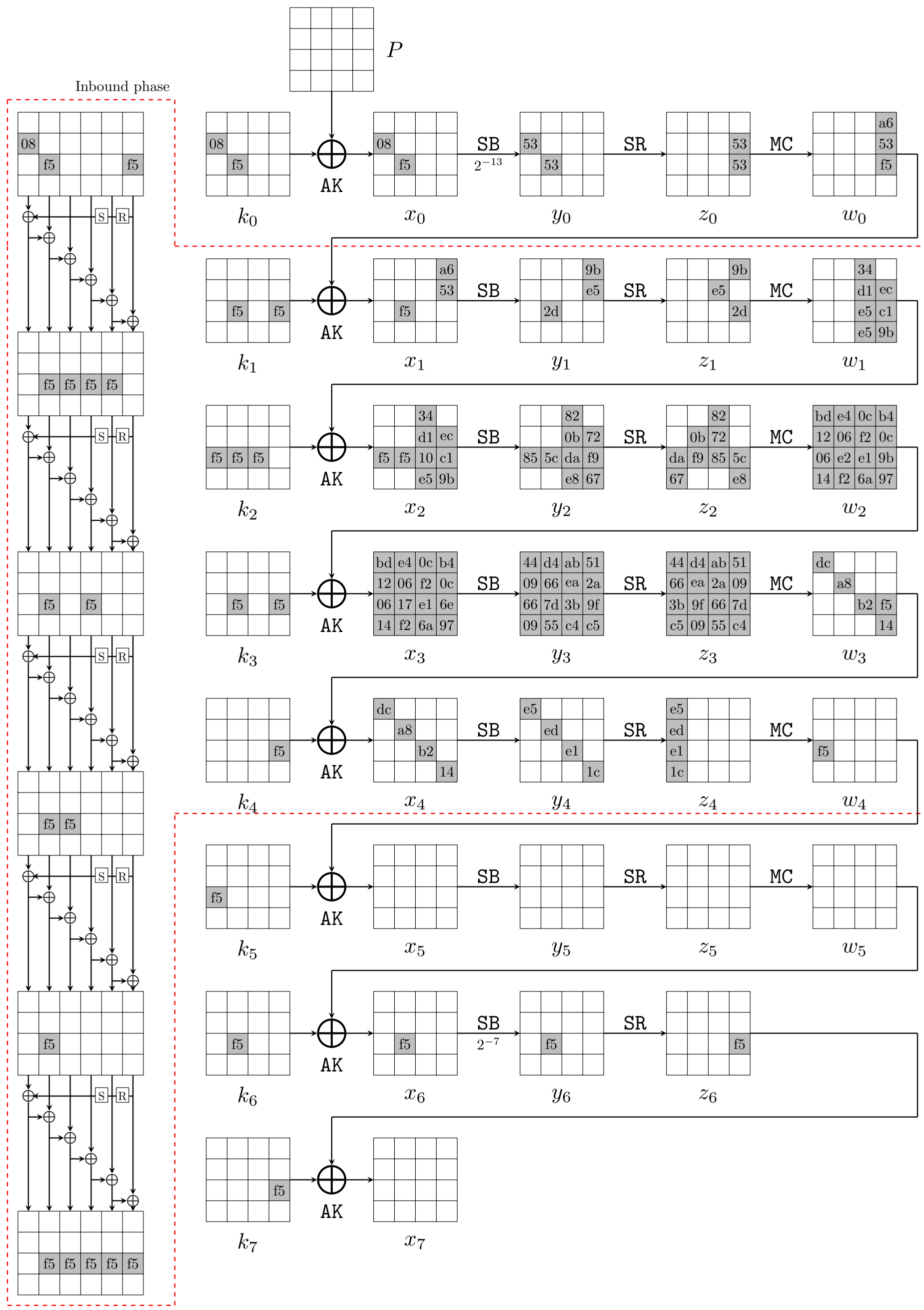

Fig. 24: The related-key differential characteristic on 7-round AES-192 in [43]

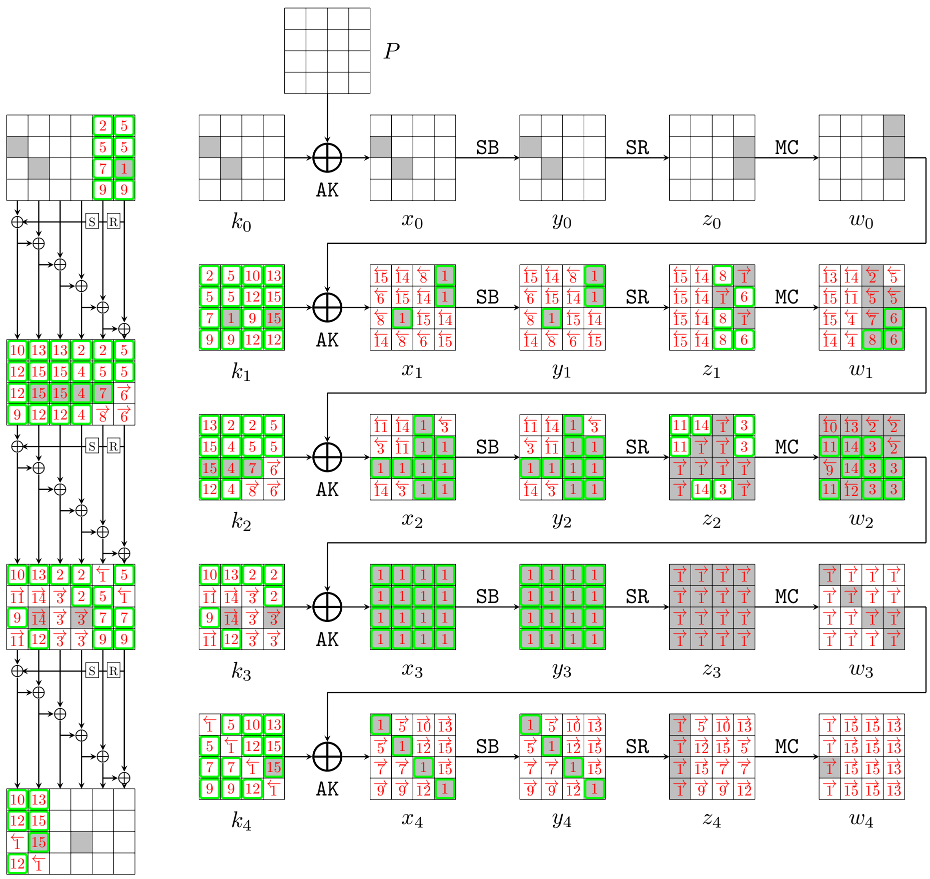

Fig. 25: Steps of the GD in the inbound phase for 7-round AES-192-DM

2. Guess $k_3[8, 12, 13]$ (marked by $\boxed{2}$) and deduce $k_1[0]$ and $k_2[4, 8]$ according to the key relations. Then compute backward to $w_2[8, 12, 13]$ and $w_1[8]$ (marked by $\overleftarrow{2}$).
3. For columns 2,3 over the MC operation in round 2, compute $w_2[9, 10, 11, 14, 15]$ and $z_2[11, 12, 13]$ (marked by $\boxed{3}$) from $z_2[8, 9, 10, 14, 15]$ and $w_2[8, 12, 13]$.
   (a) Compute backward to get $x_2[1, 7, 12]$ (marked by $\overleftarrow{3}$).
   (b) Compute forward to get $k_3[9, 10, 11, 14, 15]$ (marked by $\overrightarrow{3}$).
4. According to the key relations, deduce $k_2[5, 6, 7]$ (marked by $\boxed{4}$). Compute backward and get $w_1[6, 7]$ (marked by $\overleftarrow{4}$).
5. Guess $k_2[12, 13]$ (marked by $\boxed{5}$) and deduce $k_1[1, 4, 5]$, $k_2[9]$ and $k_4[1, 4]$ (marked by $\boxed{5}$) following the order in Table 12. Compute backward to $w_1[9, 12, 13]$ (marked by $\overleftarrow{5}$) and compute forward to $z_4[4, 13]$ (marked by $\overrightarrow{5}$).
6. For column 3 over the MC operation in round 1, compute $w_1[14, 15]$ and $z_1[13, 15]$ (marked by $\boxed{6}$) from $z_1[12, 14]$ and $w_1[12, 13]$.
   (a) Compute backward to get $x_1[1, 11]$ (marked by $\overleftarrow{6}$).
   (b) Compute forward to get $k_2[14, 15]$ (marked by $\overrightarrow{6}$).
7. According to the key relations, deduce $k_1[2]$, $k_2[10]$ and $k_4[2, 6]$ (marked by $\boxed{7}$). Compute backward to get $w_1[10]$ (marked by $\overleftarrow{7}$). Compute forward to get $z_4[10, 14]$ (marked by $\overrightarrow{7}$).
8. For column 2 of the MC operation in round 1, compute $z_1[8, 10, 11]$ and $w_1[11]$ (marked by $\boxed{8}$) from $z_1[9]$ and $w_1[8, 9, 10]$.
   (a) Compute backward to get $x_1[2, 7, 8]$ (marked by $\overleftarrow{8}$).
   (b) Compute forward to get $k_2[11]$ (marked by $\overrightarrow{8}$).
9. According to the key relations, deduce $k_1[3, 7, 10]$, $k_3[2]$ and $k_4[3, 7]$ (marked by $\boxed{9}$) following the order in Table 12.
   (a) Compute backward to get $w_2[2]$ (marked by $\overleftarrow{9}$).
   (b) Compute forward to get $x_4[3, 7]$ and $z_4[7, 11]$ (marked by $\overrightarrow{9}$).
10. Guess $k_3[0]$ (marked by $\boxed{10}$) and deduce $k_1[8]$ and $k_4[8]$ (marked by $\boxed{10}$). Compute backward to $w_2[0]$ (marked by $\overleftarrow{10}$) and compute forward to $z_4[8]$ (marked by $\overrightarrow{10}$).
11. For column 1 over the MC operation in round 2, compute $z_2[0, 1]$ and $w_2[1, 3]$ (marked by $\boxed{11}$) from $z_2[2, 3]$ and $w_2[0, 2]$.
    (a) Compute backward to get $x_2[0, 5]$ and $w_1[5]$ (marked by $\overleftarrow{11}$).
    (b) Compute forward to get $k_3[1, 3]$ (marked by $\overrightarrow{11}$).
12. According to the key relations, deduce $k_1[9, 11, 15]$, $k_2[3]$, $k_3[7]$ and $k_4[9, 11]$ (marked by $\boxed{12}$) following the order in Table 12.
    (a) Compute backward to get $w_2[7]$ (marked by $\overleftarrow{12}$).
    (b) Compute forward to get $x_4[9, 11]$ and $z_4[5, 15]$ (marked by $\overrightarrow{12}$).
13. Guess $k_3[4]$ (marked by $\boxed{13}$) and deduce $k_1[12]$, $k_2[0]$ and $k_4[12]$ (marked by $\boxed{13}$. Compute backward to $w_1[0]$ and $w_2[4]$ (marked by $\overleftarrow{13}$) and compute forward to $z_4[12]$ and $w_4[12, 13, 14, 15]$ (marked by $\overrightarrow{13}$).

14. For column 1 over the MC operation in round 2, compute $z_2[4,7]$ and $w_2[5,6]$ (marked by $\boxed{14}$) from $z_2[5,6]$ and $w_2[4,7]$.
    (a) Compute backward to get $x_2[3,4]$ and $w_1[3,4]$ (marked by $\overleftarrow{14}$). Then compute $x_1[4,9,14,3] = \mathsf{SB}^{-1} \circ \mathsf{MC}^{-1}(w_1[4,5,6,7])$ (marked by $\overleftarrow{14}$).
    (b) Compute forward to get $k_3[5,6]$ (marked by $\overrightarrow{14}$).
15. According to the key relations, deduce $k_1[13,14]$, $k_2[0,1]$ and $k_4[13,14]$ (marked by $\boxed{15}$).
    (a) Compute backward to $w_1[1,2]$ (marked by $\overleftarrow{15}$) and deduce $x_1[0,5,10,15] = \mathsf{SB}^{-1} \circ \mathsf{MC}^{-1}(w_1[0,1,2,3])$ (marked by $\overleftarrow{15}$).
    (b) Compute forward to $z_4[6,9]$ (marked by $\overrightarrow{15}$). Deduce columns 1,2 of $w_4 = \mathsf{MC}(z_4)$ (marked by $\overrightarrow{15}$).

Degree of freedom and complexity. There are totally 33 active Sboxes in the inbound phase, including $s_1 = 30$ active Sboxes with probability 2^{-7} and $s_2 = 3$ active Sboxes with probability 2^{-6}. Therefore, by accessing the DDT, there expect $2^{30+6}/2 = 2^{35}$ combinations for the 33 active Sboxes, *i.e.*, there are 2^{35} choices for the bytes marked by $\boxed{1}$. Given one out of 2^{35} choices marked by $\boxed{1}$, seven bytes $k_2[12,13], k_3[0,4,8,12,13]$ (marked by a wavy line) are guessed in steps 2, 5, 10 and 13. Since there is no conflict, there expect $2^{35+56} = 2^{81}$ starting points satisfying the inbound differential. The probability of the outbound phase is $2^{-p_{out}} = 2^{-20}$. We have enough degrees of freedom to satisfy the outbound phase. Therefore, the total complexity of the 7-round key-collision attack on AES-192-DM is about $\mathcal{T} = 2^{20}$. We have practically implemented the attack and found some key pairs (K_1, K_2) and free plaintexts P in Table 7.

1.	$w_3 = \mathsf{MC} \circ \mathsf{SR}(y_3)$	$k_4[0,5,10,15] = (x_4 \oplus w_3)[0,5,10,15]$
2.	$k_2[4] = k_3[12] \oplus k_3[8]$	$k_2[8] = k_4[0] \oplus k_3[12]$
	$k_1[0] = k_2[8] \oplus k_2[4]$	$w_2[8,12,13] = x_3[8,12,13] \oplus k_3[8,12,13]$
3.	$w_2[9,10,11], z_2[11] = \mathsf{MC}(z_2[8,9,10], w_2[8])$	$w_2[14,15], z_2[12,13] = \mathsf{MC}(z_2[14,15], w_2[12,13])$
	$k_3[9,10,11,14,15] = (w_2 \oplus x_3)[9,10,11,14,15]$	
4.	$k_2[5] = k_3[13] \oplus k_3[9]$	$k_2[6] = k_3[14] \oplus k_3[10]$
	$k_2[7] = k_3[15] \oplus k_3[11]$	
5.	$k_1[4] = k_2[12] \oplus k_2[8]$	$k_4[4] = k_2[12] \oplus k_4[0]$
	$k_4[1] = k_2[13] \oplus k_4[5]$	$k_2[9] = k_4[1] \oplus k_3[13]$
	$k_1[5] = k_2[13] \oplus k_2[9]$	$k_1[1] = k_2[9] \oplus k_2[5]$
6.	$w_1[14,15], z_1[13,15] = \mathsf{MC}(z_1[12,14], w_1[12,13])$	$k_2[14,15] = (w_1 \oplus x_2)[14,15]$
7.	$k_2[10] = k_2[14] \oplus k_1[6]$	$k_1[2] = k_2[10] \oplus k_2[6]$
	$k_4[2] = k_2[10] \oplus k_3[14]$	$k_4[6] = k_2[14] \oplus k_4[2]$
8.	$z_1[8,10,11], w_1[11] = \mathsf{MC}^{-1}(z_1[9], w_1[8,9,10])$	$k_2[11] = w_1[11] \oplus x_2[11]$
9.	$k_1[3] = k_2[11] \oplus k_2[7]$	$k_1[7] = k_2[15] \oplus k_2[11]$
	$k_4[3] = k_2[11] \oplus k_3[15]$	$k_4[7] = k_2[15] \oplus k_4[3]$
	$k_3[2] = k_4[10] \oplus \mathsf{SB}(k_4[7])$	$k_1[10] = k_3[2] \oplus \mathsf{SB}(k_2[15])$
10.	$k_1[8] = k_3[0] \oplus \mathsf{SB}(k_2[13]) \oplus const$	$k_4[8] = k_3[0] \oplus \mathsf{SB}(k_4[5]) \oplus const$
11.	$z_2[0,1], w_2[1,3] = \mathsf{MC}^{-1}(z_2[2,3], w_2[0,2])$	$k_3[1,3] = w_2[1,3] \oplus x_3[1,3]$
12.	$k_1[9] = k_3[1] \oplus \mathsf{SB}(k_2[14])$	$k_1[11] = k_3[3] \oplus \mathsf{SB}(k_2[12])$
	$k_4[9] = k_3[1] \oplus \mathsf{SB}(k_4[6])$	$k_4[11] = k_3[3] \oplus \mathsf{SB}(k_4[4])$
	$k_3[7] = k_4[11] \oplus k_4[15]$	$k_2[3] = k_3[7] \oplus k_3[11]$
	$k_1[15] = k_3[7] \oplus k_3[3]$	
13.	$k_1[12] = k_3[4] \oplus k_3[0]$	$k_2[0] = k_3[4] \oplus k_3[8]$
	$k_4[12] = k_3[4] \oplus k_4[8]$	
14.	$z_2[4,7], w_2[5,6] = \mathsf{MC}^{-1}(z_2[5,6], w_2[4,7])$	$k_3[5,6] = w_2[5,6] \oplus x_3[5,6]$
15.	$k_1[13] = k_3[5] \oplus k_3[1]$	$k_1[14] = k_3[6] \oplus k_3[2]$
	$k_2[1] = k_3[9] \oplus k_3[5]$	$k_2[2] = k_3[10] \oplus k_3[6]$
	$k_4[13] = k_3[5] \oplus k_4[9]$	$k_4[14] = k_3[6] \oplus k_4[10]$

Table 12: Equations in the guess-and-determine steps for 7-round AES-192-DM. The blue bytes are guessed.