Automatic Quantum Multi-collision Distinguishers and Rebound Attacks with Triangulation Algorithm

Zhenzhen Bao[3,4], Jian Guo[2], Shun Li[1,2](✉), and Phuong Pham[2]

[1] School of Cryptology, University of Chinese Academy of Sciences, Beijing 100089, China lishun@ucas.ac.cn
[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
guojian@ntu.edu.sg, pham0079@e.ntu.edu.sg
[3] Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing, China zzbao@mail.tsinghua.edu.cn
[4] Zhongguancun Laboratory, Beijing, China

Abstract. In EUROCRYPT 2020, Hosoyamada and Sasaki found that differential paths with probability $2^{-2n/3}$ can be useful in quantum collision attacks, v.s. $2^{-n/2}$ for classical collision attacks. This observation led to attacks for more rounds on some AES-like hash functions. In this paper, we quantize the multi-collision distinguisher proposed by Biryukov, Khovratovich, and Nikolić at CRYPTO 2009, and propose quantum multi-collision distinguishers. We use CP-tool to automatically search for the configurations for multi-collision distinguishers and rebound attacks by taking into account related-key/single-key differentials of the underlying block cipher. We apply our method to AES-like primitives including block ciphers AES, Rijndael, Saturnin and AES-hashing modes AES-DM and AES-HCF.

Keywords: post-quantum cryptography, multicollision, free variable, BHT, related-key differential trail, distinguisher

1 Introduction

Recently, post-quantum security of cryptographic systems and primitives has received more and more attention from cryptographic researchers, developers, and users due to the progress in the development of quantum computers. The security of *public-key* crypto-systems such as RSA, DSA, and ECDH/ECDSA can often be reduced to some mathematically difficult problems such as factoring and discrete logarithm. However, Shor's seminal work [32] can be used to solve both problems efficiently with a sufficiently large quantum computer, which directly destroys the security of the public-key cryptographic schemes based on them in the post-quantum world. Due to such concerns, researchers have begun to investigate and develop post-quantum cryptographic algorithms, serving as replacements of the current public-key crypto-systems, with security against attackers aided by both quantum and classical computers. In the meantime, NIST has initiated a competition to develop post-quantum standards for key establishment schemes and digital signature schemes since 2017 [29]. On the other hand, *symmetric-key* crypto-systems usually are not built upon the security assumption of hard mathematical problems due to performance needs, and the research on how quantum computers would affect their security strength is more recent. In 1996, Grover [15] found quantum algorithms could be faster for *bruteforce* search than in the classical setting. This algorithm runs in time $\sqrt{N}$ for a space of size N, due to which halved (in bits) security strength is now considered as the *generic* lower security bound of a classical symmetric-key primitive in some quantum settings. Besides, recent studies showed that there exist non-trivial quantum attacks other than direct Grover search. In 2010, Kuwakado and Morii [24] used SIMON's algorithm [33] to distinguish the 3-round Feistel scheme from a random permutation in the quantum setting. After that, SIMON's algorithm has been applied to other symmetric-key schemes such as Even-Mansour scheme [23], message authentication codes (MACs) [20], and FX construction [25]. Invented in 1997, SIMON's algorithm allows to find a "hidden period" with only polynomially many queries and time. In most of the previous works utilizing SIMON's algorithm, the attacker tries to construct a function in such a way that the existence of the hidden period depends on the key values, which can be recovered once the period is detected. In addition to SIMON's algorithm, collision-finding utilizing Grover search is another prominent approach as dedicated attacks against symmetric-key primitives in quantum setting.

1.1 Collision

Preimage, second-preimage, and collision resistance form the basic security requirements for a hash function in the classical setting, and the same is expected in the quantum setting, *e.g.*, some public-key schemes have been proven to be post-quantum secure in the quantum random oracle model (QROM) [4] when instantiated with a post-quantum secure hash function. The known generic best time bounds in the quantum setting so far are $n/2$ bits for preimage resistance due to Grover's algorithm, and $n/3$ bits for collision resistance due to the BHT algorithm [5] named after Brassard, Høyer, and Tapp in 1998.

The BHT algorithm finds collisions with a query complexity of $O(2^{n/3})$ and $O(2^{n/3})$-qubit quantum random access memory (qRAM), which is a quantum analogue of the random access memory (RAM) allowing to efficiently access data in quantum superpositions. In 2017, Chailloux, Naya-Plasencia, and Schrottenloher [8] proposed the CNS collision finding algorithm with a time complexity of $O(2^{2n/5})$, a quantum memory of $O(n)$ qubits, and a classical memory of $O(2^{n/5})$ bits. The complexities of both BHT and CNS algorithms are optimized towards lowest possible time. When it comes to time-memory tradeoff with the merit of $T \times M$, the simple Grover search achieves the best $2^{n/2}$ (although this is not proven) with $O(2^{n/2})$ time and $O(1)$ memory, while BHT gives $2^{n/3} \cdot 2^{n/3} = 2^{2n/3}$ and CNS $2^{2n/5} \cdot 2^{n/5} = 2^{3n/5}$. Memoryless version of birthday attack for collision finding in classical setting is offered by the Pollard's rho method [31] in 1975, and an extension for parallelism was given by Van Oorschot and Wiener [37] in 1999.

The above mentioned algorithms are generic and do not exploit any internal characteristics of the primitives. The first dedicated quantum collision attack on hash functions was proposed at EUROCRYPT 2020 by Hosoyamada and Sasaki in [18], which shows differentials whose probability was too low for classical collision search can become useful in the quantum setting. They applied a quantum version of the rebound attack [28] to round-reduced AES hashing modes and Whirlpool, and extended the number of attacked rounds for collision finding from 6 and 5 rounds in classical setting to 7 and 6 rounds in quantum setting, respectively. These collision finding algorithms are considered as *attacks* when they require fewer time and memory than that of BHT algorithm ($T = M = O(2^{n/3})$). Later, Dong *et al.* [12] followed the CNS algorithm and presented an improved quantum rebound attacks on AES hashing modes and Grøstl-512 in a setting, where only a small amount of qRAM is available and the required resources are less than that of CNS ($T = O(2^{2n/5})$, $qM = O(n)$, $cM = O(2^{n/5})$). Very recently, Hosoyamada and Sasaki [19] proposed the first dedicated quantum collision attacks on SHA-256 and SHA-512 in another setting where the efficiency is evaluated by the time-memory tradeoff compared against $O(2^{n/2})$ by Pollard's rho.

1.2 Quantum Multi-Collision

The generic bound of collision resistance in classical setting is $2^{n/2}$ due to birthday attack, hence a differential based collision finding algorithm constitutes an attack only if the the probability of the underlying differential path of the bruteforce search phase is higher than $2^{-n/2}$. Hosoyamada and Sasaki [18] observed that, while in the quantum setting, the generic time bound is $2^{n/3}$ due to BHT algorithm, and the admissible differential probability can be as low as $2^{-2n/3}$, for which the bruteforce search of the conforming pair costs time $2^{n/3}$ and negligible quantum or classical memory by Grover's search in quantum computers. Taking advantage of this gap in the admissible probability ($2^{-2n/3}$ in quantum v.s. $2^{-n/2}$ in classical setting), differential paths with lower probability, but for more rounds, become useful hence lead to collision attacks for more rounds in quantum setting. This gap was further enlarged by considering higher time bound in CNS algorithm in [12], with $T = O(2^{2n/5})$ and admissible probability $2^{-4n/5}$, and time-memory tradeoff in [19] with $T = O(2^{n/2})/S$ and admissible probability $2^{-n} \cdot S^2$ when S qubits are needed to implement the attack in quantum circuit or for qRAM.

Motivated by [18], in this paper we consider the problem of q-multicollision finding in the quantum setting, which is a natural generalization of the collision finding problem. And similarly to [18], we also consider the scenario where bruteforce (resp. Grover search in quantum setting) is used to find conforming pairs of a given differential path. When the search is limited by time T, the admissible differential probability is at least T^{-1} (resp. T^{-2} in quantum). In 2019, Liu and Zhandry [27] proved that the necessary and sufficient query complexity (hence tight bound) for the quantum q-multicollision problem is $N^{\frac{1}{2}\cdot(1-\frac{1}{2^q-1})}$ aided

Target	#Round	Attack	T_c	T_q	q_c	q_q	Mem	Reference
AES-128	6	CMC/QMC	$q \cdot 2^{36}$	$q \cdot 2^{18}$	✓	✓	–	[34]
	7		$q \cdot 2^{90}$	$q \cdot 2^{45}$	4	✓	–	Fig. 6
	8		$q \cdot 2^{112}$	$q \cdot 2^{56}$	9	4	–	Fig. 2
	8		$q \cdot 2^{105}$	$q \cdot 2^{52.5}$	6	✓	2^{16}	Fig. 6
AES-192	10	CMC/QMC	$q \cdot 2^{84}$	$q \cdot 2^{42}$	✓	✓	–	[16]
	12(Full)	CMC/QMC	$q \cdot 2^{102}$	$q \cdot 2^{51}$	5	✓	–	Fig. 10
Rijndael-128-160	9	CMC/QMC	$q \cdot 2^{60}$	$q \cdot 2^{30}$	✓	✓	–	Fig. 9
	10	CMC/QMC	$q \cdot 2^{90}$	$q \cdot 2^{45}$	4	✓	–	Fig. 7
	11(Full)	CMC/QMC	$q \cdot 2^{118}$	$q \cdot 2^{59}$	13	4	–	Fig. 7
Rijndael-128-224	11	CMC/QMC	$q \cdot 2^{67}$	$q \cdot 2^{33.5}$	✓	✓	–	Fig. 8
	13(Full)	CMC/QMC	$q \cdot 2^{97}$	$q \cdot 2^{48.5}$	5	✓	–	Fig. 8
Rijndael-160-192	11	CMC/QMC	$q \cdot 2^{90}$	$q \cdot 2^{45}$	✓	✓	–	Fig. 10
Rijndael-160-256	12	CMC/QMC	$q \cdot 2^{108}$	$q \cdot 2^{54}$	4	✓	–	Fig. 9
Saturnin	7	CMC/QMC	$q \cdot 2^{143}$	$q \cdot 2^{71.5}$	✓	✓	–	Fig. 3
	8	CMC/QMC	$q \cdot 2^{171.2}$	$q \cdot 2^{85.6}$	4	✓	–	Fig. 3
	10	CMC/QMC	$q \cdot 2^{249.3}$	$q \cdot 2^{124.7}$	39	6	–	Fig. 3

Table 1: Summary of results on quantum multi-collision distinguishers against AES, Rijndael, and Saturnin. Hereafter, CC is classical collision attack; QC is quantum collision attack; CMC is classical multi-collision attack; QMC is quantum multi-collision attack. q_c, q_q denotes the smallest q for valid CMC/QMC distinguishers, and ✓ for all $q \geq 3$.

by the same amount of qRAM. Following the $N^{(q-1)/q}$ bound by Suzuki *et al.* (after removing the polynomial factors), the admissible probability is $N^{\frac{1}{q}-1}$ in classical v.s. $N^{\frac{1}{2^q-1}-1}$ in quantum setting, which exhibits a similar gap as for the differential based collision attacks.

1.3 Our Contributions

Following the observation on the gap of admissible probabilities, in this paper we propose the Quantum Multi-Collision (QMC) distinguisher, as quantized version of the q-multicollision distinguisher proposed in [2]. Our model shows differentials with probability as low as 2^{-n}, for a block cipher with n-bit block size, will be useful in mounting QMC attack, compared with $2^{-2n/3}$ and $2^{-4n/5}$ for quantum collision attack considered in [18] and [12], respectively. We apply the attack framework to AES, Rijndael, and Saturnin [7], and find a rich set of results summarized in Table 1. All the results surpass the classical distinguishing attacks in number of rounds and/or the success probability.

To the best of our knowledge, our work is the first dedicated quantum distinguishing attack on block ciphers that utilize the gap between lower bounds of query complexity of generic classical multicollision and generic quantum multicollision. To demonstrate the flexibility of multi-collision attacks, the free-start collision attack 10-round AES-hashing modes are given in Table 2.

Organization. Section 2 gives a brief introduction of AES-like primitives, quantum computation, qRAMs, and quantum adversary models. Section 3 introduces the related quantum collision algorithms and quantum multi-collision algorithms for ideal functions. Then, Section 4 gives our attack framework and techniques involved, followed by applications to AES-128, Rijndael, and Saturnin in Section 5.

Target	A.R.	Attack	T_c	T_q	Mem	Reference
AES-256-DM	10	Free-start col.	2^{49}	–	–	Fig. 5
AES-256-DM	10	Free-start col.	–	$2^{25.61}$	–	Fig. 5
AES-256-DM	14	Pseudo col.	$q \cdot 2^{67}$	–	–	[2]
AES-256-DM	14	Collision	–	$2^{51.2}$	$2^{25.6}$	[8]
AES-192-HCF	7	Free-start col.	2^{64}	–	–	Fig. 5
AES-192-HCF	7	Free-start col.	–	$2^{33.37}$	–	Fig. 5
AES-256-HCF	10	Free-start col.	–	$2^{86.07}$	–	[22]
AES-256-HCF	14	Free-start col.	–	$2^{100.3}$	–	[1]

Table 2: Results on classical and quantum free-start collision attacks against AES-hashing modes

Section 6 introduces classical/quantum attacks on AES-hashing modes with use of previous techniques and enhanced trails. Section 7 concludes the paper. Some details of the work are postponed to Appendix.

2 Preliminary

2.1 Quantum Computation and Quantum RAM

Similar to the time complexity estimation on classical computers, the unit of time complexity on quantum computers refers to the computational effort required to execute the underlying primitive once. The actual time to run a quantum attack will depend on many factors including the hardware architectures of quantum computers. In what follows, we consider the simple computational model that each pair of qubits in a quantum computer can interact with one another. Based on this model, the time complexity of dedicated algorithms is evaluated and compared against the generic bounds under the same model. Such algorithms are only considered as *valid attacks* if they require less resources like time and/or space than the generic bounds. Here, space complexity refers to the number of qubits to implement the attack, and similarly that needed to implement the underlying primitive is one *unit* of space.

Random-access memory (RAM) is a form of computer memory that supports read and write in any order, and the access time is often assumed to be constant in the time complexity evaluation of cryptanalysis. Quantum random-access memory (qRAM) is the quantum analog of the RAM, which supports data access and computation in superpositions. For simplicity of complexity evaluation, we assume similarly to RAM that access time of qRAM is constant, and that for reading or writing of one cell is considered as a unit. Furthermore, we do not distinguish qubits used as memory like qRAM and the qubits used for quantum circuit implementations of a function.

2.2 Grover's algorithm

Given a search space of N elements $\{1, 2, \ldots, N\}$, a Boolean function $f : \{1, 2, \ldots, N\} \to \{0, 1\}$, and $a \triangleq |f^{-1}(1)|/N$ the probability for a random x resulting in $f(x) = 1$, the best classical algorithm with black-box access to f requires $1/a$ queries in order to find one x with $f(x) = 1$ for a probability more than 0.5. This is usually referred to as the bruteforce search in the classical setting. However, in the quantum setting with quantum black-box oracle access to f, Grover's algorithm finds x with $\Theta(\sqrt{1/a})$ quantum queries to the quantum oracle O_f[5], which is defined as:

$$|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle .$$

Starting with a uniform superposition $|\phi\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle$, by applying the Hadamard transformation $H^{\otimes n}$ to $|0\rangle^{\otimes n}$ where $n = \log_2 N$, the Grover's algorithm iteratively applies the unitary transformation $(2|\phi\rangle\langle\phi| - I)O_f$ to $|\phi\rangle$ so that the amplitudes of those x's with $f(x) = 1$ will be amplified. When measuring the resulting state, a value of x of interest will be returned with overwhelming probability. Due to this nature, Grover's is also viewed as quantum analog of bruteforce search.

2.3 Quantum multicollision algorithm

Quantum q-multicollision algorithm considers the scenario where $F : X \to Y$ is a q-to-1 function with $|X| = q|Y| = qN$. The algorithm from [27] needs $O(N^{\frac{1}{2}(1-\frac{1}{2^q-1})})$ quantum queries, aided by the same amount of qRAM. It is noted that the BHT algorithm is a special case of $q = 2$.

3 The Quantum *q*-multicollision Distinguisher

When the underlying function is ideal and can only be queried as a blackbox, Liu and Zhandry [27] as reviewed in previous section give an algorithm of complexity $O(N^{(2^{q-1}-1)/(2^q-1)})$ for the q-multicollision finding problem with proven tight bounds. Hence, a dedicated algorithm which finds q-multicollision with fewer time and/or qRAM will be considered a valid distinguisher, which in turn implies the function under attack is not ideal. Biryukov *et al.* [2] define the following function

$$F_{\Delta_K,\Delta_P}(K, P) = E_K(P) \oplus E_{K\oplus\Delta_K}(P \oplus \Delta_P),$$

where E_K is the block cipher of interest. F with (K, P) as the input can be considered as a pseudo-random function according to Patarin [30]. Then the q-multicollision for F can be defined as follows.

Definition 1 ([2]). *Given two fixed differences Δ_K and Δ_P. A q-multicollision of a cipher $E_K(\bullet)$ is a set of q $(q \geq 2)$ pairs: $\{(P_1, K_1), (P_2, K_2), \ldots, (P_q, K_q)\}$ that satisfies*

$$\begin{aligned} E_{K_1}(P_1) \oplus E_{K_1\oplus\Delta_K}(P_1 \oplus \Delta_P) = E_{K_2}(P_2) \oplus E_{K_2\oplus\Delta_K}(P_2 \oplus \Delta_P) = \\ = \cdots = E_{K_q}(P_q) \oplus E_{K_q\oplus\Delta_K}(P_q \oplus \Delta_P) = \Delta_C, \end{aligned} \tag{1}$$

where $P_i \oplus \Delta P \neq P_j$ and $K_i \oplus \Delta K \neq K_j$ with $i, j \in \{1, 2, \ldots, q\}$.

[5] Here we assume use of O_f, alternative oracle could be O_ω which flips the phase of good states: $O_\omega |x\rangle = (-1)^{f(x)} |x\rangle$

Here the two differences $\varDelta_K$ and $\varDelta_P$ are fixed following [2, Remark 2], while allowing two differences $\varDelta_K$ and $\varDelta_P$ to attackers' choice leads to a difference of complexity of the generic bounds between $q \cdot N^{\frac{q-2}{q+2}}$ in [2] for unfixed differences v.s. $\frac{q}{e} \cdot N^{\frac{q-1}{q}}$ in [35] for fixed differences. Scenario in [2, Remark 2] is considered in this paper.

From the block cipher's perspective when there exists a high probability differential path, F is essentially the XOR difference of a ciphertext pair following the differential path under plaintext difference $\varDelta_P$ and key difference $\varDelta_K$. In such case, the above q-multicollision can be translated into finding q conforming pairs, which costs time $q \cdot p^{-1/2}$ by Grover search for a differential path with probability p. To ensure this leads to a valid distinguisher with lower than generic attack complexity, we need $q \cdot p^{-1/2} \leq O(N^{(2^{q-1}-1)/(2^q-1)})$ with $N = 2^n$, then the admissible probability should be $p \geq 2^{-n(1-\frac{1}{2^q-1})}$.

Note, although the differential is under the related-key setting with key difference fixed, the entire model is however under the chosen-key setting since the secret key K is also an input to F which can be chosen freely by the attackers.

Comparison with CMC. In [2], Biryukov *et al.* applied the CMC distinguishing attack to the full 14-round AES-256. Rather than following directly the bound of CMC ($2^{n(1-1/q)}$) from [36], they considered the repeated queries under the related-key setting by allowing $(\varDelta_P, \varDelta_K, \varDelta_C)$ to attackers' control and came up with a slightly twisted bound, up to a constant-factor difference in q. In our model $(\varDelta_P, \varDelta_K, \varDelta_C)$ are fixed, we follow rightfully the bounds of [36] for classical setting and that of Liu and Zhandry's QMC algorithm [27] in quantum setting. Although both CMC and QMC allow as low as 2^{-n} as the admissible probability for sufficiently large q, QMC allows $2^{-n(1-1/(2^q-1))}$, which is smaller than that for CMC ($2^{-n(1-1/q)}$) for any fixed $q \geq 2$.

4 The Attack Framework and Techniques

To find the q-multicollision of a given block cipher, we follow Definition 1 to find q conforming pairs following a differential path. The overall attack procedure, as depicted in Figure 1, works in the following 4 steps, with time complexity optimization in mind.

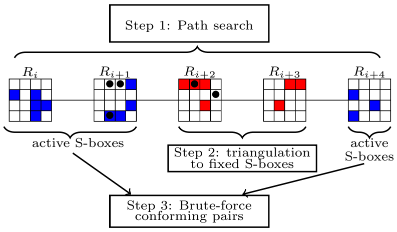


Figure 1: The attack framework

Step 1: find a high-probability differential path under the related-key setting. To find differential path with highest possible probability, automatic search tools are reviewed and that from [16] are invoked here. Δ_P and Δ_K are determined together with the path.

Step 2: some probabilistic transitions in the differential path can be fulfilled deterministically by presetting some state values. In case of our mainly concerned AES-like ciphers, the input or output of these active Sboxes can be fixed utilizing such degrees of freedom from both state and key bytes. To maximize the number of such fixed active Sboxes, triangulation algorithm developed in [21] will be reviewed and used.

Step 3: the remaining active Sboxes are fulfilled by Grover search, where candidates are generated from the remaining degrees of freedom from Step 2.

Step 4: additional adhoc optimizations are done to minimize the final complexity.

In the sequel, we introduce the techniques and algorithms used in each step.

4.1 Automatic tools for related-key differential paths

Generic solver Constraint Programming (CP) is used to solve Constraint Satisfaction Problems (CSPs). A CSP is defined by a triple $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where

- $\mathcal{X}$ is a finite set of variables;
- $\mathcal{D}$ refers to the domain, *i.e.*, the set of values each $x_i \in \mathcal{X}$ can take;
- $\mathcal{C}$ is a set of constraints including relations between variables.

When an objective function is defined, the CSP becomes a Constrained-Optimization Problem (COP). A solution of a COP is an assignment of values to all the variables in $\mathcal{X} = \{x_0, \cdots, x_{n-1}\}$ such that all constraints from $\mathcal{C} = \{c_0, \cdots, c_{m-1}\}$ are satisfied and objective function achieves maximum or minimum.

Finding an optimal related-key differential trail is a highly combinatorial problem that hardly scales. To simplify this problem, a usual and efficient way is to divide it into two steps [3,13]. Step 1 searches for all truncated differential characteristics under a given bound on the number of rounds and active S-Boxes. It may happen that no actual differential characteristic follows the truncated differential found in Step 1. Hence Step 2 examines and decides whether the truncated differential characteristics are valid, and finds the actual differential characteristic that maximizes the probability. Both steps can be approached by CP. Such CP strategy has been successful in finding related-key differential characteristics for AES [11,17], Midori [14], and SKINNY [10,26,34], in the sense that the truncated differentials match the lower bound on the number of active S-boxes (a.k.a. optimal truncated differentials).

4.2 Triangulation Algorithm

The Triangulation Algorithm(TA) proposed in [21] uses Gaussian elimination to solve systems of non-linear equations. Unlike a universal algorithm dealing with any non-linear function, it is efficient for solving system of bijective functions only. When a differential characteristic of an AES-like block cipher is given, and the attacker is given full control over the state and key values, we are interested in finding the maximum amount of active S-boxes that can be fulfilled by setting to the respective conforming values. TA serves this purpose, with state bytes and key bytes as variables, and the round function and key schedule as the system of equations. This can be applied to multiple rounds one by one, *i.e.*,TA is applied to one round, then to the next rounds with only those free variables returned by the TA from the previous round, this is repeated until all free variables are exhausted. Our implementation shows the problem sizes in our attack are small and all our TA programs can finish execution instantly on a PC.

4.3 Complexity Optimizations

To further minimize the overall complexity, the following measures are integrated into the attack procedure.

1. TA is applied to consecutive rounds, for as many rounds as possible. This process is repeated for each starting round, and the maximum number of fixed active S-boxes is selected among all choices.
2. We note it is not necessary that the optimal differential characteristic from Step 2 leads to the lowest attack complexity after the execution of TA. Hence, instead of a single optimal path, a set of sub-optimal paths are collected from Step 2, then TA is run for all the paths to identify the best one with highest remaining probability.
3. It is noted that S-box operation only applies to the last column of the round key bytes in the key schedule, while this is for all state bytes in round function. Hence, it is likely many degrees of freedom from key bytes will be used to fulfill some active S-boxes in the state. However, a key byte variable may not affect all state bytes in a bijective way.

5 Applications on AES, Rijndael and Saturnin

In this section, detailed attack procedure on AES will be given for the readers to follow the techniques, then brief results are described for subsequent targets. To make a comprehensive comparison, besides QMC we also apply the attack framework to find CMC. The primary notations used in this section are listed in Table 3.

SB	SR	MC	MR	AK	KS	RCON
SubBytes	ShiftRows	MixColumn	MixRow	AddRoundKey	KeySchedule	Round Constant

Table 3: Notations

5.1 AES

Description of AES. AES-k is a block cipher family of 128-bit block and k-bit key for $k \in \{128, 192, 256\}$. The state has 16 bytes and can be represented as a 4×4 matrix. Given a $\mathrm{N_{row}} \times \mathrm{N_{col}}$ bytes of the state matrix, where $\mathrm{N_{row}} = \mathrm{N_{col}} = 4$. Then the state is encrypted by an iterative process which is repeated for 10, 12, and 14 rounds, for AES-128, AES-192, and AES-256, respectively. An AES round function is an Substitution-Permutation Network (SPN), and composed of four consecutive operations: SB, SR, MC, and AK. The master key k is added to the state before the application of the first round function, and is used to generated to r subkeys through the KS function.

Attack Procedure on AES. We apply the search tool described in Section 4.1 to find the related-key differential paths of AES-128 reduced to 7 rounds, 8 rounds, and the full 12-round AES-192, while that for the full 14-round AES-256 has been already found and used in [2, 16]. A lower bound of probability $2^{-(n+k)}$ is set to limit the search space, where n is the state size and k is the key size in bits. This bound is used to ensure there will be at least one pair of messages conforming the differential path, utilizing all degrees of freedom from both state and key. A set of differential paths are collected.

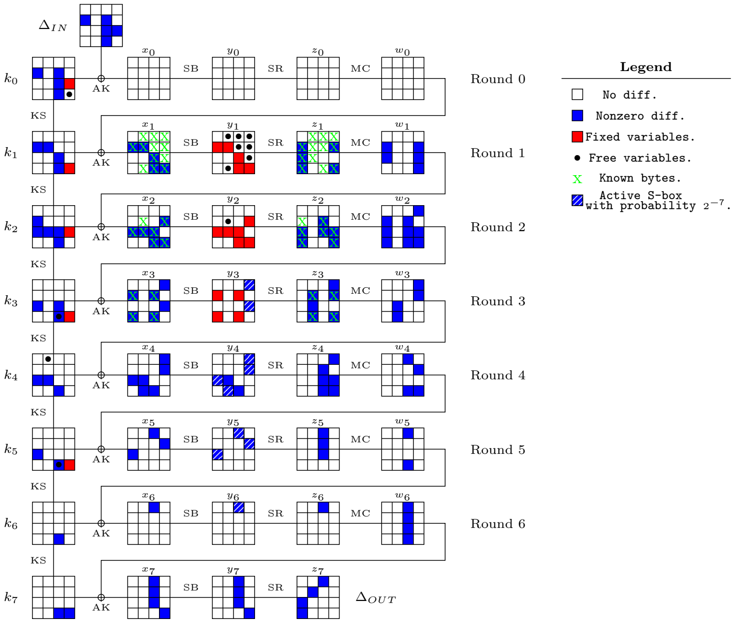


Figure 2: A differential path [6] for 8-Round AES-128. Known bytes are determined from the values of the free bytes.

[6] Legend is used for the same meaning for the subsequent differential paths

Let us denote by y_i, w_i, k_i the state after SB, MC, and the sub-key at i-th round. Each of them is an array of 16 bytes following order from columns to columns. Then, the i-th round function involving y_i, w_{i-1}, k_i and w_i can be re-expressed by Equation (2), where the first line $y_i \oplus S(w_{i-1} \oplus k_i)$ reassembles two operations AK and SB, and the second line reassembles SR and MC. The i-th round key schedule can be re-expressed by the Equation (3), relating key bytes of the current round k_i with that from the previous round k_{i-1}. The same Equation (3) is used for both AES-128 with $i = 0, \dots, 10$ and $b = 16$, and AES-192 with $i = 0, \dots, 8$ and $b = 24$. There are 32 free variables (16 from the state and 16 from the key) for AES-128, and 40 free variables (16 from the state and 24 from the key) for AES-192. Each equation will form a line with 1/0 indicating the presence of the respective variable in the matrix input to TA.

$$R_i : \begin{cases} y_i \oplus S(w_{i-1} \oplus k_i) = 0 \\ w_i \oplus \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix} \times \begin{pmatrix} y_i[0] & y_i[4] & y_i[8] & y_i[12] \\ y_i[5] & y_i[9] & y_i[13] & y_i[1] \\ y_i[10] & y_i[14] & y_i[2] & y_i[6] \\ y_i[15] & y_i[3] & y_i[7] & y_i[11] \end{pmatrix} = 0 \end{cases} \tag{2}$$

$$KS_i : \begin{cases} k_i[j] \oplus k_i[j-4] \oplus k_{i-1}[j] = 0, & j = 4, \dots b-1 \\ k_i[0] \oplus k_{i-1}[0] \oplus S(k_{i-1}[b-3]) \oplus \mathbf{RCON}_i = 0 \\ k_i[1] \oplus k_{i-1}[1] \oplus S(k_{i-1}[b-2]) = 0 \\ k_i[2] \oplus k_{i-1}[2] \oplus S(k_{i-1}[b-1]) = 0 \\ k_i[3] \oplus k_{i-1}[3] \oplus S(k_{i-1}[b-4]) = 0 \end{cases} \tag{3}$$

Cipher	Attacked Rounds	Active S-boxes Type-I + Type-II	Fixed bytes Type-I + Type-II	Final Probability p_{out}	Ref.
AES-128	6	$16 + 5$	$10 + 5$	2^{-36}	[34]
	7	$13 + 16$	$5 + 10$	2^{-90}	Fig. 6
	8	$10 + 27$	$(3 + 17)$ or $(3 + 18)$	2^{-112} or 2^{-105}	Fig. 2
AES-192	10	$23 + 8$	$10 + 8$	2^{-84}	[16]
	12	$23 + 16$	$6 + 16$	2^{-102}	Fig. 10
AES-256	14	$22 + 2$	$11 + 2$	2^{-66}	[16]

Table 4: Summary of AES related-key differential paths

Results on 8-round AES-128. After CP tool run on a PC for a few minutes, the desired differential characteristics are found on 8-round AES-128. The one, as depicted in Figure 2 and specified in Figure 6, is formed with 37 active S-boxes, whereas 6 of them are in the sub-keys and 31 are in the state. For the AES S-box, there are two types of differentials with probability of 2^{-6} and 2^{-7}, which we will refer to as Type-I and Type-II. They are depicted in the figures of differential path as boxes in blue only, and blue with white lines, respectively. Among the 37 active

S-boxes, there are 10 Type-I and 27 Type-II, which gives an overall probability of $2^{-(10\times6+27\times7)} = 2^{-249}$. After execution of TA with all possible starting round, we find the one starting with Round 1 is the best choice, which allows to fix all active S-boxes in Round 1 and Round 2, and 4 of the state in Round 3, as well as 5 out of the 6 active S-boxes in sub-keys, as highlighted in red in the Figure 2. TA finds the active S-box in k_7 cannot be fixed probably because it is too far from the Round 1 variables. After TA, 7 Type-I and 10 Type-II active S-boxes are left unfixed, resulting in a probability of $p_{out} = 2^{-(7\times6+10\times7)} = 2^{-112}$. Grover search then finds a conforming pair in 2^{56} quantum queries, and hence a QMC in $q \cdot 2^{56}$. This complexity is lower than the generic bound when $p_l = 112/128 = 0.875 < 1 - \frac{1}{2^q - 1}$, *i.e.*, $q \geq 4$. The same differential leads to a CMC attack with complexity $q \cdot 2^{112}$, which is a valid attack when $p_l < 1 - 1/q$, *i.e.*, $q \geq 9$. The gap of the q ranges can be interpreted as the differential leads to a valid QMC attack but invalid CMC attack in the range $4 \leq q \leq 8$.

Remark 1. One may wonder if the degree of freedom (DoF) is sufficient for this attack, since there are in total 32 bytes DoF, $3 + 17 = 20$ have to be used as fixed bytes, and the remaining $32 - 20 = 12$ bytes (or 96 bits) are insufficient for a probability of 2^{-112}. It is important to note that none of DoF will be lost, *i.e.*, even for the bytes used as fixed bytes, there will be 2^1 or 2^2 solutions for each such Sbox of Type-II and Type-I, respectively. These leftover DoF together with those free bytes will be used to fulfill the final p_{out}. Hence, overall it is sufficient for us to ensure the overall probability 2^{-249} requires 249 DoF to fulfill, less than the total available 32 bytes (or 256 bits). Similar assurance has been done for all presented results during the differential search.

To check if TA works as expected and no byte is over-defined in the system, we verified the entire procedure to reproduce all other state and key bytes from the set of fixed bytes and free bytes. As also highlighted in red, the fixed bytes are $\{k_0[14], k_1[15], k_2[14], k_3[15], k_5[15], y_1[1], y_1[5], y_1[10], y_1[11], y_1[15], y_2[2], y_2[6], y_2[10], y_2[11], y_2[13], y_2[15], y_3[1], y_3[3], y_3[9], y_3[11]\}$, and the free bytes are $\{k_0[15], k_3[11], k_4[4], k_5[11], y_1[4], y_1[7], y_1[8], y_1[9], y_1[12], y_1[13], y_1[14], y_2[5]\}$. From these 32 bytes, Table 7 shows step by step how the entire key state k_2 (highlighted in red) and state y_1 (highlighted in blue) are derived. MC^{-1} is the operator acting on any 4 bytes out of 8 bytes of the columns before and after MC and resulting in the remaining 4 bytes.

Fixing one more active S-box. In this part, we describe a method to fulfill one additional active S-box at $y_3[12]$ for free at the cost of some qRAM. Along with the previous fixed bytes, we fix one more byte $k_3[12]$ and receive other free variables from another run of TA. Note that byte $k_3[12]$ will not be fixed to particular value, but will be chosen so that $k_3[12] \oplus x_3[12]$ fulfill the S-box at $y_3[12]$. Denote the value of $k_3[12]$ as x. Then the value of $y_3[12]$ depends on x and some more free variables and fixed variables, more precisely

$$y_3[12] = S(03 \times S(01 \times 02^{-1}x \oplus c_1) \oplus c_2 \oplus x), \tag{4}$$

where c_1 and c_2 are 8-bit values depending on the 31 free bytes and fixed bytes. A lookup table with the triplet (c_1, c_2, x) can be precomputed and stored in qRAM for superposition access. When the values of the 31 main bytes are fixed, the corresponding (c_1, c_2) can be computed, and suitable x can be identified from the lookup table so that the active S-box at $y_3[12]$ can be fulfilled. The cost of this lookup table is classical computation effort of 2^{16} and 2^{16} qRAM. This method allows to fix one additional Type-II active S-box, resulting in the final $p_{out} = 2^{-105}$ as depicted in Figure 2.

Results on 7-round AES-128. Since the QMC for 8-round AES-128 are not valid for all q values, and the previous quantum collision attacks work for 7 rounds only, we also run our attack framework to 7-round for comparison purposes. The CP tool returns in a few minutes the differential path depicted in Figure 6 with 29 actives S-boxes, out of which 22 S-boxes are in states and 7 are in subkeys. With $(13, 16)$ and $(8, 6)$ Type-I and Type-II active S-boxes before and after TA, the respective probabilities are 2^{-190} and 2^{-90}. Then $p_l = 90/128 = 0.703$ gives a valid QMC with complexity $q \cdot 2^{45}$ for $q \geq 3$ and a valid CMC with complexity $q \cdot 2^{90}$ for $q \geq 4$. To find the CMC attack valid for all possible q, we move on to reduce the attacked round to 6.

Results on 6-round AES-128. The best related-key differential of 6-round AES-128 has been found in [34], with 16 Type-I and 5 Type-II active S-boxes, and only 6 Type-I active S-boxes are left. This path gives a final $p_{out} = 2^{-36}$ and $p_l = 0.28$ leading to a valid CMC with complexity $q \cdot 2^{36}$ for all $q \geq 3$.

Results on 12-round AES-192. Similar attack procedure is applied to AES-192, and the probabilities of the differential path before and after application of TA, as depicted in Figure 10, are 2^{-250} and 2^{-102}, respectively. The $p_l = 102/128 \approx 0.80$ gives a valid QMC with complexity $q \cdot 2^{51}$ for $q \geq 3$ and a valid CMC with complexity $q \cdot 2^{102}$ for $q \geq 4$. The attacked round reduces to 10 to obtain the CMC attack valid for $q \geq 3$ with time complexity 2^{78}. All our results on AES are summarized in Table 4.

5.2 Rijndael

Cipher	Attacked	Active S-boxes	Fixed bytes	Final Probability	Ref.
	Rounds	Type-I + Type-II	Type-I + Type-II	p_{out}	
Rijndael-128-160	9	22 + 9	12 + 9	2^{-60}	Fig. 9
	10	27 + 9	12 + 9	2^{-90}	Fig. 7
	11	21 + 20	6 + 16	2^{-118}	Fig. 7
Rijndael-128-224	11	19 + 7	9 + 6	2^{-67}	Fig. 8
	13	28 + 11	14 + 10	2^{-97}	Fig. 8
Rijndael-160-192	11	31 + 14	16 + 14	2^{-90}	Fig. 9
Rijndael-160-256	12	35 + 7	18 + 7	2^{-108}	Fig. 10

Table 5: Summary of Rijndael related-key differential paths

Description of Rijndael. Rijndael-b-k (where b is the block size and k is the key size in bits) is the predecessor of AES designed by Daemen and Rijmen [9]. It has 25 variants corresponding to each case of $4 \times \mathrm{N_{col}}$ block size (128, 160, 192, 224 or 256 bits) and the key size (128, 160, 192, 224 or 256 bits). The number of rounds for the 25 instances are 10, 11, 12, 13, and 14 depending on the maximum of block size and key size. The encryption process is the same as AES as described in Section 5.1, except for the KS, SR, and the round numbers. Here we focus on the variants of block size 128 and 160 bits, for which SR works the same as AES by circularly shifting i-th row to the left by i positions.

Results on Rijndael. Our related-key differential characteristics are obtained by modifying the tool from [16] to fit Rijndael. There are 25 instances of Rijndael, we report in this paper only the longest rounds attacked due to space limit. The QMC can be mounted on full rounds of Rijndael-128-160, Rijndael-128-224, 11-round Rijndael-160-192, and 12-rounds of Rijndael-160-256. For Rijndael-128-160 (and similarly for Rijndael-128-224), we listed the results from 9 to 11 rounds, because 9 is the maximum rounds CMC attack can reach for $q = 3$, 10 is the valid QMC attack can achieve for $q = 3$, and 11 is the maximum rounds a valid QMC works for some bigger q. When TA is applied, active S-boxes in the states of up to 3 rounds can be fixed. Details of the differential paths, before and after the application of TA, are summarized in Table 5. It is interesting to note that differential paths leading to the best attack for 3 out of the 4 variants are optimal, except for Rijndael-160-256. In this special case, a differential path with 42 active S-boxes instead of the optimal one with 40 active S-boxes is used. This path has more active S-boxes in the keys rather than the state than the optimal path. After application of TA, this sub-optimal path leads to higher final probability p_{out}. This is consistent with our observation that more free variables are available from key bytes since there are less S-box operations, hence TA has better chance to fix active S-boxes in key.

5.3 Saturnin

Description of Saturnin. Saturnin [7] is a block cipher with a 256-bit state and 256-bit key that was designed as the derivative of AES with efficient implementation by Canteaut *et al.* for the NIST lightweight cryptography competition, and it was among the round 2 candidates. It can be viewed as a 3-dimensional AES with cell size of 4 bits. The composition of two consecutive rounds starting from even round is called super-round, which is very similar to an AES round operating on 16-bit words except that the SR is replaced by a transposition exactly as used in Square, the predecessor of AES.

Results on Saturnin. On Saturnin's design website, the authors propose a challenge to dig into the security analysis of Saturnin against the related-key differential attack, starting from 9 rounds. In [6], the designers proposed the first classical related-key attack on 10 rounds, and conclude

"*A quantized version of this attack is expected to reach less rounds ...* "

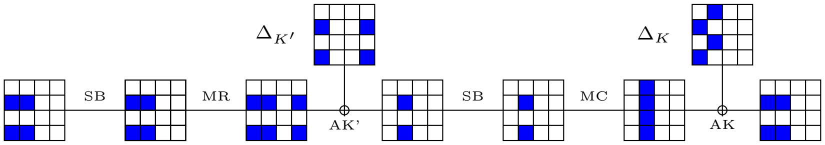


Figure 3: A differential path for 2-Round Saturnin [6]

In this paper we successfully mount QMC distinguishing attack on 10-round Saturnin. It does not directly violate designers' claim above since ours are not key-recovery attack, however reaches the 10-round boundary. We utilize the differential characteristic proposed by the designers in [6], where a 2-round iterative differential characteristic (refer to Figure 3) with probability $2^{-78.1}$ is given and repetition by 5 times leads to a 10-round related-key differential characteristic with probability $2^{-390.5}$.

Next, we prepare the system of equations as the input to TA. The key schedule of Saturnin is simple byte shuffle, which requires no extra equation to describe. Application of TA shows that all the active S-boxes in first 4 rounds of states except for the last active S-box in byte $y_3[7]$ can be fixed. Saturnin does not allow us to apply the trick used for AES to save degree of freedom of key bytes, because each byte of the key is involved in various equations. The relationship of the key byte and the corresponding state byte involves many more dependent variables $c_1, c_2, c_3, \cdots$, which increases the requirement of memory significantly. To this end, the probability of the S-box in byte $y_3[7]$ is not lower than 2^{-15}, which results in a $p_{out} = 2^{-249.3}$. This leads to a QMC with complexity $2^{124.65}$ and $q \geq 6$. The similar procedure attack is applied to 8 rounds to achieve general QMC, *i.e.,* $q \geq 3$, with $p_{out} = 2^{-171.2}$, which leads to the complexity $2^{85.6}$. To extend the CMC attack to $q = 3$, the same differential characteristic but reduced further to 7 rounds is used. This leads to a $p_{out} = 2^{-143}$, and a valid CMC attack with complexity 2^{143}, as summarized in Table 6.

Model	Rounds	Active S-boxes	Fixed bytes	Final Probability	Ref.
Saturnin	7	22	11	2^{-143}	Fig. 3
	8	24	11	$2^{-171.2}$	
	10	30	11	$2^{-249.3}$	

Table 6: Results on Saturnin' related-key differential paths

6 Rebound attacks on AES-hashing modes

The related-key trails used in multicollision distinguisher can be used to attack two AES-hashing modes in Figure 4 if $\Delta_P = \Delta_C$.

6.1 AES-DM Mode

AES-256-DM is Davies-Meyer hash mode instantiating AES-256. Let $F : \{0,1\}^{128} \times \{0,1\}^{256} \to \{0,1\}^{128}$ be AES-256-DM compress function such that $h_i = F(h_{i-1}, m_{i-1}) =$

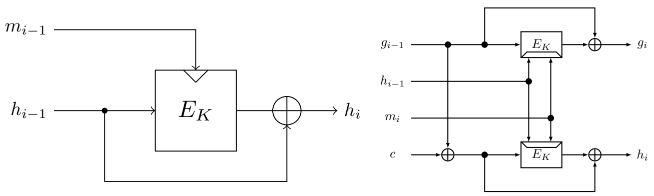


Figure 4: Left: Davies-Meyer (DM) mode. Right: Hirose's double block length (DBL) compress function.

$\mathsf{AES}_{m_{i-1}}(h_{i-1}) \oplus h_{i-1}$, then one block collision would be a pair of IVs $(h_0, h_0^\star)$ and a pair of messages $(m_0, m_0^\star)$ satisfying $\mathsf{AES}_{m_0}(h_0) \oplus \mathsf{AES}_{m_0^\star}(h_0^\star) = h_0 \oplus h_0^\star$.

The trail found by Biryukov *et al.* [2] can be cut into a 10 round trail (Figure 5) used for free-start collision attack on AES-256-DM. The time complexity is 2^{49} for outbound containing (7, 1) Type-I and Type-II active S-boxes in classical setting and $2^{25.61}$ in quantum setting.

6.2 AES-DBL Mode

The triangulation technique can also be adapted to the free-start collision attack on 7-round AES-192-HCF, which is Hirose's double block length hashing mode instantiating AES-192.Let $F : \{0,1\}^{256} \times \{0,1\}^{64} \to \{0,1\}^{256}$ be AES-192-HCF compress function such that $(g_i, h_i) = F(g_{i-1}, h_{i-1}, m_i)$, and

$$g_i = \mathsf{AES}_{h_{i-1}\|m_i}(g_{i-1}) \oplus g_{i-1}$$
$$h_i = \mathsf{AES}_{h_{i-1}\|m_i}(g_{i-1} \oplus c) \oplus g_{i-1} \oplus c$$

where $\|$ represents the concatenation and $c \in \mathbb{F}_2^{128}$ is a non-zero constant. One block collision of this hash could be found using the same method from [22], *i.e.*, considering colliding pair (g_0, h_0, m_1) and $(g_0^\star, h_0, m_1)$ with $g_0 \oplus g_0^\star = c$, which leads condition $\mathsf{AES}_{h_0\|m_1}(g_0) \oplus \mathsf{AES}_{h_0\|m_1}(g_0 \oplus c) = c$.

The attack takes advantage of using a valid differential characteristic in the middle while the rest of the trail remains truncated (Figure 5), leads to the saving of above 2^8 iterations to find the correct pair values in the classic rebound-based attack. Along with at least 34 degrees of freedom from active S-boxes (each active S-boxes contributes at least 2 admissible values) in the Super-Inbound phase, we can obtain $2^{34} \cdot 2^{8\times5} = 2^{74}$ pair values, which is enough to fulfill the probability 2^{-64} of the feed-forward cancellation $\varDelta_{IN} = \varDelta_{OUT}$ (2^{-32}) and condition $\varDelta h_0 = c$ (2^{-32}), where c has 4 non-zero bytes at some specific positions. The overall time complexity of the attack is 2^{64} in classical setting and $2^{33.37}$ in quantum setting.

7 Conclusions

In this paper, we proposed the quantum multi-collision distinguishers. Our model shows differential paths with probability as low as 2^{-n} will be useful in mounting such attacks, hence resulted in more rounds than both quantum collision attack and classic multi-collision distinguishers. We applied the attack model to AES-like ciphers including all three versions of AES, 4 versions of Rijndael, and the post-quantum block cipher design Saturnin-256. Full round distinguishing attacks are mounted on AES-192, AES-256, Rijndael-128-160, and Rijndael-128-224. Comparing with quantum collision attacks, our attack covered one more round on AES-128. We also applied the same techniques to AES-hashing modes including AES-256-DM and AES-192-HCF. Our attack framework is generic, hence can be applied to more target ciphers.

Acknowledgments

We would like to thank the anonymous reviewers of ACISP for their valuable comments and suggestions. This research is partially supported by the National Key R&D Program of China (Grants No. 2022YFB2701900) and Ministry of Education in Singapore under Grants RG93/23. Zhenzhen Bao is supported by the National Key R&D Program of China (No. 2023YFA1011200) and the National Natural Science Foundation of China (No. 62372262).

References

1. Baek, S., Cho, S., Kim, J.: Quantum cryptanalysis of the full aes-256-based davies–meyer, hirose and mjh hash functions. Quantum Information Processing 21(5), 1–32 (2022)
2. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and Related-Key Attack on the Full AES-256. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2009)
3. Biryukov, A., Nikolic, I.: Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010. LNCS, vol. 6110, pp. 322–344. Springer, Heidelberg, Germany, French Riviera (May 30 – Jun 3, 2010)
4. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random Oracles in a Quantum World. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011)
5. Brassard, G., Høyer, P., Tapp, A.: Quantum Cryptanalysis of Hash and Claw-Free Functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998: Theoretical Informatics, 3rd Latin American Symposium. LNCS, vol. 1380, pp. 163–169. Springer, Heidelberg, Germany, Campinas, Brazil (Apr 20–24, 1998)
6. Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T., Schrottenloher, A.: A note on related-key attacks on Saturnin. https://project.inria.fr/saturnin/files/2020/11/Note-RK-1.pdf (2020)
7. Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T., Schrottenloher, A.: Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. IACR Transactions on Symmetric Cryptology 2020(S1), 160–207 (2020)

8. Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 211–240. Springer, Heidelberg, Germany, Hong Kong, China (Dec 3–7, 2017)
9. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002), https://doi.org/10.1007/978-3-662-04722-4
10. Dong, X., Guo, J., Li, S., Pham, P.: Triangulating rebound attack on aes-like hashing. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13507, pp. 94–124. Springer (2022), https://doi.org/10.1007/978-3-031-15802-5_4
11. Dong, X., Li, S., Pham, P.: Chosen-key distinguishing attacks on full aes-192, aes-256, kiasu-bc, and more. IACR Cryptol. ePrint Arch. p. 1095 (2023), https://eprint.iacr.org/2023/1095
12. Dong, X., Sun, S., Shi, D., Gao, F., Wang, X., Hu, L.: Quantum Collision Attacks on AES-Like Hashing with Low Quantum Random Access Memories. In: Advances in Cryptology – ASIACRYPT 2020, Part II. pp. 727–757. LNCS, Springer, Heidelberg, Germany (Dec 2020)
13. Fouque, P.A., Jean, J., Peyrin, T.: Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 183–203. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013)
14. Gérault, D., Lafourcade, P.: Related-Key Cryptanalysis of Midori. In: Dunkelman, O., Sanadhya, S.K. (eds.) Progress in Cryptology - INDOCRYPT 2016: 17th International Conference in Cryptology in India. LNCS, vol. 10095, pp. 287–304. Springer, Heidelberg, Germany, Kolkata, India (Dec 11–14, 2016)
15. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996), https://doi.org/10.1145/237814.237866
16. Gérault, D., Lafourcade, P., Minier, M., Solnon, C.: Computing aes related-key differential characteristics with constraint programming. Artificial Intelligence 278, 103183 (2020), https://www.sciencedirect.com/science/article/pii/S0004370218303631
17. Gérault, D., Minier, M., Solnon, C.: Constraint programming models for chosen key differential cryptanalysis. In: Rueher, M. (ed.) Principles and Practice of Constraint Programming. pp. 584–601. Springer International Publishing, Cham (2016)
18. Hosoyamada, A., Sasaki, Y.: Finding Hash Collisions with Quantum Computers by Using Differential Trails with Smaller Probability than Birthday Bound. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 249–279. Springer, Heidelberg, Germany, Zagreb, Croatia (May 10–14, 2020)
19. Hosoyamada, A., Sasaki, Y.: Quantum Collision Attacks on Reduced SHA-256 and SHA-512. Cryptology ePrint Archive, Report 2021/292 (2021), https://eprint.iacr.org/2021/292
20. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking Symmetric Cryptosystems Using Quantum Period Finding. In: Robshaw, M., Katz, J. (eds.)

Advances in Cryptology – CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 207–237. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016)

21. Khovratovich, D., Biryukov, A., Nikolic, I.: Speeding up Collision Search for Byte-Oriented Hash Functions. In: Fischlin, M. (ed.) Topics in Cryptology – CT-RSA 2009. LNCS, vol. 5473, pp. 164–181. Springer, Heidelberg, Germany, San Francisco, CA, USA (Apr 20–24, 2009)
22. Kumar Chauhan, A., Kumar, A., Kumar Sanadhya, S.: Quantum free-start collision attacks on double block length hashing with round-reduced aes-256. IACR Transactions on Symmetric Cryptology 2021(1), 316–336 (Mar 2021), https://tosc.iacr.org/index.php/ToSC/article/view/8841
23. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. pp. 312–316 (01 2012)
24. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: 2010 IEEE International Symposium on Information Theory. pp. 2682–2685 (2010)
25. Leander, G., May, A.: Grover Meets Simon - Quantumly Attacking the FX-construction. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 161–178. Springer, Heidelberg, Germany, Hong Kong, China (Dec 3–7, 2017)
26. Li, S., Liu, G., Pham, P.: Rebound attacks on sfskinny hashing with automatic tools. In: Yuan, X., Bai, G., Alcaraz, C., Majumdar, S. (eds.) Network and System Security - 16th International Conference, NSS 2022, Denarau Island, Fiji, December 9-12, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13787, pp. 649–666. Springer (2022), https://doi.org/10.1007/978-3-031-23020-2_37
27. Liu, Q., Zhandry, M.: On Finding Quantum Multi-collisions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 189–218. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019)
28. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In: Dunkelman, O. (ed.) Fast Software Encryption – FSE 2009. LNCS, vol. 5665, pp. 260–276. Springer, Heidelberg, Germany, Leuven, Belgium (Feb 22–25, 2009)
29. National Institute for Standards and Technology, USA: Post-Quantum Cryptography Standardization (2017), https://csrc.nist.gov/projects/post-quantum-cryptography
30. Patarin, J.: A Proof of Security in O(2n) for the Xor of Two Random Permutations. In: Safavi-Naini, R. (ed.) ICITS 08: 3rd International Conference on Information Theoretic Security. LNCS, vol. 5155, pp. 232–248. Springer, Heidelberg, Germany, Calgary, Canada (Aug 10–13, 2008)
31. Pollard, J.M.: A monte carlo method for factorization. BIT Numerical Mathematics 15(3), 331–334 (1975), https://doi.org/10.1007/BF01933667
32. Shor, P.W.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society Press, Santa Fe, NM, USA (Nov 20–22, 1994)
33. Simon, D.R.: On the Power of Quantum Computation. SIAM J. Comput. 26(5), 1474–1483 (Oct 1997), https://doi.org/10.1137/S0097539796298637
34. Sun, S., Gerault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., Hu, L.: Analysis of AES, SKINNY, and Others with Constraint Programming. IACR Transactions on Symmetric Cryptology 2017(1), 281–306 (2017)

35. Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K.: Birthday paradox for multi-collisions. In: International Conference on Information Security and Cryptology. pp. 29–40. Springer (2006)
36. Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K.: Birthday Paradox for Multi-collisions. In: Rhee, M.S., Lee, B. (eds.) ICISC 06: 9th International Conference on Information Security and Cryptology. LNCS, vol. 4296, pp. 29–40. Springer, Heidelberg, Germany, Busan, Korea (Nov 30 – Dec 1, 2006)
37. van Oorschot, P.C., Wiener, M.J.: Parallel Collision Search with Cryptanalytic Applications. Journal of Cryptology 12(1), 1–28 (Jan 1999)

Supplementary Material

A Steps to reduce k_2

Fixed bytes: $k_0[14], k_1[15], k_2[14], k_3[15], k_5[15], y_1[1], y_1[5], y_1[10], y_1[11], y_1[15],$ $y_2[2], y_2[6], y_2[10], y_2[11], y_2[13], y_2[15], y_3[1], y_3[3], y_3[9], y_3[11]$	
Free bytes: $k_0[15], k_3[11], k_4[4], k_5[11], y_1[4], y_1[7], y_1[8], y_1[9], y_1[12], y_1[13], y_1[14], y_2[5]$	
1. $k_1[11] = k_0[15] \oplus k_1[15]$	27. $k_2[3] = S(k_2[12]) \oplus k_3[3]$
2. $k_2[15] = k_3[11] \oplus k_3[15]$	28. $k_1[14] = k_2[10] \oplus k_2[14]$
3. $k_4[15] = k_5[11] \oplus k_5[15]$	29. $k_1[10] = k_0[14] \oplus k_1[14]$
4. $k_4[11] = k_3[15] \oplus k_4[15]$	30. $k_2[6] = k_1[10] \oplus k_2[10]$
5. $k_4[7] = k_4[11] \oplus k_3[11]$	31. $x_1[6] = x_2[6] \oplus k_2[6]$
6. $k_5[7] = k_5[11] \oplus k_4[11]$	32. $y_1[3] = z_1[7], w_1[4,5,7]$ $= MC^{-1}(z_1[4,5,6], w_1[6])$
7. $k_5[3] = k_5[7] \oplus k_4[7]$	33. $k_2[5] = w_1[5] \oplus x_2[5]$
8. $w_1[15] = x_2[15] \oplus k_2[15]$	34. $x_2[7] = w_1[7] \oplus k_2[7]$
9. $y_1[6] = z_1[14], w_1[12,13,14]$ $= MC^{-1}(z_1[12,13,15], w_1[15])$	35. $z_2[8], w_2[8,9,10]$ $= MC^{-1}(z_2[9,10,11], w_2[11])$
10. $k_2[13] = x_2[13] \oplus w_1[13]$	36. $k_2[8] = x_2[8] \oplus w_1[8]$
11. $x_2[14] = k_2[14] \oplus w_1[14]$	37. $k_3[9] = w_2[9] \oplus x_3[9]$
12. $w_2[11] = x_3[11] \oplus k_3[11]$	38. $k_3[4] = k_2[8] \oplus k_3[8]$
13. $k_2[11] = k_1[15] \oplus k_2[15]$	39. $k_4[0] = k_3[4] \oplus k_4[4]$
14. $k_2[7] = k_1[11] \oplus k_2[11]$	40. $k_3[13] = k_3[9] \oplus k_2[13]$
15. $k_3[7] = k_2[11] \oplus k_3[11]$	41. $k_3[0] = S(k_3[13]) \oplus k_4[0]$
16. $k_3[3] = k_2[7] \oplus k_3[7]$	42. $k_2[4] = k_3[0] \oplus k_3[4]$
17. $k_4[3] = k_3[7] \oplus k_4[7]$	43. $k_2[0] = S(k_2[13]) \oplus k_3[0]$
18. $k_4[12] = S^{-1}(k_4[3] \oplus k_5[3])$	44. $w_2[3] = k_3[3] \oplus x_3[3]$
19. $k_3[12] = S^{-1}(k_3[3] \oplus k_4[3])$	45. $z_2[0], w_2[0,1,2]$ $= MC^{-1}(z_2[1,2,3], w_2[3])$
20. $k_4[8] = k_3[12] \oplus k_4[12]$	46. $w_1[0] = k_2[0] \oplus x_2[0]$
21. $k_3[8] = k_4[4] \oplus k_4[8]$	47. $y_1[0] = z_1[0], w_1[1,2,3]$ $= MC^{-1}(z_1[1,2,3], w_1[0])$
22. $k_2[12] = k_3[8] \oplus k_3[12]$	48. $k_2[2] = w_1[2] \oplus x_2[2]$
23. $w_1[11] = k_2[11] \oplus x_2[11]$	49. $k_3[1] = w_2[1] \oplus x_3[1]$
24. $y_1[2] = z_1[10], w_1[8,9,10]$ $= MC^{-1}(z_1[8,911], w_1[11])$	50. $k_2[1] = k_3[1] \oplus S(k_2[14])$
25. $k_2[10] = w_1[10] \oplus x_2[10]$	51. $k_3[5] = k_2[5] \oplus k_3[1]$
26. $x_2[12] = w_1[12] \oplus k_2[12]$	52. $k_2[9] = k_3[5] \oplus k_3[9]$

Table 7: Steps to derive the entire key k_2 (marked in red) and state y_1 (marked in blue) from the fixed and free bytes.

B The differential trails of our attacks

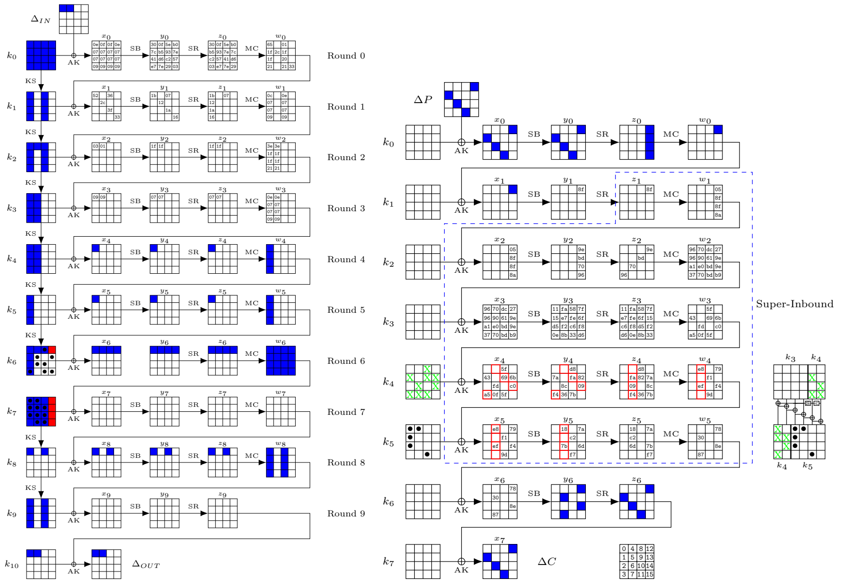


Figure 5: Left: A differential trail for free-start collision attack on 10-round AES-256-DM. Right: A differential trail for quantum free-start collision attack on 7-round AES-192-HCF.

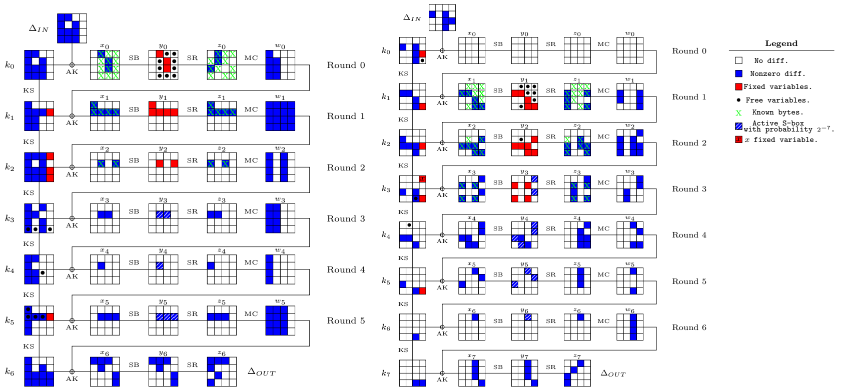


Figure 6: Left: A differential trail for 7-Round AES-128. Right: A differential trail for 8-Round AES-128 with variable x.

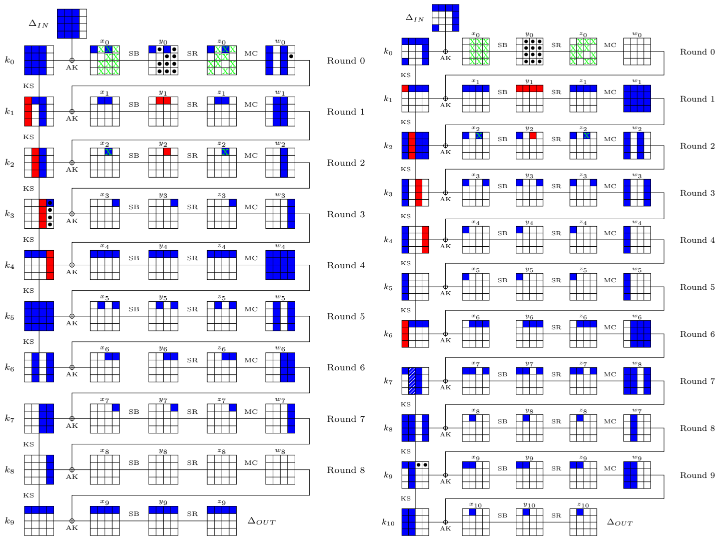
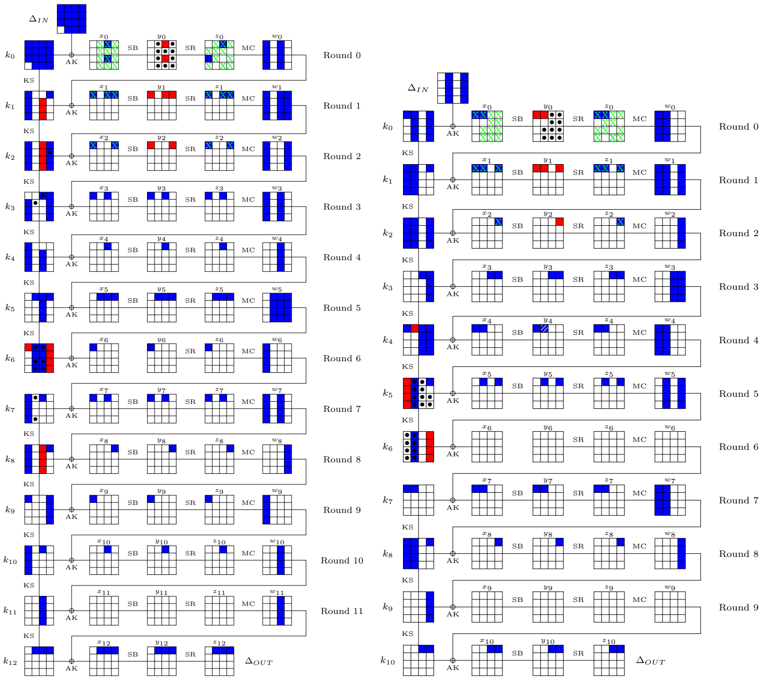


Figure 7: Left: A differential trail for 10-Round Rijndael-128-160. Right: A differential trail for 11-Round Rijndael-128-160.



Δ_{IN}
k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10} k_{11} k_{12}
KS
AK
x_0 SB y_0 SR z_0 MC w_0
Round 0
Round 1
Round 2
Round 3
Round 4
Round 5
Round 6
Round 7
Round 8
Round 9
Round 10
Round 11
Δ_{OUT}

Figure 8: Left: A differential trail for 13-Round Rijndael-128-224. Right: A differential trail for 11-Round Rijndael-128-224.

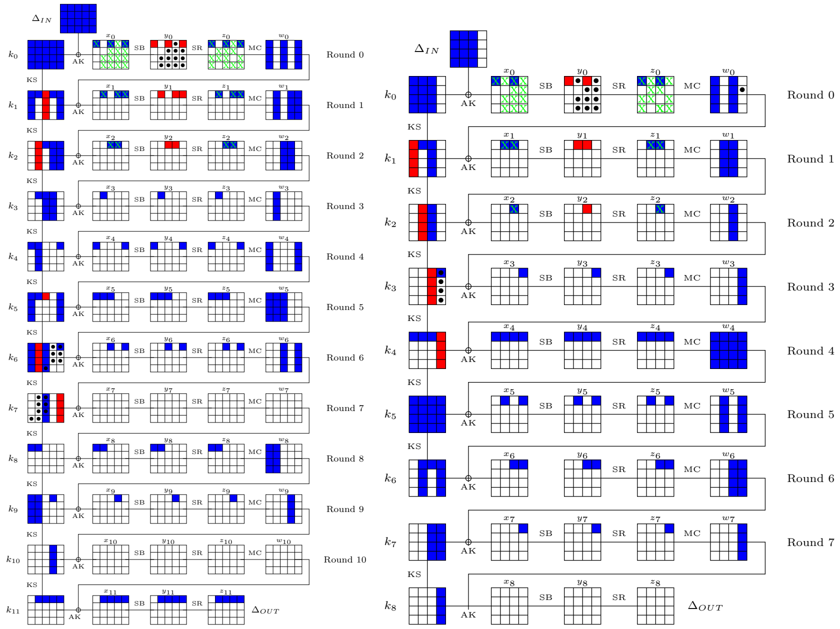


Figure 9: Left: A differential trail for 12-Round Rijndael-160-256. Right: A differential trail for 9-Round Rijndael-128-160.

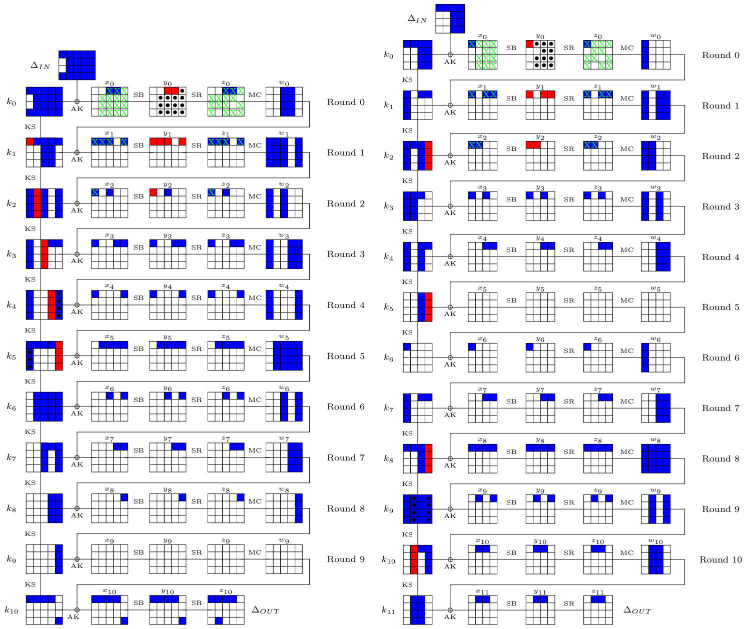


Figure 10: Left: A differential trail for 11-Round Rijndael-160-192. Right: A differential trail for 12-Round AES-192.