Quantum Multicollision-Finding Algorithm

Akinori Hosoyamada[1], Yu Sasaki[1], and Keita Xagawa[1]

NTT Secure Platform Laboratories,
3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585, Japan.
{hosoyamada.akinori,sasaki.yu,xagawa.keita}@lab.ntt.co.jp

Abstract. The current paper presents a new quantum algorithm for finding multicollisions, often denoted by l-collisions, where an l-collision for a function is a set of l distinct inputs having the same output value. Although it is fundamental in cryptography, the problem of finding multicollisions has not received much attention *in a quantum setting*. The tight bound of quantum query complexity for finding 2-collisions of random functions has been revealed to be $\Theta(N^{1/3})$, where N is the size of a codomain. However, neither the lower nor upper bound is known for l-collisions. The paper first integrates the results from existing research to derive several new observations, e.g. l-collisions can be generated only with $O(N^{1/2})$ quantum queries for a small constant l. Then a new quantum algorithm is proposed, which finds an l-collision of any function that has a domain size l times larger than the codomain size. A rigorous proof is given to guarantee that the expected number of quantum queries is $O\left(N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}\right)$ for a small constant l, which matches the tight bound of $\Theta(N^{1/3})$ for $l = 2$ and improves the known bounds, say, the above simple bound of $O(N^{1/2})$.

keywords: post-quantum cryptography, multicollision, quantum algorithm, Grover, BHT, rigorous complexity evaluation, state-of-art

1 Introduction

Finding collisions or multicollisions is a fundamental problem in theoretical computer sciences and one of the most critical problems especially in cryptography. For given finite sets X and Y with $|Y| = N$, and a function $H\colon X \to Y$, an l-collision finding problem is to find a set of l distinct inputs $x_1, \ldots, x_l$ such that $H(x_1) = \cdots = H(x_l)$. Both upper and lower bounds of query and time complexity of the l-collision finding problem are fundamental and have several applications in cryptography.

Applications of multicollisions. We often use the lower bound of query complexity (or the upper bound of the success probability) to prove the security of cryptographic schemes. Let us consider a cryptographic scheme based on Pseudo-Random Functions (PRFs). In the security proof, we replace the PRFs with truly random functions (or random oracles) and show the security of the scheme with the random oracles by information-theoretic arguments. In the latter security arguments, we often use *the lower bound of queries* for finding multicollisions of random functions. For example, Chang

and Nandi [CN08] proved the indifferentiability of the chopMD hash function construction; Jaulmes, Joux, and Valette [JJV02] proved the indistinguishability of RMAC; Hirose *et al.* [HIK+10] proved the indifferentiability of the ISO standard lightweight hash function Lesamnta-LW; Naito and Ohta [NO14] improved the indifferentiability of PHOTON and Parazoa hash functions; and Javanovic, Luykx, and Mennink [JLM14] greatly improved the security lower bounds of authenticated-encryption mode of Keyed-Sponge. The upper bound of the probability to obtain multicollisions after q queries plays an important role in their proofs.

In addition, studying and improving the upper bound for the l-collision finding problem also help our understanding, which often leads to the complexity of generic attacks. For example, l-collisions are exploited in the collision attack on the MDC-2 hash function construction by Knudsen *et al.* [KMRT09], the preimage attack on the JH hash function by Mendel and Thomsen [MT08], the internal state recovery attack on HMAC by Naito *et al.* [NSWY13], the key recovery attack on iterated Even-Mansour by Dinur *et al.* [DDKS14], and the key recovery attack on LED block cipher by Nikolić, Wang, and Wu [NWW13].

Furthermore, multicollisions also have applications in protocols. An interesting example is a micro-payment scheme, MicroMint [RS96]. Here, a coin is a bit-string the validity of which can be easily checked but hard to produce. In MicroMint, coins are 4-collisions of a function. If 4-collisions can be produced quickly, a malicious user can counterfeit coins.

Existing results for multicollisions in classical setting. The problem of finding (multi-)collisions has been extensively discussed in the classical setting. Suppose that we can access the function H in the *classical* query; that is, we can send $x \in X$ to the oracle H and obtain $y \in Y$ as $H(x)$. For a random function H, making q queries to H can find the collision of H with probability at most q^2/N. The birthday bound shows when $q \approx N^{1/2}$, we obtain a collision with probability $1/2$. This can be extended to the l-collision case. Suzuki, Tonien, Kurosawa, and Toyota [STKT08] showed that with $N^{(l-1)/l}$ queries the probability of finding an l-collision is upper bounded by $1/l!$ and lower bounded by $1/l! - 1/2(l!)^2$, which shows that the query complexity can be approximated to $N^{(l-1)/l}$ for a small constant l. To be more precise, it is shown that $O((l!)^{1/l}N^{(l-1)/l})$ evaluations of the function H finds an l-collision with probability about 1/2 if $H\colon X \to Y$ is a random function.

The above argument only focuses on the number of queries. To implement the l-collision finding algorithm, the computational cost, T, and the memory amount, S, or their tradeoff should be considered. The simple method needs to store all the results of the queries. Hence, it requires $T = S = N^{1/2}$ for collisions and $T = S = O(N^{(l-1)/l})$ for l-collisions. The collision finding algorithm can be made memoryless by using Floyd's cycle detecting algorithm [Flo67]. However, no such memoryless algorithm is known for l-collisions, thus the researcher's goal is to achieve better complexity with respect to $T \times S$ or to trade T and S for a given $T \times S$.

An l-collision can be found with $T = l \cdot N$ and $S = O(1)$ by running a brute-force preimage attack l times for a fixed target. Although this method achieves better $T \times S$ than the simple method, it cannot trade T for S. Joux and Lucks [JL09] discovered

the 3-collision finding algorithm with $T = N^{1-\alpha}$ and $S = N^{\alpha}$ for $\alpha < 1/3$ by using the parallel collision search technique. Nikolić and Sasaki [NS16] achieved the same complexity as Joux and Lucks by using an unbalanced meet-in-the-middle attack.

1.1 Collisions and Multicollisions in Quantum Setting

Algorithmic speedup using quantum computers has been actively discussed recently. For example, Grover's seminal result [Gro96] attracted cryptographers' attention because of the quantum speedup of database search. Given a function $F\colon X \to \{0,1\}$ such that there exists a unique $x_0 \in X$ that satisfies $F(x_0) = 1$, Grover's algorithm finds x_0 in $O(|X|^{1/2})$ queries.

This paper discusses the complexity of quantum algorithms in *the quantum query model*. In this model, a function H is given as a black box, and the complexity of quantum algorithms is measured as the number of quantum queries to H. A quantum query model is widely adopted, and previous studies on finding collisions in the quantum setting follow this model [BHT97,Amb07,Bel12,Yue14,Zha15].

Previous research on finding collisions and multicollisions can be classified with respect to two types of dichotomies.

Domain size and codomain size. The domain size and codomain size of the function $H\colon X \to Y$ is a sensitive problem for quantum algorithms. Some quantum algorithms aim to find collisions and multicollisions of H with $|X| \geq |Y|$, while others target H with $|X| < |Y|$. The former algorithms can be directly applied to find collisions and multicollisions of real *hash functions* such as SHA-3. The latter ones mainly target *database search* rather than hash functions. The (multi-)collision search on database can still be converted for hash functions, but it generally requires a huge complexity increase. (On the other hand, the (multi-)collision search for hash functions with $|X| \geq |Y|$ cannot be converted for a database with $|X| < |Y|$.)
Hereafter, we use "H" and "D" to denote the cases with $|X| \geq |Y|$ and $|X| < |Y|$, respectively. We note that our goal is finding a new multicollision algorithm that can be applied to real hash functions, namely the H setting.

Random function and any function. Both in classical and quantum settings, existing algorithms often assume randomness: they can find collisions only on average when H is chosen uniformly at random from $\mathrm{Map}(X,Y) := \{f \mid f\colon X \to Y\}$. If an algorithm finds collisions of *any* function $H \in \mathrm{Map}(X,Y)$, it also finds collisions of randomly chosen functions. Hence algorithms applied to any function are stronger than ones only applied to a random function. Hereafter, we use "Rnd" and "Arb" to denote the cases in which H is chosen uniformly at random and H is chosen arbitrarily, respectively. We note that the Rnd setting is sufficient for our goal and will show that our new algorithm can be applied to the Arb setting.

In the following, we revisit the existing results of collision and multicollision-finding algorithms in the quantum setting.

– Brassard, Høyer, and Tapp [BHT97] proposed a quantum algorithm **BHT**, which can be classified as H-Arb for 2-collisions. To be more precise, **BHT** finds a 2-collision of any l-to-one function with $O(N^{1/3})$ quantum queries and a memory amount of $O(N^{1/3})$.

Table 1: Summary of existing quantum algorithms to find (multi-)collisions.

	Random function "Rnd"	Arbitrary function "Arb"
Database "D"	Zhandry + Ambainis (2-col)	Ambainis (l-col) Belovs (l-col)
Hash "H"	Zhandry + Ambainis (2-col) Yuen (2-col)	BHT (2-col) Converted Ambainis (l-col) Converted Belovs (l-col) **Ours** (l-col)

- Ambainis [Amb07] studied an element distinctness problem rather than the collision finding problem, but his algorithm can be directly applied to find (multi)collisions of functions. The algorithm is for D-Arb for l-collisions with $O(M^{l/(l+1)})$ quantum queries, where M is the domain size.
- Belovs [Bel12] improved the complexity of Ambainis' algorithm [Amb07].
- Zhandry [Zha15] observed that Ambainis' algorithm [Amb07] can be modified to H-Rnd for 2-collisions with $O(N^{1/3})$ quantum queries, when $|X| = \Omega(N^{1/2})$ and $N = |Y|$.
- Yuen [Yue14] discussed the application of **BHT** when $|X| = |Y|$ and the target function H is weakened to Rnd. The complexity is $O(N^{1/3})$ quantum queries. We do not discuss its details because the discussed case in Yuen's work [Yue14] is a subset of Zhandry's extension of Ambainis' algorithm.
- Regarding the lower bound, $O((N/l)^{1/3})$ of **BHT** to find 2-collisions against l-to-one function was proved to be tight by several researchers [AS04,Amb05,Kut05]. Zhandry proved that $O(N^{1/3})$ for 2-collisions against random function is tight. That is, any quantum algorithm that finds a 2-collision against a random function requires $\Omega(N^{1/3})$ quantum queries [Zha15].[1] Obviously, $\Omega(N^{1/3})$ can also be a lower bound for $l > 2$, but no advanced lower bound is known for $l > 2$. Hülsing, Rijneveld, and Song [HRS16] studied *quantum generic security* of hash functions by considering quantum query complexity in the quantum random-oracle model. They successfully showed the upper and lower bound of *quantum query complexity* to solve the *one-wayness*, *second-preimage resistance*, *extended target-collision resistance*, and their variants.[2] Unfortunately, they did not treat collision and multicollision resistances.

The classifications of the existing algorithms are shown in Table 1. As mentioned earlier, Ambainis' algorithm [Amb07] and its improvement by Belovs [Bel12] originally focused on the database search, but they can be converted into the hash function setting with extra complexity. However, all the other approaches for the hash function setting only analyze 2-collisions. Hence, we can conclude that no quantum algorithm exists that is optimized to find l-collisions for hash functions.

[1] Zhandry showed that any quantum algorithm with q quantum queries finds a 2-collision with probability at most $O((q+2)^3/N)$ [Zha15].

[2] For example, they showed that any quantum algorithm with q quantum queries finds a preimage with probability at most $O((q+1)^2/N)$ [HRS16].

1.2 Our Contributions

In this paper, we study quantum algorithms to find l-collisions against a function $H\colon X \to Y$.

First, the problem of finding l-collisions against hash functions has not received much attention in the literature. Even if the previous work can be directly applied to l-collisions against hash functions, nobody has considered this problem and no generic attack is known. This motivates us to provide a systematization of knowledge about existing quantum algorithms. Namely, we, for the first time in this field, provide the state of the art of the complexity of finding l-collisions against hash functions with a direct application, trivial extension, and simple combination of existing results.

This state of the art sheds light on the problems that require further investigation. For the second but main contribution of this paper, we present a new quantum algorithm to find l-collisions against hash functions.

Our contributions in each part are detailed below.

Systematization of knowledge (combination of existing results).

- Our first observation is that, when H is a random function and $|X| = l|Y|$ for a small constant l, the query complexity of the l-collision finding problem is lowered to $O(N^{1/2})$ by simply applying Grover's algorithm. Hence, any meaningful generic attack in the quantum setting must achieve the query complexity below $O(N^{1/2})$. Intuitively, a preimage of the hash value can be generated with $O(N^{1/2})$ queries in the quantum setting and l-collisions are generated by generating l preimages. This corresponds to the upper bound of $O(N)$ complexity in the classical setting. (Note that this upper bound is for the Rnd setting and does not hold for the Arb setting.)
- The above observation is quite straightforward but useful to measure the effect of other attacks. For example, Ambainis' l-collision search for database [Amb07] can be converted for hash functions with $O(M^{l/(l+1)})$ complexity where M is the domain size. However, this cannot be below $O(N^{1/2})$ for any l. The same applies to the improvement by Belovs [Bel12]. Those converted algorithms can be meaningful only in the Arb setting.
- Zhandry [Zha15] discussed the application of Ambainis' l-collision search in H-Rnd and D-Rnd only for $l = 2$, although it can trivially be extended to $l > 2$. If it is extended, the complexity for $l = 3$ reaches $O(N^{1/2})$. Thus, Zhandry's idea only works for $l = 2$.
- Zhandry [Zha15] considered Ambainis' l-collision search rather than Belovs' improvement [Bel12]. If we consider Zhandry + Belovs, the complexity in H-Rnd for $l = 3$ becomes $O(N^{10/21})$, which is faster than the simple application of Grover's algorithm. Thus, it is a meaningful generic attack. For $l \geq 4$, the complexity of Zhandry + Belovs reaches $O(N^{1/2})$.
- In summary, for the Rnd setting, the tight algorithm with $O(N^{1/2})$ complexity exists for $l = 2$. There is a better generic attack than the simple application of Grover's algorithm for $l = 3$, although the lower bound is unknown. For $l \geq 4$, there is no known algorithm better than the application of Grover's algorithm, and the lower bound is also unknown. For the Arb setting, direct application of Belovs' algorithm is the existing best attack.

New quantum multicollision-finding algorithm.

- Given the above state of the art, our main contribution is a new l-collision finding algorithm with $O\left(N^{(3^{l-1}-1)/2\cdot 3^{l-1}}\right)$ quantum queries against an arbitrary function $H\colon X \to Y$ with $|X| = l|Y|$. By applying this algorithm in the Rnd setting, we achieve a speedup compared with the simple upper bound of $O(N^{1/2})$ for any l. The complexity of our algorithm matches the tight bound of $O(N^{1/3})$ for $l = 2$ and is faster than $O(10/21)$ of Zhandry + Belovs for $l = 3$. The complexity of our algorithm for a small constant l is shown in Table 2. The complexities are compared in Fig. 1.
- Unlike other algorithms for Arb, our algorithm asymptotically approaches to $O(N^{1/2})$ as l increases. The previous results by Ambainis [Amb07] asymptotically approaches to $O(M)$, and Belovs [Bel12] asymptotically approaches to $O(M^{3/4})$, respectively, where $M = |X|$. Our algorithm improves these results for $M \geq l \cdot N$. The complexities are compared in Fig. 2 for $M = l \cdot N$.
- The core idea of our algorithm is a sophisticated combination of the 3-collision algorithm in the classical setting by Joux and Lucks [JL09] and the generalized Grover algorithm for the quantum setting [BBHT98].
  In short, we recursively call a collision finding algorithm and Grover's algorithm. For example, to generate 3-collisions, we first iterate the 2-collision finding algorithm of $O(N^{1/3})$ complexity $O(N^{1/9})$ times. Then, we search for the preimage of one of $O(N^{1/9})$ 2-collisions by using Grover's algorithm, which runs with $O(N^{4/9})$ complexity. To generate 4-collisions, we iterate the 3-collision finding algorithm of $O(N^{4/9})$ complexity $O(N^{1/27})$ times, then search for the preimage of one of $O(N^{1/27})$ 3-collisions with $O(N^{13/27})$ complexity.
  In classical setting, the recursive application of the algorithm of [JL09] has never been discussed in literature. This is because the resulting complexity easily exceeds the information theoretically upper bound of $O(N^{(l-1)n/l})$. In contrast, no such upper bounds are known in quantum complexity, thus we can obtain advantages with the recursive application.
- Finally, we provide a rigorous complexity evaluation of our algorithm, which is another main focus of this paper. The point of our proof is that lower and upper bounding the number of collisions of H is necessary for lower bound success probability. Our evaluation suggests that our algorithm finds a 2-collision of SHA3-512 with 2^{179} quantum queries and finds a 3-collision with 2^{238} quantum queries.

Table 2: Quantum query complexity of our l-collision finding algorithm. Query denotes $\log_N(\text{query})$, which asymptotically approaches $1/2$ as l increases.

l	2	3	4	5	6	7	8
Query	$\frac{1}{3}$	$\frac{4}{9}$	$\frac{13}{27}$	$\frac{40}{81}$	$\frac{121}{243}$	$\frac{364}{729}$	$\frac{1093}{2187}$

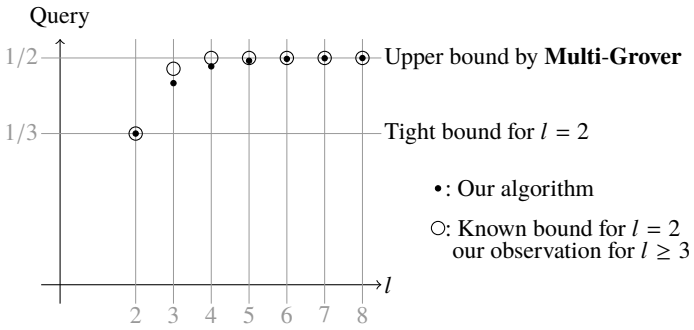

Fig. 1: Quantum query complexity needed to find l-collision in H-Rnd setting. Query denotes $\log_N(\text{query})$.

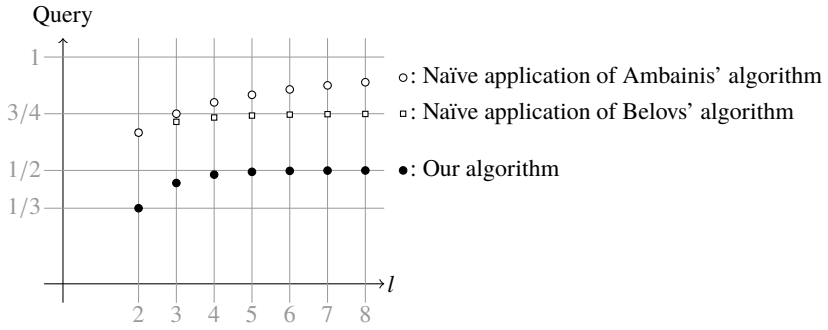

Fig. 2: Quantum query complexity for finding an l-collision in H-Arb setting. Query denotes $\log_N(\text{query})$.

2 Preliminaries

Notation. We define *l-collision* as follows.

Definition 2.1 (l-collision). *Let l be a positive integer. Let X and Y be finite spaces. Let $H\colon X \to Y$ be a function from X to Y. Let $\{x_1, x_2, \ldots, x_l\}$ be a subset of X and let y be an element of Y. We define $(\{x_1, x_2, \ldots, x_l\}, y)$ as an l-collision of H if the pair satisfies $x_i \neq x_j$ for $i \neq j$ and $y = H(x_1) = H(x_2) = \cdots = H(x_l)$. Two l-collisions $c = (\{x_1, x_2, \ldots, x_l\}, y)$ and $c' = (\{x'_1, x'_2, \ldots, x'_l\}, y')$ are said to be equal if and only if $\{x_1, x_2, \ldots, x_l\} = \{x'_1, x'_2, \ldots, x'_l\}$ as sets and $y = y'$.*

If $|X| = l \cdot |Y|$ and a function $H\colon X \to Y$ satisfies $|H^{-1}(H(x))| = l$ for any $x \in X$, we call H *l-to-one function*. If l is clear by context, we simply call H *regular function*.

Complexity of quantum algorithm. Suppose that we are given a function H as a black box and can query *a quantum state* to the function H; that is, we can send a quantum superposition, say, $\sum_{x\in X} \alpha_x |x\rangle |b\rangle$ to the oracle H and obtain $\sum_{x\in X} \alpha_x |x\rangle |b \oplus H(x)\rangle$. In the *quantum query model*, the complexity of a quantum algorithm is measured by the number of quantum queries to H that the algorithm makes. Many existing studies on collision problems in quantum setting follow this model [BHT97,Amb07,Bel12,Zha15], and the quantum query complexity of collision problems must be understood when we make security proofs in the *quantum random oracle model* [BDF+11], which corresponds to the *random oracle model* in a classical setting. As for time complexity, we will discuss it in Section 6.

In the rest of the paper, we assume that readers already have sufficient basic knowledge about the quantum circuit model and omit a detailed explanation of it.

2.1 Grover's Algorithm and Its Generalization

Grover's algorithm [Gro96] was proposed for fast database search in a quantum setting. The problem of database search is modeled as follows:

Problem 2.1 (Quantum Database Search). Suppose that there is a function $F\colon X \to \{0, 1\}$ such that there is only one element $x_0 \in X$ that satisfies $F(x_0) = 1$. The problem is to find x_0 under the condition that we are allowed to access a *quantum* oracle of F.

Grover's algorithm can solve this problem with high probability, making quantum queries to F for roughly $\sqrt{|X|}$ times. This means that the complexity needed for an exhaustive search in a quantum setting is the square root of one in the classical setting. For example, an exhaustive key search against AES-128 will succeed with approximately 2^{64} quantum queries.

The database search problem described above is naturally extended so that F has more than one preimage of 1. A formal description is given below.

Problem 2.2 (Generalized Quantum Database Search). Suppose that there is a function $F\colon X \to \{0, 1\}$ and we are allowed to make quantum queries to F. Then, find x_0 that satisfies $F(x_0) = 1$.

Boyer, Brassard, Høyer, and Tapp proposed a quantum algorithm solving this problem [BBHT98]. The advantage of their algorithm is that it can be applied without knowing the number of $x \in X$ that satisfies $F(x) = 1$ in advance.

Theorem 2.1 ([BBHT98] Theorem 3). *Let X be a finite set and $F\colon X \to \{0,1\}$ be a function. Let $t = |\{x \in X \mid F(x) = 1\}|$. If $t \leq \frac{3}{4}|X|$, there exists a quantum algorithm* **BBHT** *that finds $x \in X$ that satisfies $F(x) = 1$ with an expected number of quantum queries to F at most $9/2 \cdot \sqrt{|X|/t}$, without knowing t in advance. When $t = 0$, this algorithm will never abort.*

The above algorithm **BBHT** is applicable only to the case $t \leq \frac{3}{4}|X|$, but we want an algorithm that is also applicable to the case $t > \frac{3}{4}|X|$. Now we consider the following algorithm $\mathcal{A}$. $\mathcal{A}$ runs **BBHT**, and simultaneously choose random elements from X independently and uniformly at random, and make queries to F. $\mathcal{A}$ makes exactly one query when **BBHT** makes one query, and $\mathcal{A}$ stops at once if it finds $x \in X$ such that $F(x) = 1$. This algorithm $\mathcal{A}$ is also applicable to the case $t > \frac{3}{4}|X|$, and it finds $x \in X$ such that $F(x) = 1$ with an expected number of quantum queries to F at most

$$\max\left\{2 \cdot \frac{9}{2}\sqrt{\frac{|X|}{t}}, 2 \cdot \frac{4}{3}\right\} = 9\sqrt{\frac{|X|}{t}}.$$

We also call this algorithm **BBHT**. Now we have the following corollary.

Corollary 2.1. *Let X be a finite set and $F\colon X \to \{0,1\}$ be a function. Let $t = |\{x \in X \mid F(x) = 1\}|$. There exists a quantum algorithm* **BBHT** *that finds $x \in X$ that satisfies $F(x) = 1$ with an expected number of quantum queries to F at most $9 \cdot \sqrt{|X|/t}$, without knowing t in advance. When $t = 0$, this algorithm will never abort.*

3 Systematization of Knowledge on Quantum Multicollision Algorithms

In the classical setting, l-collision on hash functions can be found with $O(N^{(l-1)/l})$ queries for a small constant l.

However, the problem has not received much attention in the quantum setting. This section surveys previous work and integrates the findings of different researchers to make several new observations on this topic.

3.1 Survey of Previous Work

We review the algorithm **BHT** [BHT97] because our new algorithm explained in Section 4 is an extension of it. We also survey previous studies, classifying them in two types: element l-distinctness problem (D-Arb), and collision finding problem on random functions (D-Rnd and H-Rnd).

BHT: Collision finding problem on l-to-one functions. For simplicity, we describe **BHT** only for the case $l = 2$. Let X, Y be sets that satisfy $|X| = 2 \cdot |Y|$, $|Y| = N$, and $H\colon X \to Y$ be a 2-to-one function.

The basic idea of **BHT** is as follows. First, we choose a parameter k ($k = N^{1/3}$ will turn out to be optimal) and a subset $X' \subset X$ of cardinality k. We then make a list $L = \{(x, H(x))\}_{x \in X'}$. Second, we use the **BBHT** algorithm to find an element $x \in X$ such that there exists $x_0 \in X'$ that satisfies $(x_0, H(x)) \in L$ and $x \neq x_0$, i.e., we try to find a pair $(x_0, H(x_0)) \in L$ that can be extended to a collision $(\{x, x_0\}, H(x_0))$. The precise description of **BHT** is as follows.

Definition 3.1 (BHT(H, k)).

1. *Choose an arbitrary subset $X' \subset X$ of cardinality k.*
2. *Make a list $L = \{(x, H(x))\}_{x \in X'}$ by querying $x \in X'$ to H.*
3. *Sort L in accordance with $H(x)$.*
4. *Check whether L contains a 2-collision, i.e., there exist $(x, H(x)), (y, H(y)) \in L$ such that $x \neq y$ and $H(x) = H(y)$. If so, output the 2-collision $(\{x, y\}, H(x))$. Otherwise proceed to the next step.*
5. *Construct the oracle $F\colon X \to \{0, 1\}$ by defining $F(x) = 1$ if and only if there exists $x_0 \in X'$ such that $(x, H(x_0)) \in L$ and $x \neq x_0$.*
6. *Run* **BBHT**(F) *to find $\tilde{x} \in X'$ such that $F(\tilde{x}) = 1$.*
7. *Find $x_0 \in X'$ that satisfies $H(\tilde{x}) = H(x_0)$ from the list L. Output the 2-collision $(\{\tilde{x}, x_0\}, H(x_0))$.*

This algorithm makes k quantum queries in Step 2 and $O(\sqrt{N/k})$ quantum queries in Step 6 (in fact, in constructing the list L, we need no advantage of quantum calculation, so queries in Step 2 can also be made *classically* if we are allowed to access a classical oracle of H). Thus, the total number of quantum queries is $O(k + \sqrt{N/k})$, which is minimized when $k = N^{1/3}$. Brassard *et al.* gave the following theorem [BHT97].

Theorem 3.1 ([BHT97, Theorem 1]). *Suppose that X and Y are finite sets that satisfy $|X| = 2 \cdot |Y|$, and $H\colon X \to Y$ is a 2-to-one function. Let $N = |Y|$ and k be an integer such that $1 \leq k \leq N$.* **BHT** *finds a 2-collision of H with an expected quantum query complexity $O(k + \sqrt{N/k})$ and memory complexity $O(k)$. In particular, when $k = N^{1/3}$,* **BHT** *finds a 2-collision of H with expected quantum query complexity $O(N^{1/3})$ and memory complexity $O(N^{1/3})$.*

Element l-distinctness problem (l-collisions in D-Arb). Consider *the element l-distinctness problem*, in which we are given access to the oracle $H\colon X' \to Y$ to find whether there exist distinct $x_1, \ldots, x_l$ such that $H(x_1) = \cdots = H(x_l)$, i.e., there exits an l-collision of H. Note that H obviously has an l-collision if $|X'| \geq (l-1)|Y|$, and the element l-distinctness problem considers the collision detecting problem on the *database* rather than the hash function.

Ambainis [Amb07] proposed a quantum algorithm based on quantum walks that solves the element l-distinctness problem. His algorithm finds not only whether there exists an l-collision but also an actual l-collision value $(\{x_1, \ldots, x_l\}, y)$ and can be applied even for finding collisions in $|X'| \geq (l-1)|Y|$. His algorithm requires $O(|X'|^{l/(l+1)})$

quantum queries to H. This algorithm was later improved by Belovs [Bel12], who developed an algorithm that requires $O(|X'|^{1-2^{l-2}/(2^l-1)}) = o(|X'|^{3/4})$ quantum queries.[3]

Although the algorithms by Ambainis and Belovs can be applied to find an l-collision for $|X'| \geq (l-1)|Y|$, the complexity increases as the domain size $|X'|$ increases. These algorithms are inefficient to find collisions of hash functions, since the domain size of cryptographic hash functions is exponentially larger than the codomain size, and we often regard the problem size as dependent on the codomain size $|Y|$ not the domain size $|X'|$. Hence we need another dedicated quantum algorithm to efficiently find collisions of hash functions. The black circles and rectangles in Fig. 2 correspond to the query complexity for naïve applications of Ambainis' algorithm and Belovs' algorithm for hash functions, respectively.

Collision finding problem on random functions (l-collisions in D-Rnd and H-Rnd). Among variants of the collision problem, the *collision finding problem on random functions* is the most significant problem in the context of cryptography. We introduce algorithms for $l = 2$ in the following.

A modification of **BHT**. Let us consider a modification of **BHT**, denoted **BHT′**, in which we choose a subset X' uniformly at random. This small modification yields two important improvements of **BHT**:

- Brassard *et al.* [BHT97] mentioned that if $|X| \geq l|Y|$,then $\mathbf{BHT}'(H, N^{1/3})$ finds a collision with quantum query complexity $O(N^{1/3})$ with constant probability.
- Yuen [Yue14] showed that if $|X| = |Y|$ and H is random, then $\mathbf{BHT}'(H, N^{1/3})$ finds a collision with quantum query complexity $O(N^{1/3})$ with constant probability.

Zhandry's algorithm. Zhandry [Zha15] proposed a quantum algorithm finding a collision with $O(N^{1/3})$-quantum queries even if $|X| = \Omega(N^{1/2})$ and H is random. This improves the restrictions of **BHT** and **BHT′**, $|X| \geq 2|Y|$ [BHT97], or $|X| = |Y|$ and H is random [Yue14].

His algorithm is summarized as follows:

1. Choose a random subset $X' \subset X$ of size $N^{1/2}$.
2. Invoke Ambainis' algorithm for $H|_{X'} : X' \to Y$ and obtain a collision.

The collision exists if H is random because of the birthday bound and the query complexity is $O(|X'|^{2/3}) = O((N^{1/2})^{2/3}) = O(N^{1/3})$.

3.2 New Observations

This section gives our new observations, which are summarized as:

1. In quantum setting, the trivial upper bound for finding an l-collision of a random function is $O(N^{1/2})$.

[3] For $l = 3$, there exists a further improvement of time complexity by Belovs, Childs, Jeffery, Kothari, and Magniez [BCJ+13] and Jeffery [Jef14]. However, the quantum query complexity is still $\tilde{O}(|X'|^{1-2^{3-2}/(2^3-1)}) = \tilde{O}(|X'|^{5/7})$.

2. We can find a 3-collision of a random function with quantum query complexity $O(N^{10/21})$.

Observation 1 is obtained by applying a generalized Grover algorithm, and Observation 2 is obtained by combining the idea of Zhandry [Zha15] and the result of Belovs [Bel12].

Trivial upper-bound for finding l-collisions in quantum setting. In the classical setting, the trivial upper bound for finding an l-collision is $O(N)$ because of the following algorithm:

1. Choose an element $x_1 \in X$ uniformly at random.
2. Operate exhaustive search to find x_i for $i = 2, \ldots, l$ that satisfies $H(x_i) = H(x)$.
3. Output $(\{x_1, \ldots, x_l\}, H(x_1))$ as an l-collision.

In the quantum setting, we can replace the exhaustive search with **BBHT**. We call this algorithm **Multi-Grover**, described as follows:

Definition 3.2 (Multi-Grover(H)).

1. *Choose an element $x_1 \in X$ uniformly at random and set $L = \{x_1\}$.*
2. *While $|L| < l$, do:*
   (a) *Invoke* **BBHT**(F) *to find $x \in X$ such that $H(x) = H(x_1)$, where we implement $F\colon X \to \{0,1\}$ as $F(x) = 1$ if and only if $H(x) = H(x_1)$.*
   (b) *If $x \notin L$, then $L \leftarrow L \cup \{x\}$.*
3. *Output $(L, H(x_1))$ as an l-collision.*

Roughly speaking, each step in the loop requires $O(N^{1/2})$ queries to find x_i. Thus, the total query complexity is $O(N^{1/2})$ for a small constant l. Therefore, to achieve a meaningful improvement, we need to find an l-collision with fewer than $O(N^{1/2})$ quantum queries.

We note that the lower bound of 2-collisions in [Zha15] also applies to multicollisions. Hence, complexity of any multicollision-finding algorithm is between $O(N^{n/3})$ by 2-collisions and $O(N^{n/2})$ by the trivial upper bound. This corresponds to between birthday bound and preimage bound in the classical setting.

Extension of element l-distinctness to l-collision. We observe that algorithms for l-distinctness problem can be used to find l-collisions of a random function $H\colon X \to Y$ by extending Zhandry's idea. Let X, Y be finite sets with $|Y| = N$ and $|X| \geq (l!)^{1/l} N^{(l-1)/l}$. Let $H\colon X \to Y$ be a random function.

1. Choose a random subset $X' \subset X$ of size $(l!)^{1/l} N^{(l-1)/l}$
2. Invoke Belovs' algorithm for $H|_{X'}\colon X' \to Y$ and obtain an l-collision

According to Suzuki, Tonien, Kurosawa, and Toyoda [STKT08], $H|_{X'}$ has an l-collision with probability approximately $1/2$. Thus, we observe that Belovs' algorithm can find

an l-collision of $H|_{X'}$ with quantum query complexity $O\left((N^{1-2^{l-2}/(2^l-1)})^{(l-1)/l}\right)$.[4] This matches the tight bound $\Theta(N^{1/3})$ for $l = 2$ [Zha15] and gives a new upper bound $O(N^{10/21})$ for $l = 3$, which is crucially lower than the trivial bound $O(N^{1/2})$ (see section 3.2). The white rectangles for $l = 2, 3$ in Figure 1 correspond to this algorithm.

Note that for the case of H-Rnd, if $l \geq 4$, $(N^{1-2^{l-2}/(2^l-1)})^{(l-1)/l}$ becomes greater than or equal to $N^{1/2}$, which matches the trivial bound for finding l-collisions. Therefore, we have to make another quantum algorithm if we want to find l-collisions for $l \geq 4$ with fewer than $N^{1/2}$ quantum queries.

Our algorithm given in the next section finds an l-collision with the same query complexity as existing work for $l = 2$, and less query complexity than observations above for $l \geq 3$.

4 New Quantum Algorithm for Finding Multicollisions

Now we describe our algorithm for finding multicollisions. We begin with intuitive arguments about how to come up with an algorithm for finding multicollisions by extending the **BHT** algorithm and then give a formal description of our algorithm.

4.1 Intuitive Discussion from 2-Collisions to l-Collisions

First, we intuitively assume that $\mathbf{BHT}(H, k)$ can find a collision for a function $H\colon X \to Y$ if $|X| = 2N$ without any modification, because the expected number of preimages $|H^{-1}(y)|$ for each $y \in Y$ is 2 when H is chosen uniformly at random from $\mathrm{Map}(X, Y)$. (See Section 3.1 and the original paper [BHT97] for this justification.) Recall that the principle of $\mathbf{BHT}(H, k)$ is to make a list L of 1-collisions the size of which is k and to extend 1-collisions in L to 2-collisions with the **BBHT** algorithm. Constructing the list L requires k quantum queries, and **BBHT** makes $O(\sqrt{N/k})$ quantum queries, so the total number of quantum queries is $O(k + \sqrt{N/k})$. The optimal k that minimizes $k + \sqrt{N/k}$ satisfies $k = \sqrt{N/k}$, which is $k = N^{1/3}$ and then $O(k + \sqrt{N/k}) = O(N^{1/3})$.

Next we consider to find a 3-collision of a function $H\colon X \to Y$ under the condition $|X| = 3N$. We take a similar strategy to that of **BHT**, i.e., we make a list L of 2-collisions the size of which is k, and extend 2-collisions in L to 3-collisions with the **BBHT** algorithm. We can find a 2-collision of H with $\mathbf{BHT}(H, N^{1/3})$, which makes $O(N^{1/3})$ quantum queries. Constructing the list L requires $k \cdot N^{1/3}$ queries, and **BBHT** makes $O(\sqrt{N/k})$ quantum queries, so the total number of quantum queries is $O(k \cdot N^{1/3} + \sqrt{N/k})$. The optimal k that minimizes $k \cdot N^{1/3} + \sqrt{N/k}$ satisfies $k \cdot N^{1/3} = \sqrt{N/k}$. This is $k = N^{1/9}$ and then $O(k \cdot N^{1/3} + \sqrt{N/k}) = O(N^{4/9})$. Hence, our new algorithm improves the bound $O\left(N^{10/21}\right)$ for $l = 3$, which we observed in the previous section.

[4] The approach is not improved by picking a smaller random subset. For example, consider finding 3-collision of random function H. If we pick a smaller random subset X' of size N^b with $b < 2/3$,, then the probability that X' contains a 3-collision is roughly N^{3b}/N^2. Thus, we need to iterate Belovs' algorithm N^2/N^{3b} times, where each iteration makes $N^{5b/7}$ queries. Therefore, the total number of queries is $N^{(14-16b)/7} > N^{10/21}$ for $b < 2/3$.

Similarly to above, we can find l-collisions of a function $H\colon X \to Y$ under the condition $|X| = lN$, i.e., we construct a list L of $(l-1)$-collisions of the size k, and extend $(l-1)$-collisions in L to l-collisions using **BBHT**. By inductive argument, we can find that constructing the list L requires $k \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})}$ queries, and **BBHT** makes $O(\sqrt{N/k})$ quantum queries, so the total number of quantum queries is $O(k \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})} + \sqrt{N/k})$. The optimal k that minimizes $k \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})} + \sqrt{N/k}$ satisfies $k \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})} = \sqrt{N/k}$, which is $k = N^{1/3^{l-1}}$, and then

$$O\left(k \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})} + \sqrt{N/k}\right) = O\left(N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}\right)$$

holds. Again, our new algorithm improves the trivial bound $N^{1/2}$ for l-collisions, $l \geq 4$.

If there exists an algorithm finding l-collisions in the case $|X| = lN$, then we can use it to find l-collisions in the case $|X| > lN$ with the same number of queries and the same memory size, by choosing a subset $X' \subset X$ of size lN and by operating the algorithm on $H|_{X'}$.

4.2 Formal Description of Our Algorithm

Formalizing the above arguments, we obtain a quantum algorithm that finds l-collisions of any function $H\colon X \to Y$ with $|Y| = N$ and $|X| \geq lN$. As briefly introduced in Section 4.1, our main idea is to construct a recursive algorithm **MColl**. The algorithm below focuses on the procedure. Complexity analysis of **MColl** will be given in the next section. Although our algorithm is an extension of **BHT**, the definition of the function F is slightly modified to simplify the complexity analysis.

MColl(H, l)

1. If $|X| > lN$, then choose a subset $X' \subset X$ such that $|X'| = lN$ uniformly at random and operate **MColl**$(H|_{X'}, l)$. Otherwise proceed to the next step.
2. If $l = 1$, then choose x from X uniformly at random and output $(\{x\}, H(x))$. Otherwise proceed to the next step.
3. Operate **MColl**$(H, l-1)$ repeatedly for $N^{1/3^{l-1}}$ times and obtain $(l-1)$-collisions $c^{(i)} = (\{x_1^{(i)}, x_2^{(i)}, \ldots, x_{l-1}^{(i)}\}, y^{(i)})$. Store these $(l-1)$-collisions in a list L.
4. Sort L in accordance with $y^{(i)}$.
5. Check whether L contains duplication, i.e., there exist indices $i \neq j$ such that $c^{(i)} = c^{(j)}$. If it does, then stop and restart from Step 3. Otherwise proceed to the next step.
6. Check whether L contains an l-collision. If there is an l-collision, then output it. Otherwise proceed to the next step.
7. Define $F\colon X \to \{0,1\}$ by $F(x) = 1$ if and only if $H(x) = y^{(i)}$ holds for $1 \leq {}^{\exists}i \leq N^{1/3^{l-1}}$ (F can be implemented in a quantum circuit by calling H twice as shown in Fig. 3).
8. Operate **BBHT**(F). Let $\tilde{x} \in X$ be the obtained answer, which satisfies $F(\tilde{x}) = 1$.
9. Find i_0 that satisfies $H(\tilde{x}) = y^{(i_0)}$ from the list L. If $\tilde{x} \in \{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}\}$, then stop and restart from Step 3. Otherwise output an l-collision $(\{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}, \tilde{x}\}, y^{(i_0)})$.

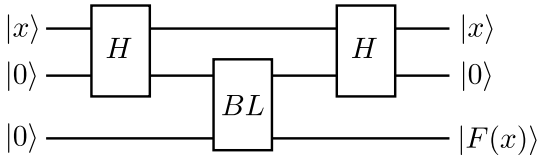

Fig. 3: Quantum circuit of F. H is the function we want to find collisions, and $BL\colon Y \to \{0,1\}$ is the binary function that is defined by $BL(y) = 1$ if and only if there exists $c^{(i)} = \left(\{x_1^{(i)}, \ldots, x_{l-1}^{(i)}\}, y^{(i)}\right) \in L$ such that $y = y^{(i)}$. Here BL corresponds to a quantum circuit $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus BL(x)\rangle$.

5 Complexity Analysis of MColl

In this section we analyze the complexity of **MColl**. First, we discuss complexity intuitively in Section 5.1 and then give formal arguments and proofs in Section 5.2.

5.1 Intuitive Analysis

We intuitively discuss the complexity of our algorithm. In the following, we show that **MColl**(H, l) finds that an l-collision with memory complexity is approximately $N^{1/3}$ and the expected quantum query complexity is at most approximately $l! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}$.

First, we consider memory complexity. The claim obviously holds for $l = 1$. In the case $l \geq 2$, the algorithm uses memory only for storing the list L. The memory size needed for L is $N^{1/3^{l-1}}$, which is less than or equal to $N^{1/3}$. Thus, the memory complexity is at most $N^{1/3}$.

Next, we consider quantum query complexity. We upper bound the expected number of quantum queries by approximately

$$Q_l := (N/P_l) \cdot l! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})},$$

where P_l is the number of the points in Y that have at least l preimages for a fixed H. Regarding (N/P_l) as constant, we obtain the desired bound.

The claim obviously holds for $l = 1$. Assume that the claim holds for $(l-1)$. Since Step 3 makes $N^{1/3^{l-1}}$-times calls of **MColl**$(H, l-1)$, the number of queries made in operating Step 3 once is approximately

$$\begin{aligned} N^{1/3^{l-1}} \cdot Q_{l-1} &= N^{1/3^{l-1}} \cdot \left((N/P_{l-1}) \cdot (l-1)! \cdot N^{(3^{l-2}-1)/(2\cdot 3^{l-2})}\right) \\ &= (N/P_{l-1}) \cdot (l-1)! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}. \end{aligned} \tag{1}$$

Note that **BBHT**(F) finds an element x that satisfies $F(x) = 1$ with approximately $\sqrt{1/p}$ queries to F, where $p = \Pr_{x \leftarrow X}[F(x) = 1]$. Since L contains $N^{1/3^{l-1}}$ elements, here we approximately argue that $p \approx N^{1/3^{l-1}}/|X| \approx N^{(1-3^{l-1})/3^{l-1}}$ and thus the number of queries to F is approximately $\sqrt{1/p}$, which is further approximated to $N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}$.

From the construction of F, the number of queries to H in Step 8 is twice the number of queries to F (see Fig. 3), so the number of queries to H is

$$2 \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}. \tag{2}$$

Summing up the numbers of queries in Steps 3 and 8 in Eqs. (1) and (2), we obtain the number of queries to H in the case that $\mathbf{MColl}(H, l)$ does not stop in Steps 5 or 9 as

$$((N/P_{l-1}) \cdot (l-1)! + 2) \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})} \approx (N/P_{l-1}) \cdot (l-1)! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}. \tag{3}$$

Now let q denote the probability that $\mathbf{MColl}(H, l)$ outputs without being terminated at Steps 5 or 9. Then the overall quantum query complexity is approximately $(1/q) \cdot \left((N/P_{l-1}) \cdot (l-1)! \cdot N^{(3^{l-1}-1)/2\cdot 3^{l-1}}\right)$. We assume that q equals the probability that an l-collision is outputted in Step 9, since the probability that Step 5 finds a duplication in L is very small when l is a small constant, and ignoring Step 6 only decreases q and increases the overall complexity. Intuitively, we can assume that q equals the product of two probabilities in Step 9:

1. The probability that the $(l-1)$-collision $\{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}\}$ can be extended to an l-collision.
2. The probability that $\tilde{x} \notin \{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}\}$ holds, under the condition in which $\{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}\}$ can be extended to an l-collision.

The probability of 1. is approximately P_l/P_{l-1}, and the probability of 2. is lower bounded by $1/l$. Thus, we have $q \geq (P_l/P_{l-1}) \cdot (1/l)$. Consequently, we have overall approximated complexity

$$(1/q) \cdot \left((N/P_{l-1}) \cdot (l-1)! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}\right),$$

which is at most

$$Q_l = (N/P_l) \cdot l! \cdot N^{(3^{l-1}-1)/2\cdot 3^{l-1}}.$$

This validates the claim.

5.2 Precise Analysis

The discussion in the previous section is very informal with many approximations. This section gives the precise bound and proof. The main theorem in this section is as follows:

Theorem 5.1. *Let X and Y be finite sets with $|Y| = N$ and $|X| \geq l \cdot |Y|$. Let $H\colon X \to Y$ be an arbitrary function. For $l \geq 1$, $\mathbf{MColl}(H, l)$ finds an l-collision with expected quantum query complexity at most*

$$\left(1 + 18\sqrt{2}e\right) \cdot \left(\frac{2lN^{1/3}}{2lN^{1/3} - 1}\right)^{l-1} \cdot l \cdot l! \cdot N^{(3^{l-1}-1)/(2\cdot 3^{l-1})}$$

and memory complexity $N^{1/3}$.

Remark 5.1. Expected time complexity of $\mathbf{MColl}(H, l)$ is upper bounded by the product of expected quantum query complexity and $O(T_H + \lg N)$, where T_H is the time needed to make a quantum query to H once, using $O(N^{1/3})$ qubits. See Section 6 for details.

Proof. It suffices to show that the claim holds in the case $|X| = l \cdot |Y|$. The proof for memory complexity is the same as that we described in the previous section. In the following, we consider quantum query complexity.

For $l \geq 1$, define A_l as

$$A_l := \text{the total number of quantum queries to } H \text{ that } \mathbf{MColl}(H, l) \text{ makes,}$$

and for $l \geq 2$, define B_l, C_l as

$$B_l := \text{the number of quantum queries to } H \text{ made in Step 3,}$$
$$C_l := \text{the number of quantum queries to } H \text{ made in Step 8.}$$

For $l \geq 2$, we consider a modification of $\mathbf{MColl}(H, l)$, denoted by $\mathbf{MColl}'(H, l)$, which never restarts from Step 3 once it stops in Steps 5 or 9. Let D_l be the total number of quantum queries to H that $\mathbf{MColl}'(H, l)$ makes. Let success denote the event such that $\mathbf{MColl}'(H, l)$ outputs an l-collision. Then we have

$$\mathrm{E}[D_l] = \mathrm{E}[B_l] + \mathrm{E}[C_l]$$

and

$$\mathrm{E}[A_l] = \frac{\mathrm{E}[D_l]}{\Pr[\mathsf{success}]} = \frac{\mathrm{E}[B_l] + \mathrm{E}[C_l]}{\Pr[\mathsf{success}]} \tag{4}$$

for $l \geq 2$. In addition, since $\mathrm{E}[B_l] = N^{1/3^{l-1}} \cdot \mathrm{E}[A_{l-1}]$ for $l \geq 2$, we have

$$\mathrm{E}[A_l] = \frac{N^{1/3^{l-1}} \cdot \mathrm{E}[A_{l-1}] + \mathrm{E}[C_l]}{\Pr[\mathsf{success}]}. \tag{5}$$

We will show two lemmas on bounds for $\mathrm{E}[C_l]$ and $\Pr[\mathsf{success}]$ in Sects. 5.3 and 5.4, respectively:

Lemma 5.1. *For $l \geq 2$, $\mathrm{E}[C_l] \leq 18 \cdot \sqrt{\frac{l}{l-1}} \cdot N^{\frac{3^{l-1}-1}{2 \cdot 3^{l-1}}}$ holds.*

Lemma 5.2. *For $l \geq 2$, $\Pr[\mathsf{success}] \geq \frac{l-1}{l} \cdot \frac{1}{l} \cdot \left(1 - \frac{1}{2l} \cdot N^{\frac{2}{3^{l-1}}-1}\right)$ holds.*

Putting them in the inequality (5), we obtain

$$\mathrm{E}[A_l] \leq \left(N^{1/3^{l-1}}\, \mathrm{E}[A_{l-1}] + 18\sqrt{\frac{l}{l-1}} N^{\frac{3^{l-1}-1}{2 \cdot 3^{l-1}}}\right) \cdot \frac{l^2 f_l}{l-1}, \tag{6}$$

where

$$f_l = \frac{1}{1 - \frac{1}{2l} \cdot N^{\frac{2}{3^{l-1}}-1}}.$$

Let $\{g_l\}_{1\le l}$ be a sequence of numbers defined by $g_1 = 1$ and

$$g_l = \left(g_{l-1} + \frac{18\sqrt{l/(l-1)}}{(l-1)\cdot(l-1)!}\right) f_l$$

for $l \ge 2$.

We show the following claims:

Claim. For $l \ge 1$, $\mathrm{E}[A_l] \le g_l \cdot l \cdot l! \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}$ holds.

Claim. For $l \ge 1$, $g_l \le \left(1 + 18\sqrt{2}e\right) \cdot \left(\frac{2lN^{1/3}}{2lN^{1/3}-1}\right)^{l-1}$ holds.

Combining them, we obtain for $l \ge 1$,

$$\mathrm{E}[A_l] \le \left(1 + 18\sqrt{2}e\right) \cdot \left(\frac{2lN^{1/3}}{2lN^{1/3}-1}\right)^{l-1} \cdot l \cdot l! \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}$$

as we wanted. □

Proof (Proof of Claim). We give a proof of this claim by induction on l. Since $\mathrm{E}[A_1] = 1$, the claim holds for $l = 1$. Now we assume that the claim holds for $(l-1)$. By the induction, we have

$$\begin{aligned}
\mathrm{E}[A_l] &\le \left(N^{1/3^{l-1}}\,\mathrm{E}[A_{l-1}] + 18\sqrt{\frac{l}{l-1}}N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}\right)\cdot\frac{l^2 f_l}{l-1} \\
&\le \left(N^{1/3^{l-1}}\left(g_{l-1}\cdot(l-1)\cdot(l-1)!\cdot N^{\frac{3^{l-2}-1}{2\cdot 3^{l-2}}}\right) + 18\sqrt{\frac{l}{l-1}}N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}\right)\cdot\frac{l^2 f_l}{l-1} \\
&= \left(g_{l-1} + \frac{18\sqrt{l/(l-1)}}{(l-1)\cdot(l-1)!}\right)\cdot f_l \cdot l \cdot l! \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}} \\
&= g_l \cdot l \cdot l! \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}
\end{aligned}$$

and the claim also holds for any $l \ge 1$. □

Proof (Proof of Claim). Finally, we upper bound g_l. Letting $h_l = \frac{18\sqrt{l/(l-1)}}{(l-1)\cdot(l-1)!}$, we have $g_l = (g_{l-1} + h_l) f_l$. Since $f_l \ge 1$ holds for $l \ge 2$, we have

$$g_l = (g_{l-1} + h_l)\, f_l = ((g_{l-2} + h_{l-1})\, f_{l-1} + h_l)\, f_l \le (g_{l-2} + h_{l-1} + h_l)\, f_{l-1} f_l.$$

Continuing calculations, we obtain $g_l \leq \left(1 + \sum_{i=2}^{l} h_i\right) \cdot \prod_{i=2}^{l} f_i$. Thus, we have

$$\begin{aligned}
g_l &\leq \left(1 + \sum_{i=2}^{l} h_i\right) \cdot \prod_{i=2}^{l} f_i \\
&= \left(1 + \sum_{i=2}^{l} \frac{18\sqrt{i/(i-1)}}{(i-1)\cdot(i-1)!}\right) \prod_{i=2}^{l} \frac{1}{1 - \frac{1}{2l} \cdot N^{\frac{2}{3^{i-1}}-1}} \\
&\leq \left(1 + \sum_{i=2}^{l} \frac{18\sqrt{2}}{(i-1)!}\right) \prod_{i=2}^{l} \frac{1}{1 - \frac{1}{2l} \cdot N^{-1/3}} \leq \left(1 + 18\sqrt{2}\left(\sum_{i=2}^{l} \frac{1}{(i-1)!}\right)\right) \prod_{i=2}^{l} \frac{2lN^{1/3}}{2lN^{1/3}-1} \\
&\leq \left(1 + 18\sqrt{2}\left(\sum_{i=0}^{\infty} \frac{1}{i!}\right)\right) \left(\frac{2lN^{1/3}}{2lN^{1/3}-1}\right)^{l-1} = \left(1 + 18\sqrt{2}e\right) \left(\frac{2lN^{1/3}}{2lN^{1/3}-1}\right)^{l-1},
\end{aligned}$$

as we wanted. □

5.3 Proof of Lemma 5.1

Note that

$$\mathrm{E}[C_l] \leq \mathrm{E}[C_l \mid \text{Step 8 is operated}]$$

holds, and we upper bound the conditional expectation $\mathrm{E}[C_l \mid \text{Step 8 is operated}]$. When the algorithm operates in Step 8, it has already passed Steps 5 and 6. Thus, L has neither duplication nor l-collision. In particular, we can assume that L is a list of completely distinct $(l-1)$ collisions of H, i.e., $y^{(i_1)} = y^{(i_2)}$ holds if and only if $i_1 = i_2$. Thus, we have

$$|F^{-1}(1)| = \left|\bigcup_{i=1}^{N^{1/3^{l-1}}} H^{-1}(y^{(i)})\right| = \sum_{i=1}^{N^{1/3^{l-1}}} |H^{-1}(y^{(i)})| \geq (l-1)\cdot N^{1/3^{l-1}}$$

and

$$\frac{|F^{-1}(1)|}{|X|} \geq \frac{(l-1)\cdot N^{1/3^{l-1}}}{l\cdot N}.$$

Since **MColl** makes two quantum queries to H while making one query to F (See Fig. 3), we have

$$\mathrm{E}[C_l \mid \text{Step 8 is operated}] \leq 2\cdot 9\cdot \sqrt{\frac{l\cdot N}{(l-1)\cdot N^{1/3^{l-1}}}} = 18\cdot \sqrt{\frac{l}{l-1}} \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}$$

by Corollary 2.1 as we wanted.

5.4 Proof of Lemma 5.2

Next, we lower bound $\Pr[\mathsf{success}]$. Note that

$$\Pr[\mathsf{success}] = \Pr\left[c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right] \cdot \Pr\left[\mathsf{success} \,\middle|\, c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right]$$

holds.

We need two lemmas. For the proof of Lemma 5.3, we refer readers to Shoup's textbook [Sho08]. The proof of Lemma 5.4 is given in Appendix A.

Lemma 5.3 ([Sho08, Theorem 8.26]). *Let $[d]$ be the set of integers $\{1, 2, \ldots, d\}$, and $[d]^{\times n}$ be the n-array Cartesian power set of $[d]$ for positive integers d, n. If $s = (s_1, s_2, \ldots, s_n)$ is chosen uniformly at random from $[d]^{\times n}$, then the probability that $s_i \neq s_j$ holds for all $i \neq j$ is lower bounded by $1 - n^2/(2d)$.*

Lemma 5.4. *Let X and Y be finite sets with $|Y| = N$ and $|X| = lN$. Let H be a function from X to Y. Then the number of l-collisions and $(l-1)$-collisions of H are greater than or equal to N and lN, respectively.*

First, we lower bound $\Pr\left[c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right]$. From the construction of **MColl**, we can assume that **MColl**$(H, l-1)$ outputs an $(l-1)$-collision of H uniformly at random. Thus, we can assume that elements $c^{(i)} \in L$ are chosen independently and uniformly at random from the set of $(l-1)$-collisions of H. By Lemma 5.4, the number of $(l-1)$-collisions of H is at least $l \cdot N$. Moreover, if n is fixed, $1 - n^2/2d$ is a monotonically increasing function on d. Therefore, by Lemma 5.3, we have

$$\Pr\left[c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right] \geq 1 - \frac{(N^{1/3^{l-1}})^2}{2lN} = 1 - \frac{1}{2l} \cdot N^{\frac{2}{3^{l-1}}-1}. \tag{7}$$

Second, we lower bound $\Pr\left[\mathsf{success} \;\middle|\; c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right]$. Note that the event $\mathsf{success}$ occurs if and only if

$$y^{(i)} = y^{(j)} \text{ for some } i \neq j, \tag{8}$$

or

$$c^{(i_0)} \text{ can be extended to an } l\text{-collision, and } \tilde{x} \notin \left\{x_1^{(i_0)}, x_2^{(i_0)}, \ldots, x_{l-1}^{(i_0)}\right\}. \tag{9}$$

occurs. Recall that $\tilde{x}$ is the output of Step 8 and i_0 is an index satisfying $H(\tilde{x}) = y^{(i_0)}$. The event (8) corresponds to the event in which **MColl** finds an l-collision in Step 6, and the event (9) corresponds to the event in which **MColl** finds an l-collision in Step 9.

Now, let $\mathcal{L}$ be all the possible lists L that satisfy $c^{(i)} \neq c^{(j)}$ for $i \neq j$. Let $\mathcal{L}_1 \subset \mathcal{L}$ denote the set of lists in which there exists l-collisions, i.e., there are two indices $i \neq j$ such that $y^{(i)} = y^{(j)}$, and $\mathcal{L}_2 \subset \mathcal{L}$ denotes the set of lists in which there is no l-collision, i.e., $y^{(i)} \neq y^{(j)}$ holds for $i \neq j$. Then we have $\mathcal{L} = \mathcal{L}_1 \coprod \mathcal{L}_2$. **MColl** finds an l-collision in Step 6 if and only if $L \in \mathcal{L}_1$. In the following, we ignore Step 4 and consider that L is not sorted for simplicity.

For a fixed $L \in \mathcal{L}$, let A^L and B^L denote the sets of elements in L that can and cannot be extended to l-collisions, respectively. We have that $L = A^L \coprod B^L$, $|A^L|$ equals the number of $y^{(i)}$ such that $|H^{-1}(y^{(i)})| \geq l$, and $|B^L|$ equals the number of $y^{(i)}$ such that $|H^{-1}(y^{(i)})| = l - 1$. Define $\langle A^L \rangle$, $\langle B^L \rangle$ by

$$\langle A^L \rangle := \left| \bigcup_{c^{(i)}=(\ldots,y^{(i)}) \in A^L} H^{-1}(y^{(i)}) \right| \text{ and } \langle B^L \rangle := \left| \bigcup_{c^{(i)}=(\ldots,y^{(i)}) \in B^L} H^{-1}(y^{(i)}) \right|,$$

which are the numbers of preimages of $y^{(i)}$'s in A^L and B^L, respectively. Note that

$$\Pr[\mathsf{success} \mid c^{(i)} \neq c^{(j)} \text{ for } i \neq j] = \sum_{L \in \mathcal{L}} \Pr[\mathsf{success} \mid L] \Pr[L].$$

holds.

If $L \in \mathcal{L}_2$, then success occurs if and only if the event (9) occurs, that is, $\tilde{x}$ can be used to construct an l-collision with an $(l-1)$-collision in L. Note that $\tilde{x}$ is chosen uniformly at random from the set

$$\left(\bigcup_{c^{(i)}=(\ldots,y^{(i)}) \in A^L} H^{-1}(y^{(i)}) \right) \bigcup \left(\bigcup_{c^{(i)}=(\ldots,y^{(i)}) \in B^L} H^{-1}(y^{(i)}) \right),$$

and the event (9) occurs if and only if

$$\tilde{x} \in \bigcup_{c^{(i)}=(\ldots,y^{(i)}) \in A^L} H^{-1}(y^{(i)}) \wedge \tilde{x} \neq x_j^{(i)} \text{ for all } i \text{ and } j$$

holds. Now we have

$$\Pr\left[\tilde{x} \in \bigcup_{c^{(i)} \in A^L} H^{-1}(y^{(i)}) \,\middle|\, L \right] = \frac{\langle A^L \rangle}{\langle A^L \rangle + \langle B^L \rangle},$$

and

$$\Pr\left[\tilde{x} \neq x_j^{(i)} \text{ for all } i \text{ and } j \,\middle|\, L, \tilde{x} \in \bigcup_{c^{(i)} \in A^L} H^{-1}(y^{(i)}) \right] \geq \frac{1}{l},$$

which suggests that

$$\begin{aligned}
\Pr[\mathsf{success} \mid L] &= \Pr\left[\tilde{x} \in \bigcup_{c^{(i)} \in A^L} H^{-1}(y^{(i)}) \wedge \tilde{x} \neq x_j^{(i)} \text{ for all } i \text{ and } j \,\middle|\, L \right] \\
&= \Pr\left[\tilde{x} \in \bigcup_{c^{(i)} \in A^L} H^{-1}(y^{(i)}) \,\middle|\, L \right] \\
&\quad \cdot \Pr\left[\tilde{x} \neq x_j^{(i)} \text{ for all } i \text{ and } j \,\middle|\, L, \tilde{x} \in \bigcup_{c^{(i)} \in A^L} H^{-1}(y^{(i)}) \right] \\
&\geq \frac{\langle A^L \rangle}{\langle A^L \rangle + \langle B^L \rangle} \cdot \frac{1}{l}.
\end{aligned}$$

In addition, we have $\langle A^L \rangle \geq l \cdot |A^L|$ since $y^{(i)} \neq y^{(j)}$ holds for $i \neq j$ if $L \in \mathcal{L}_2$. Thus, we have $\langle A^L \rangle \geq l \cdot |A^L| \geq (l-1) \cdot |A^L|$ and $\langle B^L \rangle = (l-1) \cdot |B^L|$. This yields that

$$\frac{\langle A^L \rangle}{\langle A^L \rangle + \langle B^L \rangle} = \frac{1}{1 + \frac{\langle B^L \rangle}{\langle A^L \rangle}} \geq \frac{1}{1 + \frac{(l-1)|B^L|}{(l-1)|A^L|}} = \frac{|A^L|}{|A^L| + |B^L|}.$$

Thus, we have

$$\Pr[\mathsf{success} \mid L] \geq \frac{|A^L|}{|A^L| + |B^L|} \cdot \frac{1}{l} \tag{10}$$

for $L \in \mathcal{L}_2$. Moreover, since $\Pr[\mathsf{success} \mid L] = 1$ for $L \in \mathcal{L}_1$, the inequality (10) also holds for $L \in \mathcal{L}_1$. Therefore, we have

$$\begin{aligned}\Pr\left[\mathsf{success} \;\middle|\; c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right] &= \sum_{L \in \mathcal{L}} \Pr[\mathsf{success} \mid L] \Pr[L] \\ &\geq \frac{1}{l} \sum_{L \in \mathcal{L}} \frac{|A^L|}{|A^L| + |B^L|} \cdot \Pr[L].\end{aligned}$$

Now we use the following lemmas the proofs of which are given in Appendix B and Appendix C, respectively.

Lemma 5.5. *Let X, Y be finite sets such that $|X| = l \cdot |Y|$, and H be a function from X to Y. Let A, B denote the sets of $(l-1)$-collisions of H that can and cannot be extended to l-collisions, respectively. Then we have*

$$\frac{|A|}{|A| + |B|} \geq \frac{l-1}{l}.$$

Lemma 5.6. *Let X, Y be finite sets such that $|X| = l \cdot |Y|$, and H be a function from X to Y. Let A, B denote the sets of $(l-1)$-collisions of H that can and cannot be extended to l-collisions, respectively. Then we have*

$$\sum_{L \in \mathcal{L}} \frac{|A^L|}{|A^L| + |B^L|} \cdot \Pr[L] = \frac{|A|}{|A| + |B|}.$$

By the above lemmas, we have

$$\Pr\left[\mathsf{success} \;\middle|\; c^{(i)} \neq c^{(j)} \text{ for } i \neq j\right] \geq \frac{l-1}{l} \cdot \frac{1}{l}.$$

Consequently, $\Pr[\mathsf{success}]$ is lower bounded as $\Pr[\mathsf{success}] \geq \frac{l-1}{l} \cdot \frac{1}{l} \cdot \left(1 - \frac{1}{2l} \cdot N^{\frac{2}{3^{l-1}} - 1}\right)$, that completes the proof.

6 Discussions on Time Complexity

The previous section only focused on quantum query complexity. This section discusses time complexity of **MColl**. We measure the unit of time complexity by the number of executions of quantum gates, which operate primary binary calculations on $\lg N$-bit strings such as NOT, AND, OR, and XOR. For a function $F\colon X \to \{0,1\}$, **BBHT** finds an x_0 such that $F(x_0) = 1$ in time $O(\sqrt{|X|/t} \cdot T')$, where $t = |\{x \in X \mid F(x) = 1\}|$ and T' is the time for evaluating F once.

To begin with, we show the following theorem.

Theorem 6.1. *Let X, Y be finite sets with $|Y| = N$ and $|X| \geq l \cdot |Y|$. For any function $H\colon X \to Y$,* **MColl**(H, l) *runs in expected time*

$$C \cdot \left(\frac{2lN^{1/3}}{2lN^{1/3} - 1}\right)^{l-1} \cdot l \cdot l! \cdot N^{(3^{l-1}-1)/2\cdot 3^{l-1}} \cdot (T_H + \lg N)$$

for some constant C, using $O(N^{1/3})$ qubits, where T_H denotes the time needed to make a quantum query to H.

Proof. Let A'_l be the running time of **MColl**(H, l) for $l \geq 1$. For $l \geq 2$, let B'_l, G'_l, C'_l, K'_l be the running time of Steps 3, 4, 8, and 9, respectively. Similarly to the inequality 4, we have

$$\mathrm{E}[A'_l] = \frac{\mathrm{E}[B'_l] + \mathrm{E}[G'_l] + \mathrm{E}[C'_l] + \mathrm{E}[K'_l]}{\Pr[\text{success}]} \tag{11}$$

for $l \geq 2$. We have

$$\begin{aligned}
\mathrm{E}[B'_l] &= N^{1/3^{l-1}} \cdot \mathrm{E}[A'_{l-1}], \\
\mathrm{E}[G'_l] &= O\left(N^{1/3^{l-1}} \lg N^{1/3^{l-1}}\right) = O\left(N^{1/3^{l-1}} \lg N\right), \\
\mathrm{E}[K'_l] &= O\left(\lg N^{1/3^{l-1}}\right) = O\left(\lg N\right),
\end{aligned}$$

since Steps 4 and 9 can be done classically. In addition, we have that

$$\begin{aligned}
\mathrm{E}[C'_l] &= \mathrm{E}[C_l] \cdot \left(T_H + O\left(\lg N^{1/3^{l-1}}\right)\right) = O\left(\mathrm{E}[C_l] \cdot \left(T_H + \lg N^{1/3^{l-1}}\right)\right) \\
&= O\left(\sqrt{\frac{l}{l-1}} \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}} \cdot (T_H + \lg N)\right),
\end{aligned}$$

which follows from the construction of the quantum circuit of F (See Fig. 3) and the claim below. See Appendix D for details of this claim.

Claim. The quantum circuit BL can be constructed so that it runs in time $O(\lg N^{1/3^{l-1}})$ using $O(N^{1/3^{l-1}})$ qubits.

Eventually, we have

$$\begin{aligned}
\mathrm{E}[A'_l] &= O\left(\frac{N^{1/3^{l-1}} \cdot \mathrm{E}[A'_{l-1}] + N^{1/3^{l-1}} \lg N + \sqrt{\frac{l}{l-1}} \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}} \cdot (T_H + \lg N) + \lg N}{\Pr[\text{success}]}\right) \\
&\leq O\left(\frac{N^{1/3^{l-1}} \cdot \mathrm{E}[A'_{l-1}] + \sqrt{\frac{l}{l-1}} \cdot N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}}{\Pr[\text{success}]} \cdot (T_H + \lg N)\right) \\
&= O\left(\left(N^{1/3^{l-1}} \mathrm{E}[A'_{l-1}] + \sqrt{\frac{l}{l-1}} N^{\frac{3^{l-1}-1}{2\cdot 3^{l-1}}}\right) \cdot \frac{l^2 f_l}{l-1} \cdot (T_H + \lg N)\right).
\end{aligned}$$

The above equation yields the claim of Theorem 6.1 due to the same argument as that in the proof of Theorem 5.1.

Remark 6.1. From the viewpoint of time complexity, there are a few criticisms of existing quantum 2-collision finding algorithms [GR03,Ber09]. They are based on the observation that *memory size* is essentially the same as *machine size* for quantum machines, since we have to embed data that we use in a quantum algorithm into the quantum circuit of the algorithm.

Note that these criticisms only focus on collisions of *random* functions and thus are invalid when we consider finding collisions of *any* function. Furthermore, the target of these criticisms is *time complexity*, and our main result (Theorem 5.1), which focuses on *quantum query complexity*, is out of the scope of these criticisms.

7 Conclusion

Finding multicollisions is one of the most important problems in cryptology, both for attack and provable security. In the post-quantum era, this problem needs to be studied in a quantum setting to realize quantum-secure cryptographic schemes. This paper systematized knowledge on the multicollision-finding problem in a quantum setting and proposed a new quantum multicollision-finding algorithm. Our algorithm finds an l-collision of *any* function $H\colon X \to Y$, where $|X| \geq l \cdot |Y|$, with expected quantum query complexity $O(N^{(3^{l-1}-1)/2\cdot 3^{l-1}}) = o(N^{1/2})$ and memory complexity $O(N^{1/3})$ for a small constant l. If our algorithm is applied to the *random* function, the complexity matches the known tight bound for $l = 2$, improves the simple combination of Zhandry and Belovs' results for $l = 3$, and for the first time improves the simple bound of $O(N^{1/2})$ for $l \geq 4$. Getting rid of the condition $|X| \geq l \cdot |Y|$ and proving a lower bound to find an l-collision are left for future work.

The quantum stuff in this paper is encapsulated in Grover's algorithm, and the results can equally well be understood as query complexity given a "Grover black-box" without assuming any knowledge of quantum theory on the reader. We hope this paper encourages researchers in classical setting to actively discuss quantum algorithms.

References

Amb05. Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1:37–46, 2005. https://arxiv.org/abs/quant-ph/0304162.

Amb07. Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, 2007. The preliminary version appeared in FOCS 2004. See https://arxiv.org/abs/quant-ph/0311001.

AS04. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, July 2004.

BBHT98. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortsch. Phys.*, 46(4-5):493–505, June 1998. https://arxiv.org/abs/quant-ph/9605034.

BCJ+13. Aleksandrs Belovs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *ICALP 2013, Part I*, pages 105–122, 2013. See http://arxiv.org/abs/1302.3143 and http://arxiv.org/abs/1302.7316.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, 2011. https://eprint.iacr.org/2010/428.

Bel12. Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *FOCS 2012*, pages 207–216, 2012. https://arxiv.org/abs/1205.1534v2.

Ber09. Daniel J. Bernstein. Cost analysis of hash collisions: will quantum computers make SHARCS obsolete? In *SHARCS 2009*, 2009.

BHT97. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. *CoRR*, quant-ph/9705002, 1997. See also Quantum Cryptanalysis of Hash and Claw-Free Functions. LATIN 1998: 163-169. See https://arxiv.org/abs/quant-ph/9705002.

CN08. Donghoon Chang and Mridul Nandi. Improved indifferentiability security analysis of chopMD hash function. In *FSE 2008*, pages 429–443, 2008.

DDKS14. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Cryptanalysis of iterated Even-Mansour schemes with two keys. In *ASIACRYPT 2014, Part I*, pages 439–457, 2014.

Flo67. Robert W. Floyd. Nondeterministic algorithms. *Journal of the ACM*, 14(4):636–644, 1967.

GR03. Lov Grover and Terry Rudolph. How significant are the known collision and element distinctness quantum algorithms? *CoRR*, quant-ph/0309123, 2003. See GR04.

Gro96. Lov. K Grover. A fast quantum mechanical algorithm for database search. In *STOC 1996*, pages 212–219, 1996. https://arxiv.org/abs/quant-ph/9605043.

HIK+10. Shoichi Hirose, Kota Ideguchi, Hidenori Kuwakado, Toru Owada, Bart Preneel, and Hirotaka Yoshida. A lightweight 256-bit hash function for hardware and low-end devices: Lesamnta-LW. In *ICISC 2010*, pages 151–168, 2010.

HRS16. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016*, pages 387–416, 2016. https://eprint.iacr.org/2015/1256.

Jef14. Stacey Jeffery. *Frameworks for Quantum Algorithms*. PhD thesis, University of Waterloo, 2014.

JJV02. Éliane Jaulmes, Antoine Joux, and Frédéric Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. In *FSE 2002*, pages 237–251, 2002.

JL09. Antoine Joux and Stefan Lucks. Improved generic algorithms for 3-collisions. In *ASIACRYPT 2009*, pages 347–363, 2009. https://eprint.iacr.org/2009/305.

JLM14. Philipp Jovanovic, Atul Luykx, and Bart Mennink. Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In *ASIACRYPT 2014, Part I*, pages 85–104, 2014. https://eprint.iacr.org/2014/373.

KMRT09. Lars R. Knudsen, Florian Mendel, Christian Rechberger, and Søren S. Thomsen. Cryptanalysis of MDC-2. In *EUROCRYPT 2009*, pages 106–120, 2009.

Kut05. Samuel Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005. https://arxiv.org/abs/quant-ph/0304162.

MT08. Florian Mendel and Søren S. Thomsen. An observation on JH-512. Available online, 2008. http://ehash.iaik.tugraz.at/uploads/d/da/Jh_preimage.pdf.

NO14. Yusuke Naito and Kazuo Ohta. Improved indifferentiable security analysis of PHOTON. In *SCN 2014*, pages 340–357, 2014.

NS16. Ivica Nikolić and Yu Sasaki. A new algorithm for the unbalanced meet-in-the-middle problem. In *ASIACRYPT 2016*, pages 627–647, 2016.

NSWY13. Yusuke Naito, Yu Sasaki, Lei Wang, and Kan Yasuda. Generic state-recovery and forgery attacks on ChopMD-MAC and on NMAC/HMAC. In *IWSEC 2013*, pages 83–98, 2013.

NWW13. Ivica Nikolić, Lei Wang, and Shuang Wu. Cryptanalysis of round-reduced LED. In *FSE 2013*, pages 112–129, 2013.

RS96. Ronald L. Rivest and Adi Shamir. PayWord and MicroMint – two simple micropayment schemes. In *SPW 1996*, pages 69–87, 1996.

Sho08. Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2nd edition, January 2008.

STKT08. Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota. Birthday paradox for multi-collisions. *IEICE Transactions*, 91-A(1):39–45, 2008. The preliminary version is in *ICISC 2006*.

Yue14. Henry Yuen. A quantum lower bound for distinguishing random functions from random permutations. *Quantum Information & Computation*, 14(13-14):1089–1097, 2014. https://arxiv.org/abs/1310.2885.

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7-8):557–567, May 2015.

A Proof of Lemma 5.4

Define sequence of functions $\{H_i : X \to Y\}_{i \geq 0}$ as follows. First, define H_0 by $H_0 = H$. For each $i \geq 0$, if $|H_i^{-1}(y)| = l$ holds for all $y \in Y$ (i.e. H_i is a regular function), then define H_{i+1} by $H_{i+1} = H_i$. Otherwise, choose $x_1 \in X$ that satisfies $|H_i^{-1}(H_i(x_1))| > l$ and $y_2 \in Y$ that satisfies $|H_i^{-1}(y_2)| < l$ and define H_{i+1} by

$$H_{i+1}(x) = \begin{cases} H_i(x) & (x \neq x_1), \\ y_2 & (x = x_1). \end{cases}$$

Note that there exists an index i_0 such that $H_{i_0} = H_{i_0+j}$ holds for all $j \geq 0$ since $|X|, |Y| < \infty$, and $|X| = l \cdot |Y|$.

Let a_i be the number of l-collisions of H_i and k_i be $|H_i^{-1}(H_i(x_1))|$. When i is incremented, then the number of the preimages of y_2 is incremented, and the number of l-collisions is increased at most 1 accordingly. On the other hand, the number of the preimages of $H_i(x_1)$ is decremented, and the number of l-collisions is decreased by $\binom{k_i}{l} - \binom{k_i - 1}{l}$ accordingly. Therefore, we have

$$a_{i+1} \leq a_i - \left(\binom{k_i}{l} - \binom{k_i - 1}{l} \right) + 1.$$

Since $k_i = |H_i^{-1}(H_i(x_1))| > l$,

$$\binom{k_i}{l} - \binom{k_i - 1}{l} = \binom{k_i - 1}{l} + \binom{k_i - 1}{l - 1} - \binom{k_i - 1}{l - 1} = \binom{k_i - 1}{l - 1} \geq 1$$

holds, and thus we have $a_i \geq a_{i+1}$ for $i \geq 0$. By constructing the sequence $\{H_i\}_{i \geq 0}$, there exists an integer i_0 such that $H_{i_0} = H_{i_0+j}$ holds for all $j \geq 0$. Since H_{i_0} satisfies $|H_{i_0}^{-1}(y)| = l$ for all $y \in Y$, we have $a_{i_0} = N$. Therefore, $a_0 \geq a_1 \geq \cdots \geq a_{i_0} = N$ holds, which completes the proof for l-collisions.

Next, let b_i denote the number of $(l-1)$-collisions of H_i. Similarly, for the proof for l-collisions, we have

$$b_{i+1} \leq b_i - \left(\binom{k_i}{l-1} - \binom{k_i - 1}{l-1}\right) + (l-1).$$

Since $k_i = |H_i^{-1}(H_i(x_1))| > l$,

$$\binom{k_i}{l-1} - \binom{k_i - 1}{l-1} = \binom{k_i - 1}{l-1} + \binom{k_i - 1}{l-2} - \binom{k_i - 1}{l-2} = \binom{k_i - 1}{l-2} \geq \binom{l-1}{l-2} = l - 1$$

holds, and thus we have $b_i \geq b_{i+1}$ for $i \geq 0$. Since H_{i_0} satisfies $|H_{i_0}^{-1}(y)| = l$ for all $y \in Y$, we have $b_{i_0} = l \cdot N$. Therefore, $b_0 \geq b_1 \geq \cdots \geq b_{i_0} = l \cdot N$ holds, which completes the proof.

B Proof of Lemma 5.5

First, we have $|B| \leq N - 1$, since it contradicts the condition $|X| = l \cdot |Y|$ if we assume $|B| \geq N$. Since $|A| + |B| \geq l \cdot N$ holds by Lemma 5.4, we have

$$\frac{|A|}{|A| + |B|} = 1 - \frac{|B|}{|A| + |B|} \geq 1 - \frac{N-1}{l \cdot N} = \frac{l - 1 + \frac{1}{N}}{l} \geq \frac{l-1}{l},$$

which completes the proof.

C Proof of Lemma 5.6

Let S be the direct union set of A and B, that is, $S := A \coprod B$. Consider a trial $\mathsf{T}(a, b; k)$ in which we choose k elements independently and uniformly at random from S, and make an list of chosen elements (here we consider k-permutations of $|S|$, rather than a combination), where $a = |A|$ and $b = |B|$.

Since L is an element of $\mathcal{L}$, which is the set of all the possible lists L that satisfy $c^{(i)} \neq c^{(j)}$ for $i \neq j$, L can be regarded as the list made by operating this trial with $k = N^{1/3^{l-1}}$. The sets A^L and B^L correspond to the sets of elements in the list L chosen from A and B, respectively.

We consider trial $\mathsf{T}(a, b; k)$ for non-negative integers a, b, k such that $a \geq 1$ or $b \geq 1$, and $1 \leq k \leq a + b$. In considering the trial $\mathsf{T}(a, b; k)$, we focus on cardinality of sets $a = |A|$ and $b = |B|$, rather than sets A, B themselves. We show the following claim:

Claim. For non-negative integers a, b, k such that $a \geq 1$ or $b \geq 1$, and $1 \leq k \leq a + b$,

$$\mathop{\mathrm{E}}_{a,b,k}[|A^L|] = \frac{ka}{a+b}$$

holds, where $\mathrm{E}_{a,b,k}$ denotes the expected value corresponding to the trial $\mathsf{T}(a, b; k)$.

If the above claim holds, then we can finish the proof of our lemma, because the statement of our lemma corresponds to the trial $\mathsf{T}(|A|, |B|; N^{1/3^{l-1}})$ and we have

$$\sum_{L \in \mathcal{L}} \frac{|A^L|}{|A^L| + |B^L|} \cdot \Pr[L] = \mathop{\mathrm{E}}_{|A|,|B|,N^{1/3^{l-1}}} \left[\frac{|A^L|}{|A^L| + |B^L|} \right] = \mathop{\mathrm{E}}_{|A|,|B|,N^{1/3^{l-1}}} \left[\frac{|A^L|}{N^{1/3^{l-1}}} \right]$$
$$= \frac{1}{N^{1/3^{l-1}}} \cdot \frac{N^{1/3^{l-1}}|A|}{|A| + |B|} = \frac{|A|}{|A| + |B|}.$$

Now, we prove the claim by induction on $|S| = |A| + |B| = a + b$. If $a + b = 1$, then it is obvious. Assume that the claim holds for $a + b - 1$. For each element $s \in S$, let $\mathcal{L}_s \subset \mathcal{L}$ denote the set of lists the first element of which is s. Then, since $\mathcal{L} = \coprod_{s \in S} \mathcal{L}_s$, we have

$$\begin{aligned}
\mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \right] &= \sum_{s \in S} \mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \;\middle|\; L \in \mathcal{L}_s \right] \cdot \Pr[L \in \mathcal{L}_s] \\
&= \sum_{s \in A} \mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \;\middle|\; L \in \mathcal{L}_s \right] \cdot \Pr[L \in \mathcal{L}_s] \\
&\quad + \sum_{s \in B} \mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \;\middle|\; L \in \mathcal{L}_s \right] \cdot \Pr[L \in \mathcal{L}_s].
\end{aligned}$$

Note that

$$\begin{aligned}
\mathop{\mathrm{E}}_{k,a,b} \left[|A^L| - 1 \;\middle|\; L \in \mathcal{L}_s \right] &= \mathop{\mathrm{E}}_{k-1,a-1,b} \left[|A^L| \right] \text{ for } s \in A, \\
\mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \;\middle|\; L \in \mathcal{L}_s \right] &= \mathop{\mathrm{E}}_{k-1,a,b-1} \left[|A^L| \right] \text{ for } s \in B,
\end{aligned}$$

holds, and by assumption we have

$$\mathop{\mathrm{E}}_{k-1,a-1,b} \left[|A^L| \right] = \frac{(k-1)(a-1)}{a+b-1} \text{ and } \mathop{\mathrm{E}}_{k-1,a,b-1} \left[|A^L| \right] = \frac{(k-1)a}{a+b-1}.$$

Therefore, we have

$$\begin{aligned}
\mathop{\mathrm{E}}_{k,a,b} \left[|A^L| \right] &= \sum_{s \in A} \left(\frac{(k-1)(a-1)}{a+b-1} + 1 \right) \cdot \Pr[L \in \mathcal{L}_s] + \sum_{s \in B} \left(\frac{(k-1)a}{a+b-1} \right) \cdot \Pr[L \in \mathcal{L}_s] \\
&= a \cdot \left(\frac{(k-1)(a-1)}{a+b-1} + 1 \right) \cdot \frac{1}{a+b} + b \cdot \frac{(k-1)a}{a+b-1} \cdot \frac{1}{a+b} \\
&= \frac{a(k-1)(a-1) + a(a+b-1)}{(a+b-1)(a+b)} + \frac{ab(k-1)}{(a+b-1)(a+b)} \\
&= \frac{a(k-1)\left((a-1)+b\right) + a(a+b-1)}{(a+b-1)(a+b)} \\
&= \frac{\left(a(k-1)+a\right)(a+b-1)}{(a+b-1)(a+b)} = \frac{ka}{a+b},
\end{aligned}$$

which completes the proof.

D Constructing Quantum Circuit of BL

This section shows that the quantum circuit BL can be constructed so that it runs in time $O(\lg N^{1/3^{l-1}})$, using $O(N^{1/3^{l-1}})$ qubits. We regard that primary operations on $\lg_2 N$-bit strings such as NOT, AND, OR, and XOR take unit time. In the following, we regard that a function f corresponds to a quantum circuit that calculates $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$.

The quantum circuit BL is constructed as illustrated in Fig. 4, and consists of three kinds of gates: Expand, J, and $\mathsf{OR}_{\mathsf{all}}$ (See Fig. 5). Expand: $Y \to Y^{\times N^{1/3^{l-1}}}$ is the iteration function Expand: $y \mapsto (y, \ldots, y)$, and J: $Y^{\times N^{1/3^{l-1}}} \to \{0,1\}^{\times N^{1/3^{l-1}}}$ is defined by

$$J\colon \left(y_1, \cdots, y_{N^{1/3^{l-1}}}\right) \mapsto \left(J_1(y_1), \ldots, J_{N^{1/3^{l-1}}}(y_{N^{1/3^{l-1}}})\right),$$

where $J_i\colon Y \to \{0,1\}$ is the function defined by $J_i(y) = 1$ if and only if $y = y^{(i)}$. The binary function $\mathsf{OR}_{\mathsf{all}}\colon \{0,1\}^{\times N^{1/3^{l-1}}} \to \{0,1\}$ calculates the OR of all $N^{1/3^{l-1}}$ input bits.

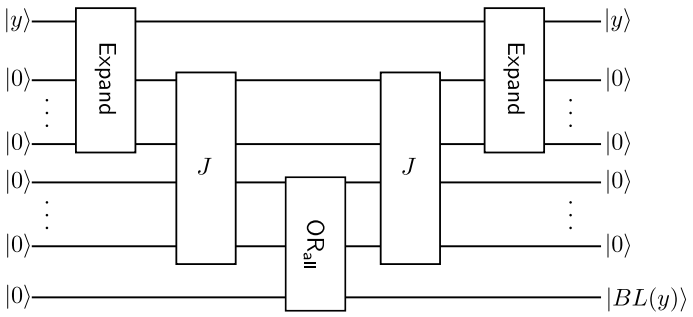

Fig. 4: Quantum circuit of BL.

The circuit of BL illustrated in Fig. 5 runs as follows. The input string y is first sent to the gate Expand, which expands y to $(y, \ldots, y)$ and the output is sent to J. Recall that $y^{(i)}$ is in the list of collisions $c^{(i)} = (\{x_1, \ldots, x_{l-1}\}, y^{(i)})$ that is made by **MColl**. The gate J runs gates J_i in parallel, each of which can access the data $y^{(i)}$ and checks whether y equals $y^{(i)}$, and sends outputs to $\mathsf{OR}_{\mathsf{all}}$. The output of J is sent to $\mathsf{OR}_{\mathsf{all}}$, which calculates $b_1 \vee \cdots \vee b_{N^{1/3^{l-1}}}$, here each b_i corresponds to the output of J_i. Consequently, $\mathsf{OR}_{\mathsf{all}}$ outputs 1 if and only if there exists an index i such that $y = y^{(i)}$, which is the value $BL(y)$. The last two gates, J and Expand, are used to reset ancilla qubits.

Next, we prove that Expand, J, and $\mathsf{OR}_{\mathsf{all}}$ run in $O(\lg N^{1/3^{l-1}})$ time, using $O(N^{1/3^{l-1}})$ qubits. Note that if a function can be *classically* calculated by a boolean circuit that has a binary tree structure of depth D and width W, then it can be implemented as a quantum circuit that runs in time $O(D)$, using $O(W)$ qubits. This quantum circuit runs primary gates such as NOT, AND, OR, and XOR up to $O(W)$ in parallel and operates $O(D)$ steps.

The function $\mathsf{OR}_{\mathsf{all}}$ can be classically calculated by a boolean circuit that has a binary tree structure, which has leaves corresponding to inputs and a root corresponding to output, with depth $O(\lg N^{1/3^{l-1}})$ and width $O(N^{1/3^{l-1}})$. Thus, $\mathsf{OR}_{\mathsf{all}}$ can be constructed

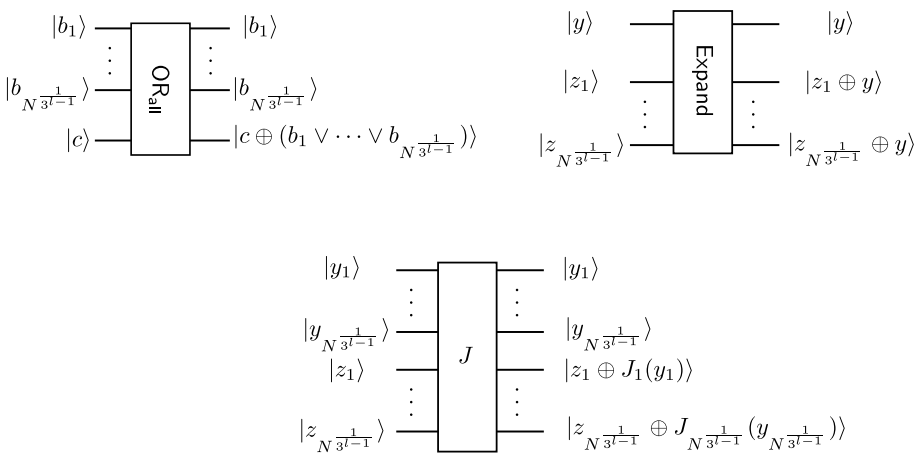

Fig. 5: Quantum circuits of $\mathsf{OR}_{\mathsf{all}}$, Expand, and J.

as a *quantum* circuit that runs in time $O(\lg N^{1/3^{l-1}})$, using $O(N^{1/3^{l-1}})$ qubits. As for J, it calculates all J_i at the same time in parallel. Since each J_i can be classically calculated in time $O(1)$ using $O(1)$ primary operations, the quantum circuit of J_i can be constructed so that it runs in time $O(1)$, using $O(1)$ qubits. Therefore, J runs in time $O(1)$, using $O(N^{1/3^{l-1}})$ qubits. The function Expand can also be classically calculated using a binary tree structure, which has a root corresponding to the input and leaves corresponding to the output, with depth $O(\lg N^{1/3^{l-1}})$ and width $O(N^{1/3^{l-1}})$. We first copy the input y to obtain (y, y), and then copy (y, y) to obtain (y, y, y, y). Repeating the copy procedure for $O(\lg N)$ times, we can obtain the output of Expand. , the quantum circuit Expand can also be constructed so that it runs in time $O(\lg N^{1/3^{l-1}})$, using $O(N^{1/3^{l-1}})$ qubits. Note that, in quantum circuits, this copy procedure is realized as XOR gate $\mathsf{XOR}\colon\ |y\rangle|0\rangle \mapsto |y\rangle|y\rangle$, and does not contradict the no-cloning theorem.