Quantum security of subset cover problems[*]

Samuel Bouaziz–Ermann[†], Alex B. Grilo[‡] and Damien Vergnaud [§]
LIP6, Sorbonne Université, CNRS

Abstract

The subset cover problem for $k \geq 1$ hash functions, which can be seen as an extension of the collision problem, was introduced in 2002 by Reyzin and Reyzin to analyse the security of their hash-function based signature scheme HORS. The security of many hash-based signature schemes relies on this problem or a variant of this problem (e.g. HORS, SPHINCS, SPHINCS+, . . .).

Recently, Yuan, Tibouchi and Abe (2022) introduced a variant to the subset cover problem, called restricted subset cover, and proposed a quantum algorithm for this problem. In this work, we prove that any quantum algorithm needs to make $\Omega\left(k^{-\frac{2^{k-1}}{2^k-1}} \cdot N^{\frac{2^{k-1}-1}{2^k-1}}\right)$ queries to the underlying hash functions to solve the restricted subset cover problem, which essentially matches the query complexity of the algorithm proposed by Yuan, Tibouchi and Abe.

We also analyze the security of the general (r,k)–subset cover problem, which is the underlying problem that implies the unforgeability of HORS under a r-chosen message attack (for $r \geq 1$). We prove that a generic quantum algorithm needs to make $\Omega\left(N^{k/5}\right)$ queries to the underlying hash functions to find a $(1,k)$–subset cover. We also propose a quantum algorithm that finds a (r,k)–subset cover making $O\left(N^{k/(2+2r)}\right)$ queries to the k hash functions.

1 Introduction

Cryptographic hash functions are functions mapping arbitrary-length inputs to fixed-length outputs and are one of the central primitives in cryptography. They serve as building blocks for numerous cryptographic primitives such as key-establishment, authentication, encryption, or digital signatures. In particular, *one-time signatures* – i.e. in which the signing key can be used only once – based only on hash functions were proposed by Lamport as soon as in 1979 [10]. The basic idea is to evaluate a cryptographic hash function on secret values to generate the public verification key and to authenticate a single message by revealing a subset of those secret pre-images.

With the development of quantum technologies, which may bring drastic attacks against widely deployed cryptographic schemes based on the hardness of integer factorization or the discrete logarithm in Abelian groups [13], hash-based signatures have regained interest within the realm of ”post-quantum” cryptography and the recent NIST standardization process. In particular, the SPHINCS+ candidate [3] has been selected in 2022 for standardization by NIST and other constructions are standardized by IETF/IRTF. The SPHINCS+ signature scheme and its predecessor SPHINCS [2] make use of a Merkle-hash tree and of HORST, a variant of a hash-based scheme called HORS [12]. HORS (for “Hash to Obtain Random Subset”) uses a hash function to select the subset of secret pre-images to reveal in a signature and the knowledge of these secrets for several subsets may not be enough to produce a forgery, a property that makes HORS a *few-time signature* scheme.

[*] This work was partially funded by PEPR integrated project EPiQ ANR-22-PETQ-0007 part of Plan France 2030.

[†] samuel.bouaziz@ens-rennes.fr

[‡] Alex.Bredariol-Grilo@lip6.fr

[§] damien.vergnaud@lip6.fr

More concretely, the security of HORS (and HORST) relies on the hardness of finding a subset cover (SC) for the underlying hash function. More formally, to define the (r,k)–*SC problem*, we consider the hash function as the concatenation of $k \geq 1$ hash functions h_1, …, h_k (with smaller outputs) and the problem is to find, for some integer $r \geq 1$, $r+1$ elements $x_0, x_1 \ldots, x_r$ in the hash function domain such that $x_0 \notin \{x_1, \ldots, x_r\}$, and

$$\{h_i(x_0) | 1 \leq i \leq k\} \subseteq \bigcup_{j=0}^{r} \{h_i(x_j) | 1 \leq i \leq k\}.$$

The hardness of this problem for concrete popular hash functions has not be studied in depth but Aumasson and Endignoux [1] proved in 2017 a lower bound on the number of queries to hash functions for the SC problem in the Random Oracle Model (ROM). However, the exact security of HORS (and more generally HORST, SPHINCS and SPHINCS+) with respect to quantum attacks is still not clear. Since quantum computing provides speedups for many problems (e.g. Grover's search algorithm [8] and Brassard, Høyer, and Tapp [6] collision search algorithm), it is important to provide lower bounds in a quantum world.

1.1 Our results

In this paper, we explore the difficulty of finding subset cover for idealized hash functions for quantum algorithms. We also consider a variant called the k-restricted subset cover (k–RSC) problem where, given k functions $h_1, \ldots, h_k$, one has to find $k+1$ elements $x_0, x_1 \ldots, x_k$ such that:

$$\forall 1 \leq i \leq k, h_i(x_0) = h_i(x_i)$$

and $x_0 \notin \{x_1, \ldots, x_r\}$. This variant was defined recently by Yuan, Tibouchi and Abe [14], who showed a quantum algorithm to solve it. The main contributions of this work are:

1. **Lower bound on k–RSC**: we prove that $\Omega\left(k^{-\frac{2^{k-1}}{2^k-1}} \cdot N^{\frac{2^{k-1}-1}{2^k-1}}\right)$ quantum queries to the idealized hash functions are needed to find a k–RSC with constant probability.
(Theorem 16)

2. **Lower bound on $(1,k)$–SC**: we prove that $\Omega\left((k!)^{-1/5} \cdot N^{k/5}\right)$ quantum queries to the idealized hash functions are needed to find a $(1,k)$–SC with constant probability.
(Theorem 23)

3. **Upper bound on (r,k)–SC:** we present a quantum algorithm that finds (r,k)–SC with constant probability with $O\left(N^{k/(2+2r)}\right)$ queries to the hash functions when k is divisible by $r+1$, and $O\left(N^{k/(2+2r)+1/2}\right)$ otherwise.
(Theorem 32)

1.2 Technical Overview

To prove our lower bounds on the query complexity, we use the technique called *compressed random oracle model* introduced by Zhandry in [15]. Its goal is to record information about the queries of an adversary A in the quantum random oracle model and permits "on-the-fly" simulation of random oracles (or *lazy sampling*) by considering the uniform superposition of all possible random oracles instead of picking a single random oracle at the beginning of the computation. The technique uses a register to keep a record of a so-called *database* of the random oracle and this register is updated whenever A makes a query to the random oracle. At the end of A's computation, the reduction can measure the register of the database, and the distribution of the outputs is uniformly random, as if we had chosen a random oracle at the beginning of its computation. This new register that contains the database is at the gist of our lower bounds.

In Section 3, we prove the lower bound on the query complexity to solve the RSC problem. We consider an algorithm A after i quantum queries to the random oracle and call its state at this moment $|\psi_i\rangle$. Our goal is to compute an upper bound for the value $|P_k^{RSC}|\psi_i\rangle|^2$, where P_k^{RSC} is the projection onto the databases that contain a k–RSC. Computing such a bound leads to a lower bound on the number of queries needed for solving k–RSC with constant probability. To prove our bound, we proceed by induction: assuming we proved a bound for the k'–RSC problem for all $k' < k$, we prove a bound for the k–RSC problem. The analysis is naturally divided into two parts: whenever A finds a k–RSC after i quantum queries, it means that either:

1. A finds it after $i-1$ quantum queries;

2. or A finds it with the i^{th} quantum query.

The first case is recursive and it remains to bound $|P_k^{RSC}|\psi_i\rangle|$ in the second case. Here, the database (after $i-1$ quantum queries) must contain a certain number of k'–RSC (for some $k' < k$), in order for A to find k–RSC with the i^{th} query. Using this strategy, we obtain a recursive formula from which we can deduce the bound on $|P_k^{RSC}|\psi_i\rangle|$.

In Section 4.1, we prove a lower bound for the $(1,k)$–SC problem. The idea of the proof is similar to the proof for the lower bound of the k–RSC problem but we have to compute a bound for another problem that we define: the *j–repetition* problem.

Finally in Sections 4.2 and 4.3, we design a family of quantum algorithms for finding a (r,k)-–SC. These algorithms are inspired by the algorithm from [14] to solve the k–RSC problem and [11]'s algorithm for finding multi-collisions. These algorithms are recursive and take as input two parameters $t, k' \in \mathbb{N}$ and perform the following:

1. Find t distinct $(r-1,k')$–SC;

2. Find the (r,k)–SC.

The parameters t and k' are chosen in order to optimize the complexity of the algorithm. The first step is done by applying $r-1$ times the algorithm for the value k', and the second step uses Grover's algorithm.

1.3 Related works, discussion and open problems

Collision-finding. The link between finding a multi-collision and finding a subset cover was first discussed in [14], since their algorithm is inspired from the one for finding multi-collisions in [11]. In the latter, they also show a lower bound for finding multi-collisions, and our proof of lower bounds uses the same technique they used. We make use of the compressed oracle technique, first introduced by Zhandry in [15], and generalize the proof of the lower bound on multi-collisions to the RSC and SC problems.

Restricted Subset Cover. There is currently only one quantum algorithm for finding RSC [14]. Our lower bound for finding a RSC matches their upper bound when k, the number of functions, is constant. However when k is not a constant, their algorithms makes $O\left(k \cdot N^{\frac{2^{k-1}-1}{2^k-1}}\right)$ queries to

$h_1, \dots, h_k$, which roughly leaves a $k^{3/2}$ gap between the best known attack and our lower bound. To the best of our knowledge, this is the first lower bound on the RSC problem for a quantum algorithm, and there are no such result for classical algorithms. It would be interesting to see if we can close this gap further.

Tighter bounds for $(1,k)$–SC. When k is constant, the lower bound for $(1,k)$–SC is $\Omega\left(N^{k/5}\right)$, while our algorithm for this problem makes $O\left(N^{k/4}\right)$ queries to the oracle. It would be interesting to tighten this gap, especially since the results for $(1,k)$–SC are probably necessary to prove the lower bounds (r,k)–SC for $r \geq 2$.

For non-constant k, our lower bound for $(1,k)$–SC is $\Omega\left(C_k^{-1/5} \cdot N^{k/5}\right)$, where $C_k = \sum_{j=2}^{k} \frac{k!}{(j-1)!} \leq k! \cdot e$. Notice that this term cannot be neglected for large values of k. For example with $k = \log(N)$, we have $C_k \geq N$. In comparison, our best algorithm for $(1,k)$–RSC, the factor in k is $\binom{k}{(k+1)/2}^{-1/2} \leq \frac{2^{(k+1)/2}}{\left(\frac{k+1}{2}\cdot\pi\right)^{1/4}}$, which is very far from our bound on C_k. It would also be interesting to see if we can tighten this gap.

Bounds for (r,k)–SC. Unfortunately, expanding our result for the (r,k)–SC problem is much more complicated than the case $r = 1$ and actually even proving the case $r = 2$ is not simple. To prove such a result, one would need a bound for the problem of finding j distinct $(1,k)$–SC problem. While proving such a bound is challenging, it is also unclear what the problem of finding j distinct $(1,k)$–SC is. Indeed, an important property for our technique in the first lower bound proofs is that by making one query to the oracle, the adversary cannot find two or more k–RSC. The same property must hold for the problem of finding j distinct $(1,k)$–SC, and this definition and subsequent analysis remain open.

Security of SPHINCS and SPHINCS+. The signature scheme SPHINCS relies on the HORST scheme (for "HORS with trees") which adds a Merkle tree to the HORS scheme to compress the public key. The security of HORST also relies on the (r,k)–SC problem but the security of SPHINCS rely on different security notions of the underlying hash functions. In particular, it depends on a variation of the SC problem classed the *target subset cover* (TSC) problem [12]. The main difference comes from the fact that the message signed using HORST is an unpredictable function of the actual message and this prevents an attacker to construct a subset cover beforehand.

Nevertheless, the authors of [2] stated an existential unforgeability result for SPHINCS [2, Theorem 1] under q_s-adaptive chosen message attacks. The success probability in such attacks is roughly upper-bounded by:

$$\sum_{r=1}^{\infty} \min\left(2^{r(\log q_s - h)+h}, 1\right) \cdot Succ_A((r,k) - SC),$$

where h is the height of the tree used in SPHINCS, and $Succ_A((r,k) - SC)$ denotes the success probability of an adversary A to find a (r,k)–SC. The authors made the assumption that this term is negligible for any probabilistic adversary A and our quantum lower bound on the query number to find a $(1,k)$–SC can be seen as a first step towards proving this assumption (for idealized hash functions). To assess the security of SPHINCS from [2, Theorem 1] for concrete parameters such as those proposed in [2] (namely $h = 60, q_s = 2^{30}$), it would also be necessary to upper-bound the success probabilities $Succ_A((2,k) - SC)$ and $Succ_A((3,k) - SC)$, which we leave for future work.

SPHINCS+ is an enhancement of SPHINCS, which makes the scheme more efficient and its security relies on another variant of the SC problem, namely the interleaved target subset cover

(ITSC) problem. It would also be interesting to see if our methods can be used to prove similar bounds for the TSC and ITSC problems. At last, one could also try to design algorithms for these two problems, as no quantum algorithms for them exist yet to the best of our knowledge.

2 Preliminaries

We assume the reader is familiar with the theory of quantum information and for completeness, we recall Grover's algorithm and Quantum Fourier Transform (QFT) in Appendix A. We denote the concatenation by $||$.

2.1 Compressed oracle technique

We now present the key ingredients of Zhandry's compressed oracle technique, first defined in [15] and refined in [7].

We consider the *Quantum Random Oracle Model*, first defined in [4]. In this model, we are given black-box access to a *random* function $H : \mathcal{X} \to \mathcal{Y}$. For our model, the adversary will work on three different registers $|x, y, z\rangle$. The first register is the query register, the second register is the answer register and the third register is the work register. The first two registers are used for queries and answers to the oracle, while the last register is for the adversary's other computations. We first define the unitary StO that represents the *Standard Oracle* and that computes as follows:

$$StO \sum_{x,y,z} \alpha_{x,y,z} |x, y, z\rangle \to \sum_{x,y,z} \alpha_{x,y,z} |x, y + H(x), z\rangle$$

This unitary corresponds to a query to H.

Now, we define Zhandry's compressed oracle. In this model, instead of starting with a random function H, we start with the uniform superposition of all random functions $|H\rangle$, where $|H\rangle$ encodes the truth table of the function H. Let $\mathcal{H} = \{H : \mathcal{X} \to \mathcal{Y}\}$ be the set of all possible functions H. We define a new register, the database register $|H\rangle$, that starts in the uniform superposition $\frac{1}{|\mathcal{H}|} \sum_{H\in\mathcal{H}} |H\rangle$. This register starts in product state with the other registers, and Zhandry's idea is that instead of modifying the adversary's register when querying the oracle, we will modify the database register instead. To do so, we simply consider the *Fourier basis* for the y and the H register before querying the Standard Oracle.

We write this unitary O and it works as follows:

$$O \sum_{x,y,z} \alpha_{x,y,z} |x, y, z\rangle \otimes \sum_{H\in\mathcal{H}} \alpha_H |H\rangle \to \sum_{x,y,z} \alpha_{x,y,z} |x, y, z\rangle \otimes \sum_{H\in\mathcal{H}} \alpha_H |H \oplus (x, y)\rangle$$

This unitary can be implemented by applying the QFT to the registers $|y\rangle$ and $|H\rangle$, applying the Standard Oracle, then applying the $QFT^\dagger$ again on the $|y\rangle$ and $|H\rangle$ registers.

Finally, we define the compression part. The idea behind the compression is that for every x in the database mapped to $|0\rangle$, we remap it to $|\perp\rangle$, a new value outside of $\mathcal{Y}$. More formally, the compression part is done by applying:

$$Comp = \bigotimes_x \left(|\perp\rangle \langle 0| + \sum_{y:y\neq 0} |y\rangle \langle y| \right)$$

in the Fourier basis.

Since at the start of the computation, the database will be initiated with the uniform superposition over all $\mathcal{H}$ possible, then after q queries the state of the database can be described with q vectors. In order to apply the compression as a unitary, we declare that $Comp\,|\bot\rangle = |0\rangle$.

Now, we can define the *Compressed Oracle*:

$$cO = Comp \circ O \circ Comp^{\dagger}.$$

Of course the compression part inevitably creates some losses, compared to only using the Standard Oracle. The precise characterization of these losses is given in one of Zhandry's lemma, and can be stated as follows:

Lemma 1 (Zhandry)**.** *Let A be an algorithm that makes queries to a random oracle $H : \mathcal{X} \to \mathcal{Y}$, and output $(x_1, \ldots, x_k, y_1, \ldots, y_k) \in \mathcal{X}^k \times \mathcal{Y}^k$. Let p be the probability that $\forall 1 \leq i \leq k$, $H(x_i) = y_i$. Similarly, consider the algorithm A running with the Compressed Oracle cO, and output $(x'_1, \ldots, x'_k, y'_1, \ldots, y'_k) \in \mathcal{X}^k \times \mathcal{Y}^k$. Let p' be the probability that $\forall 1 \leq i \leq k$, $H(x'_i) = y'_i$. Then:*

$$\sqrt{p} \leq \sqrt{p'} + \sqrt{\frac{k}{|\mathcal{Y}|}}$$

In the rest of the paper, we will have that $\sqrt{\frac{k}{|\mathcal{Y}}}$ is negligible, and thus we will neglect this term.

In the following, we will model the adversary (A) as a series of computation alternating between unitaries and oracle calls. The adversary's quantum state will first be initialized to $|0\rangle^{\otimes N}$. Then, his computation will be decomposed as:

$$A = U_k cO U_{k-1} cO \ldots cO U_2 cO U_1 \tag{1}$$

So that, if $\psi_i = \sum_{x,y,z,D} \alpha_{x,y,z,D} |x, y, z, D\rangle$ is the state of the adversary after i quantum queries to cO, then U_{i+1} operates on the registers x, y and z only. We also define *database properties*:

Definition 2 (Database property)**.** *A database property is a subset of $\mathcal{H}$. Any database property D can be seen as a projector on $\mathcal{H}$, as follows:*

$$\sum_{d \in D} |d\rangle \langle d|$$

We write $\mathcal{D} = \{I | I \subseteq \mathcal{H}\}$ the set of all subspaces of $\mathcal{H}$, that also corresponds to the set of all database properties.

We know state and prove two lemmas adapted from [11] that we will use thoroughly in this paper. The first lemma will allow use to ignore the unitaries that the adversary A apply on the first registers of the state.

Lemma 3 (adapted from Lemma 8 from [11])**.** *For any unitary U, any projector P, and any state $|\phi\rangle$,*

$$|(I \otimes P) \cdot (U \otimes I) |\phi\rangle| = |(I \otimes P) |\phi\rangle|$$

The second lemma bounds the amplitude of measuring a database that satisfies a property P at the i^{th} step of the algorithm, i.e. just after the i^{th} query to the oracle. In this bound, the first term captures the case where we succeed to find a database that satisfies P before the i^{th} query. The second term capture the case where we did not have it before the i^{th} query, but found it with the i^{th} one.

Lemma 4 (adapted from Lemma 9 from [11])**.** *Let $|\phi_i\rangle$ be the state of an algorithm A just before the i^{th} quantum query to cO, and $|\psi_i\rangle$ the state of the same algorithm right after the i^{th} quantum query to cO. Let P be any projector on D. We have that:*

$$|P\,|\psi_i\rangle\,| \leq |P\,|\phi_i\rangle| + |PcO(I-P)\,|\phi_i\rangle|$$

Proof.

$$\begin{aligned}|P\,|\psi_i\rangle| = |PcO\,|\phi_i\rangle| &= |PcO(P\,|\phi_i\rangle + (I-P)\,|\phi_i\rangle)| \\ &\leq |P\,|\phi_i\rangle| + |PcO(I-P)\,|\phi_i\rangle)|\,.\end{aligned}$$

□

2.2 The problem of subset cover and its variants

We define the problem of subset cover, given two spaces $\mathcal{X}$ and $\mathcal{Y}$.

Definition 5 ((r,k)–SC)**.** *Let $k, r \in \mathbb{N}^*$. Let $h_1, \cdots, h_k : \mathcal{X} \to \mathcal{Y}$. A* (r,k)–SC *for $(h_1, \cdots, h_k)$ is a set of $r+1$ elements $x_0, x_1, x_2, \cdots, x_r$ in $\mathcal{X}$ such that:*

$$\{h_i(x_0)|1 \leq i \leq k\} \subseteq \bigcup_{j=1}^{r} \{h_i(x_j)|1 \leq i \leq k\}$$

In other words, for each $1 \leq i \leq k$, there exists a $1 \leq j \leq r$ and a $1 \leq \ell \leq k$ such that $h_i(x_0) = h_\ell(x_j)$.

We notice two facts regarding the parameters of (r,k)–SC. First, we have that the problem becomes easier when r increases. Secondly, we have that when $r > k$, a (r,k)–SC contains a (k,k)-–SC. Thus finding a (r,k)–SC when $r > k$ is the same as when $r = k$. For simplicity, we use k–SC as a shorthand of (k,k)–SC.

We also define the database properties $P^{SC}_{(r,k)}$ of containing a (r,k)–SC, that is the set of databases that contains a (r,k)–SC. More formally, we have that:

$$P^{SC}_{(r,k)} = \left\{ D \in \mathcal{D} \,\middle|\, \exists x_0, x_1, \ldots, x_r, \forall i \neq 0, x_0 \neq x_i, H(x_0) \subseteq \bigcup_{i=1}^{r} H(x_i) \right\},$$

where for $x \in \mathcal{X}, H(x) = \{h_1(x), \ldots, h_k(x)\}$.

We follow now with the definition of a harder variation of the k–subset cover called the *k–restricted subset cover* (k–RSC).

Definition 6 (k–RSC)**.** *Let $k \in \mathbb{N}^*$. Let $h_1, \ldots, h_k : \mathcal{X} \to \mathcal{Y}$. A* k–restricted subset cover *(k–RSC) for $(h_1, \ldots, h_k)$ is a set of $k+1$ elements $x_0, x_1, x_2, \ldots, x_k$ in $\mathcal{X}$ such that:*

$$\forall i \in \{1, \ldots, k\}, h_i(x_0) = h_i(x_i) \text{ and } x_0 \neq x_i.$$

We also define the database properties $P^{RSC}_{k,\ell}$ of k distinct ℓ–RSC, that is the set of databases that contains k distinct ℓ–RSC. More formally, we have that:

$$P^{RSC}_{k,\ell} = \left\{ D \in \mathcal{D} \,\middle|\, \begin{array}{l} \exists x_{0,1}, \ldots, x_{\ell,1}, \forall i \neq 0, x_{0,1} \neq x_{i,1}, \forall i, h_i(x_{0,1}) = h_i(x_{i,1}) \\ \exists x_{0,2}, \ldots, x_{\ell,2}, \forall i \neq 0, x_{0,2} \neq x_{i,2}, \forall i, h_i(x_{0,2}) = h_i(x_{i,2}) \\ \vdots \\ \exists x_{0,\ell}, \ldots, x_{\ell,k}, \forall i \neq 0, x_{0,k} \neq x_{i,k}, \forall i, h_i(x_{0,k}) = h_i(x_{i,k}) \\ \forall i \neq j, (h_1(x_{0,i}), \ldots, h_\ell(x_{0,i})) \neq (h_1(x_{0,j}), \ldots, h_\ell(x_{0,j})) \end{array} \right\} \tag{2}$$

This problem was introduced in [14], in which the authors describe an algorithm that finds a k–RSC in $O\left(kN^{\frac{1}{2}\left(1-\frac{1}{2^{k+1}-1}\right)}\right)$ quantum queries to $h_1, \ldots, h_k$ when the h_i's are such that $|\mathcal{X}| \geq (k+1)|\mathcal{Y}|$.

We discuss now the last condition in Equation (2). We remark that while such condition was not explicitly imposed in [11] for their lower bound for finding multi-collisions, this property is implicitly and extensively used in their proof. Such a property is needed because when they count k–collisions (that is, k distinct $x_1, \ldots, x_k$ such that $H(x_1) = \cdots = H(x_k)$), they are actually interested in the number of possible *images* that would be helpful to reach a $(k+1)$–collision. In particular, this is helpful since one query can only transform *one* k–collision (with such a property) into a $(k+1)$–collision.

In our case, the last line of (2) ensures that the "supporting set" of the k–RSC (i.e. the set of images of the $x_{0,i}$ by the different random functions $h_1, \ldots, h_k$) is unique. As in the multi-collision case, this condition will be crucial to extend k–RSC to a $(k+1)$–RSC, and for this reason we define it explicitly in $P^{RSC}_{k,\ell}$.

Finally, we state a result from [11], regarding the amplitude of finding j distinct 2–collisions:

Lemma 7 (adapted from [11], Corollary 11)**.** *Let $f^{col}_{i,j}$ be the amplitude of the D containing at least j distinct 2–collisions after i quantum queries. Then:*

$$f^{col}_{i,j} \leq \left(\frac{e \cdot i^{3/2}}{j\sqrt{N}}\right)^j$$

3 Lower bound on the k–restricted subset cover problem

In this section, we prove a lower bound for the k–RSC problem defined in Definition 6. This section follows closely [11]'s proof of their lower bound on finding multi-collisions. We will first prove a lower bound for the problem when $k = 2$. Then, we will prove a lower bound for finding ℓ distinct 2–RSC, which will be necessary in our induction step. Finally, we will prove the induction step in the last subsection and obtain a lower bound on finding s distinct k–RSC.

3.1 Finding a 2–restricted subset cover

In this section, we will prove that the number of queries necessary to find a 2–RSC is $\Omega(N^{3/7})$, matching the query complexity of the quantum algorithm proposed in [14], up to a constant factor.

As presented in Definition 6, in the 2–RSC problem, we are given 2 random functions h_1, h_2 such that for $i \in \{1, 2\}$, $h_i : \mathcal{X} \to \mathcal{Y}$. The main theorem of this subsection can be stated as follows:

Theorem 8. *Given two random functions $h_1, h_2 : \mathcal{X} \to \mathcal{Y}$ where $|N| = \mathcal{Y}$, a quantum algorithm needs to make $\Omega(N^{3/7})$ queries to h_1 and h_2 to find a 2–RSC with a constant probability.*

In order to prove this theorem, we first introduce some database properties:

- $P'_{\ell-col-h_1}$ corresponds to the set of databases that contain *at least ℓ distinct* collisions on h_1.[1] As explained in the previous section, here we will use the fact that we cannot reach a

[1] We do not define the equivalent property for h_2. Since both h_1 and h_2 are random functions, we can swap them when considering database property by symmetry, thus we do not need to define more unnecessary properties.

database containing $\ell+2$ or more collisions from a database containing ℓ collisions by making a single query:

$$P'_{\ell-col-h_1} = \left\{ D \in \mathcal{D} \middle| \begin{array}{l} \exists x_1, \dots, x_\ell, y_1, \dots, y_\ell, \forall i, h_1(x_i) = h_1(y_i) \neq \perp \\ \forall i, x_i \neq y_i \\ \forall i \neq j, (h_1(x_i), h_2(x_i)) \neq (h_1(x_j), h_2(x_j)) \end{array} \right\}$$

- $P_{\ell-col-h_1}$ corresponds to the set of databases that contain *exactly ℓ distinct* collisions on h_1:

$$P_{\ell-col-h_1} = P'_{\ell-col-h_1} \cap \neg P'_{(\ell+1)-col-h_1}$$

- $P_{preimage-h_1}$ corresponds to the set of databases that contain the preimage of 0:[2]

$$P_{preimage-h_1} = \{D \in \mathcal{D} | \exists x, h_1(x) = 0\}$$

Finally, for $i, \ell \in \mathbb{N}$, we write:

$$\widetilde{f}^{col}_{i,\ell} = \left|P_{\ell-col-h_1} \left|\psi_i\right\rangle\right|, f^{col}_{i,\ell} = \left|P'_{\ell-col-h_1} \left|\psi_i\right\rangle\right|, g_i = \left|P^{RSC}_{1,2} \left|\psi_i\right\rangle\right|, \tag{3}$$

where $|\psi_i\rangle$ is the state just after the i^{th} query to $H = (h_1, h_2)$ and $P^{RSC}_{1,2}$ was defined in Equation (2). For convenience, we write $P_2 = P^{RSC}_{1,2}$ in this section.

The goal here is to bound the term g_i, and to achieve this we first prove a recursive formula that involves $\widetilde{f}^{col}_{i,\ell}$ as well:

Lemma 9. *For every $i \in \mathbb{N}$, we have that:*

$$g_i \leq g_{i-1} + \sqrt{2 \sum_{\ell \geq 0} \frac{\ell}{N} \widetilde{f}^{col}_{i-1,\ell}{}^2} + \frac{i-1}{N}. \tag{4}$$

Proof. Let $i \in \mathbb{N}$. Let $|\phi_i\rangle$ be the state just before the i^{th} query to $H = (h_1, h_2)$, namely

$$|\phi_i\rangle = \sum_{x,y,z,D} \alpha_{x,y,z,D} \left|x, y, z\right\rangle \otimes \left|D\right\rangle,$$

where x is the query register, y is the answer register, z is the work register and D is the database register. Let $|\psi_i\rangle$ be the state right after the i^{th} query to H, namely

$$|\psi_i\rangle = \sum_{x,y,z,D} \frac{1}{\sqrt{N^2}} \sum_{u'} \omega_n^{uu'} \alpha_{x,y,z,D} \left|x, y, z\right\rangle \otimes \left|D \oplus (x, u')\right\rangle.$$

From Lemma 4, we have that:

$$\left|P_2 \left|\psi_i\right\rangle\right| \leq \left|P_2 \left|\phi_i\right\rangle\right| + \left|P_2 cO(I - P_2) \left|\phi_i\right\rangle\right|. \tag{5}$$

We focus now on bounding the second term:

[2] Note that the amplitude of finding *any* preimage is the same as the amplitude of finding the preimage of 0.

$$
\begin{aligned}
|P_2 cO(I-P_2)\,|\phi_i\rangle| &= \left| P_2 cO \sum_{\substack{x,y,z\\ D\text{: no 2–RSC}}} \alpha_{x,u,z,D}\,|x,u,z,D\rangle \right| \\
&= \left| P_2 \sum_{\substack{x,y,z\\ D\text{: no 2–RSC}}} \frac{1}{\sqrt{N^2}} \sum_{u'} \omega_n^{uu'} \alpha_{x,u,z,D}\,|x,u,z,D\oplus(x,u')\rangle \right| \\
&= \left| \sum_{u'} P_2 \sum_{\substack{x,y,z\\ D\text{: no 2–RSC}}} \frac{1}{\sqrt{N^2}} \omega_n^{uu'} \alpha_{x,u,z,D}\,|x,u,z,D\oplus(x,u')\rangle \right|.
\end{aligned}
$$

We analyse now the possibilities for achieving a 2–RSC, considering the different cases of the inner sum. We have four possible ways to get from D that does not have a 2–RSC to $D_{u'} := D \oplus (x,u')$ that has a 2–RSC.

- $(x = x_2)$ Here, we consider the case where there exists an x_0 and x_1 such that $h_1(x_0) = h_1(x_1)$ and we query x such that $h_2(x_0) = h_2(x)$. If we have found ℓ collisions of h_1 in D, then ℓ values of u' can make $D_{u'}$ contain a 2–RSC, out of the N possible values for the outcome of h_2 (notice that the value of $h_1(x)$ is not relevant for this case).
- $(x = x_1)$ Similar to the previous case, but swapping the roles of h_1 and h_2.
- $(x = x_0)$ Otherwise, we consider the case where we query x such that we have x_1 and x_2 (which might be equal), such that $h_1(x) = h_1(x_1)$ and $h_2(x) = h_2(x_2)$. Only $i-1$ values of u' will make $D_{u'}$ contain a collision on h_1. Similarly, only $i-1$ values of u' will make $D_{u'}$ contain a collision on h_2.

Thus, we have

$$
|P_2 cO(I-P_2)\,|\phi_i\rangle| \le \left(2 \cdot \sum_{\ell \ge 0} \frac{\ell}{N} \left| P_{\ell-col-h_1} |\phi_i\rangle \right|^2 \right)^{1/2} + \frac{(i-1)}{N}, \tag{6}
$$

and we give the details on Equation (6) in Appendix B.1.

Let $|\psi_{i-1}\rangle$ be the state just after the $(i-1)^{th}$ query, and let U_i be the state such that $|\phi_i\rangle = (U_i \otimes I)\,|\psi_{i-1}\rangle$ (see Equation (1)). Note that we also have $|\psi_i\rangle = cO \cdot (U_i \otimes I)\,|\psi_{i-1}\rangle$. Using Lemma 3, we get that:

$$
\begin{aligned}
|P_2 cO(I-P_2)\,|\phi_i\rangle| &\le \sqrt{2 \sum_{\ell \ge 0} \frac{\ell}{N} \left| P_{\ell-col-h_1} (U_i \otimes I)\,|\psi_{i-1}\rangle \right|^2 + \frac{i-1}{N}} \\
&\le \sqrt{2 \sum_{\ell \ge 0} \frac{\ell}{N} \left| P_{\ell-col-h_1} |\psi_{i-1}\rangle \right|^2 + \frac{i-1}{N}}.
\end{aligned} \tag{7}
$$

Similarly, using Lemma 3:

$$\left|P_2 \left|\phi_i\right\rangle\right| = \left|P_2 \left(U_i \otimes I\right) \left|\psi_{i-1}\right\rangle\right| = \left|P_2 \left|\psi_{i-1}\right\rangle\right|. \tag{8}$$

Then, using Equation (5), Equation (7) and Equation (8), and the notation from Equation (3), we have:

$$g_i \leq g_{i-1} + \sqrt{2 \sum_{\ell \geq 0} \frac{\ell}{N} {\widetilde{f}^{col}_{i-1,\ell}}^{2}} + \frac{i-1}{N}.$$

□

We will now expand this recursive formula to obtain a bound on g_i.

Lemma 10. *For every $i \in \mathbb{N}$, we have that:*

$$g_i \leq \sqrt{2} \sum_{j=1}^{i-1} \sqrt{\frac{\mu_3(j)}{N}} + \sqrt{2} \cdot 2^{-9.5 N^{1/8}} + \frac{i^2}{N},$$

where

$$\mu_3(j) = \max\left\{2e \frac{j^{3/2}}{\sqrt{N}}, 10N^{1/8}\right\}.$$

Proof. From Lemma 9, we expand recursively Equation (4), and obtain (using that $g_0 = 0$):

$$g_i \leq \sum_{j=1}^{i-1} \sqrt{2 \sum_{\ell \geq 0} \frac{\ell}{N} {\widetilde{f}^{col}_{j,\ell}}^{2}} + \sum_{j=1}^{i-1} \frac{j}{N}. \tag{9}$$

The second term of Equation (9) can be bounded by

$$\sum_{j=1}^{i-1} \frac{j}{N} \leq \sum_{j=1}^{i-1} \frac{i}{N} \leq \frac{i^2}{N}. \tag{10}$$

As for the first term of Equation (9), we have:

$$\begin{aligned}
\sum_{j=1}^{i-1} \sqrt{2 \sum_{\ell \geq 0} \frac{\ell}{N} {\widetilde{f}^{col}_{j,\ell}}^{2}} &= \sqrt{2} \sum_{j=1}^{i-1} \sqrt{\sum_{\ell=0}^{\mu_3(j)} \frac{\ell}{N} {\widetilde{f}^{col}_{j,\ell}}^{2} + \sum_{\ell > \mu_3(j)} \frac{\ell}{N} {\widetilde{f}^{col}_{j,\ell}}^{2}} \\
&\leq \sqrt{2} \sum_{j=1}^{i-1} \left(\sqrt{\sum_{\ell=0}^{\mu_3(j)} \frac{\ell}{N} {\widetilde{f}^{col}_{j,\ell}}^{2}} + \sqrt{\sum_{\ell > \mu_3(j)} 1 \cdot {\widetilde{f}^{col}_{j,\ell}}^{2}} \right) \\
&\leq \sqrt{2} \sum_{j=1}^{i-1} \left(\sqrt{\frac{\mu_3(j)}{N} \cdot 1} + f^{col}_{j,\mu_3(j)} \right) \\
&\leq \sqrt{2} \left(\sum_{j=1}^{i-1} \sqrt{\frac{\mu_3(j)}{N}} + \sum_{j=1}^{i-1} f^{col}_{j,\mu_3(j)} \right) \qquad (11)
\end{aligned}$$

where in the second inequality, we used the fact that the term $\sum_{\ell>\mu_3(j)} \widetilde{f}^{col}_{j,\ell}{}^{2}$ is equal to the amplitude of finding *at least* $\mu_3(j)$ distinct ℓ–collisions on h_1, thus is exactly equal to $f^{col}_{j,\mu_3(j)}{}^{2}$ (defined in Equation (3)).

It follows that

$$\sum_{j=1}^{i-1} f^{col}_{j,\mu_3(j)} \leq \sum_{j=1}^{i-1} \left(\frac{e\cdot j^{3/2}}{\mu_3(j)\cdot\sqrt{N}}\right)^{\mu_3(j)} \leq \sum_{j=1}^{i-1}\left(\frac{1}{2}\right)^{10N^{1/8}} \leq 2^{-9.5N^{1/8}}, \tag{12}$$

where the first inequality comes from Lemma 7, the second inequality comes from the definition of $\mu_3(j)$ and in the last inequality we assume that $i \leq N^{1/2}$. Indeed, otherwise A can execute [14]'s algorithm whose query complexity for finding a k–RSC is upper-bounded by $O\left(N^{1/2}\right)$.

Putting together Equation (9), Equation (10), Equation (11) and Equation (12) gives the result. □

We can now use Lemma 10 to prove Theorem 8

Proof of Theorem 8. Using Lemma 10, we have for $i \in \mathbb{N}$:

$$g_i \leq \sqrt{2}\sum_{j=1}^{i-1}\sqrt{\frac{\mu_3(j)}{N}} + \sqrt{2}\cdot 2^{-9.5N^{1/8}} + \frac{i^2}{N}.$$

We can bound the first term by:

$$\begin{aligned}
\sqrt{2}\sum_{j=1}^{i-1}\sqrt{\frac{\mu_3(j)}{N}} &= \sqrt{2}\left(\sum_{j:\mu_3(j)=2e\cdot\frac{j^{3/2}}{\sqrt{N}}} \frac{\sqrt{2ej^{3/2}}}{N^{3/4}} + \sum_{j:\mu_3(j)=10N^{1/8}} \frac{\sqrt{10N^{1/8}}}{N^{1/2}}\right)\\
&\leq \sqrt{2}\left(\sum_{j=1}^{i-1}\frac{\sqrt{2ej^{3/2}}}{N^{3/4}} + \sum_{j:\mu_3(j)=10N^{1/8}} \frac{\sqrt{10N^{1/8}}}{N^{1/2}}\right)\\
&\leq 2\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + \left(\frac{10}{2e}\right)^{2/3}\cdot N^{5/12}\cdot\frac{\sqrt{10N^{1/8}}}{N^{1/2}}\\
&\leq 2\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + O(N^{-1/48}),
\end{aligned}$$

where the second inequality comes from counting the number of j such that $\mu_3(j) = 10N^{1/8}$, which is equal to the number of j such that $2e\frac{j^{3/2}}{\sqrt{N}} \leq 10N^{1/8}$.

Thus, we have the following bound on g_i:

$$g_i \leq \sqrt{2}\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + \frac{i^2}{N} + O(N^{-1/48}).$$

So when $i = o(N^{3/7})$, we have $g_i = o(1)$. Hence if we want g_i to be constant, i.e. not $o(1)$, we must have $i = \Omega\left(N^{3/7}\right)$. □

3.2 Finding k distinct 2-restricted subset cover

We are now interested in bounding the number of queries needed to find k distinct triplets that satisfy 2–RSC. We have the following result:

Theorem 11. *Given two random functions $h_1, h_2 : \mathcal{X} \rightarrow \mathcal{Y}$ where $N = |\mathcal{Y}|$, a quantum algorithm needs to make $\Omega(k^{4/7} \cdot N^{3/7})$ queries to h_1 and h_2 to find k distinct 2–RSC with constant probability, for any $k \leq N^{1/8}$.*

To prove this theorem, we first introduce some notation. We denote $P_{2,k,\ell}$ the set of database that satisfies k distinct 2–RSC, and that contain exactly ℓ collisions on h_1. Using the notation from the Section 3.1 and Equation (2), we have that $P_{2,k,\ell} = P^{RSC}_{k,2} \cap P_{\ell-col-h_1}$. We denote $g_{i,k} = \left| P^{RSC}_{k,2} \left|\psi_i\right\rangle \right|$ and $\widehat{g}_{i,k,\ell} = \left| P_{2,k,\ell} \left|\psi_i\right\rangle \right|$, where $\left|\psi_i\right\rangle$ is the state just after the i^{th} query to $H = (h_1, h_2)$.

Our goal is to bound $g_{i,k}$, and as in the previous subsection, we will first prove a recursive formula stated in the next lemma.

Lemma 12. *For every $i \in \mathbb{N}$, and every $k \in \mathbb{N}$, we have that:*

$$g_{i,k} \leq g_{i-1,k} + \sqrt{2 \sum_{\ell \geq 0} \frac{\ell}{N} \widehat{g}^2_{i-1,k-1,\ell}} + \frac{(i-1)}{N} g_{i-1,k-1}.$$

Proof. From Lemma 4, we have the following inequality:

$$\left| P^{RSC}_{k,2} \left|\psi_i\right\rangle \right| \leq \left| P^{RSC}_{k,2} \left|\phi_i\right\rangle \right| + \left| P^{RSC}_{k,2} cO(I - P^{RSC}_{k,2}) \left|\phi_i\right\rangle \right|.$$

And we have that:

$$\begin{aligned}
&\left| P^{RSC}_{k,2} cO(I - P^{RSC}_{k,2}) \left|\phi_i\right\rangle \right| \\
&\leq \left| P^{RSC}_{k,2} \sum_{\substack{x,y,z \\ D\text{:k-1 2-RSC}}} \frac{1}{\sqrt{N^2}} \sum_{u'} \omega_n^{uu'} \alpha_{x,u,z,D} \left| x, u, z, D \oplus (x, u') \right\rangle \right| \\
&\leq \left(2 \sum_{\ell \geq 0} \frac{\ell}{N} \sum_{\substack{x,y,z \\ D\text{:k-1 2-RSC} \\ \ell \text{ collisions} \\ \text{on } h_1}} \left|\alpha_{x,u,z,D}\right|^2 \right)^{1/2} + \left(\frac{(i-1)^2}{N^2} \sum_{\substack{x,y,z \\ D\text{:k-1 2-RSC}}} \left|\alpha_{x,u,z,D}\right|^2 \right)^{1/2} \\
&\leq \left(2 \sum_{\ell \geq 0} \frac{\ell}{N} \left| P_{2,k-1,\ell} \left|\phi_i\right\rangle \right|^2 \right)^{1/2} + \left(\frac{(i-1)^2}{N^2} \left| P^{RSC}_{k-1,2} \left|\phi_i\right\rangle \right|^2 \right)^{1/2},
\end{aligned}$$

where the second equality uses the same cases as for the case $k = 1$ in Lemma 9.

Using Lemma 3 and previous notation (as in Lemma 9), we obtain that:

$$
\begin{aligned}
g_{i,k} &\leq g_{i-1,k} + \left(2\sum_{\ell\geq 0}\frac{\ell}{N}\widehat{g}^2_{i-1,k-1,\ell}\right)^{1/2} + \left(\frac{(i-1)^2}{N^2}g_{i-1,k-1}{}^2\right)^{1/2} \\
&\leq g_{i-1,k} + \left(2\sum_{\ell\geq 0}\frac{\ell}{N}\widehat{g}^2_{i-1,k-1,\ell}\right)^{1/2} + \frac{(i-1)}{N}g_{i-1,k-1}.
\end{aligned}
$$

□

Following the proof from the case $k = 1$, we will split the sum in two using $\mu_3(j)$ as a threshold. We also define a new notation that will simplify expressions:

Definition 13.

$$A_i = \sum_{\ell=0}^{i-1}\sqrt{2}\left(\sqrt{\frac{\mu_3(\ell-1)}{N}} + \frac{\ell-1}{N}\right),$$

where

$$\mu_3(\ell) = \max\left\{2e\frac{\ell^{3/2}}{\sqrt{N}}, 10N^{1/8}\right\}.$$

Before bounding $g_{i,k}$, we first prove a bound on A_i.

Lemma 14. *For every $i \in \mathbb{N}$, we have that:*

$$A_i \leq 2\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + \sqrt{2}\frac{i^2}{N} + O\left(N^{-1/48}\right).$$

It follows that $A_i < 2eN^{1/8}$ for $i \leq N^{1/2}$.

We leave the proof of Lemma 14 to Appendix B.2. We can now state the lemma that bounds $g_{i,k}$.

Lemma 15. *For every $i \in \mathbb{N}$ and $k \in \mathbb{N}$, we have that:*

$$g_{i,k} < \frac{A_i^k}{k!} + \sqrt{2}\cdot 2^{-N^{1/8}}.$$

Proof. We write $f^{col}_{i,j} = \left|P'_{j-col-h_1}\left|\phi_i\right\rangle\right|$. From Lemma 12, we have that:

$$
\begin{aligned}
g_{i,k} &\leq g_{i-1,k} + \sqrt{2\sum_{\ell\geq 0}\frac{\ell}{N}\cdot\widehat{g}^2_{i-1,k-1,\ell}} + \frac{i-1}{N}\cdot g_{i-1,k-1} \\
&\leq g_{i-1,k} + \sqrt{2}\left(\sqrt{\frac{\mu_3(i-1)}{N}}\cdot g_{i-1,k-1} + f^{col}_{i-1,\mu_3(i-1)}\right) + \frac{i-1}{N}\cdot g_{i-1,k-1} \\
&= g_{i-1,k} + \sqrt{2}\left(\sqrt{\frac{\mu_3(i-1)}{N}} + \frac{i-1}{N}\right)g_{i-1,k-1} + \sqrt{2}\cdot f^{col}_{i-1,\mu_3(i-1)}, \qquad (13)
\end{aligned}
$$

where the second inequality comes from separating the sum in two, similar to the proof of Lemma 10.

Following [11]'s proof for Lemma 14, by expanding the recursion we get:

$$g_{i,k} \leq \frac{A_i^k}{k!} + \sqrt{2} \cdot e^{A_i} 2^{9.5 N^{1/8}}. \tag{14}$$

For completeness, the proof of Equation (14) is given in Appendix B.3. Using Lemma 14, we can bound the second term, and:

$$g_{i,k} < \frac{A_i^k}{k!} + \sqrt{2} \cdot 2^{-N^{1/8}}.$$

$\square$

We can now prove the main theorem of this subsection.

Proof of Theorem 11. Following from Lemma 15, we have that:

$$g_{i,k} \leq \frac{A_i^k}{k!} + \sqrt{2} \cdot 2^{-N^{1/8}} \leq \left(\frac{A_i \cdot e}{k}\right)^k + \sqrt{2} \cdot 2^{-N^{1/8}}.$$

We now use the bound on A_i of Lemma 14:

$$g_{i,k} \leq \left(\frac{2e^{3/2}}{k} \cdot \frac{i^{7/4}}{N^{3/4}} + \frac{\sqrt{2}e}{k} \cdot \frac{i^2}{N} + \frac{e}{k} \cdot O\left(N^{-1/48}\right)\right)^k + \sqrt{2} \cdot 2^{-N^{1/8}}.$$

So if $i = o(k^{4/7} \cdot N^{3/7})$, then $g_{i,k} = o(1)$. Hence if we want $g_{i,k}$ to be a constant, i.e. not $o(1)$, we must have $i = \Omega\left(k^{4/7} \cdot N^{3/7}\right)$. $\square$

3.3 Finding k distinct s-restricted subset cover

In this section, we generalize the result to the problem of finding k distinct s–RSC, for any $s \geq 3$ and any $k \geq 1$. We are given s random functions $h_1, \ldots, h_s$ such that for any $i \in [1, s]$, $h_i : \mathcal{X} \to \mathcal{Y}$. We will prove the following theorem.

Theorem 16. *Given s random functions $h_1, \ldots, h_s : \mathcal{X} \to \mathcal{Y}$ where $N = |\mathcal{Y}|$, a quantum algorithm needs to make $\Omega\left(s^{-\frac{2^{s-1}}{2^s-1}} \cdot k^{\frac{2^{s-1}}{2^s-1}} \cdot N^{\frac{2^{s-1}-1}{2^s-1}}\right)$ queries to $h_1, \ldots, h_s$ to find k distinct s–RSC with constant probability, for any $s \geq 1$ and any $k \geq N^{1/2^{s+1}}$.*

And naturally we have the following corollary for $k = 1$:

Corollary 17. *Given s random functions $h_1, \ldots, h_s : \mathcal{X} \to \mathcal{Y}$ where $N = |\mathcal{Y}|$, a quantum algorithm needs to make $\Omega\left(s^{-\frac{2^{s-1}}{2^s-1}} \cdot N^{\frac{2^{s-1}-1}{2^s-1}}\right)$ queries to $h_1, \ldots, h_s$ to find one s–RSC with constant probability.*

In order to prove Theorem 16, we first define some notations, starting with the notation for the amplitudes. We define:

1. $f_{i,j}$ as the amplitude of the $D \in \mathcal{D}$ containing at least j distinct $(s-1)$–RSC after i quantum queries.

2. $\widehat{g}_{i,j,k}$ as the amplitude of the $D \in \mathcal{D}$ containing at least j distinct $(s-1)$–RSC and exactly k distinct s–RSC after i quantum queries.

3. $g_{i,k}$ as the amplitude of the $D \in \mathcal{D}$ containing exactly k distinct s–RSC after i quantum queries.

More formally, let $|\phi_i\rangle$ ($|\psi_i\rangle$) be the state of the algorithm just before (resp. after) the i^{th} query to the oracle. We have:

$$\begin{aligned} f_{i,j} &= \left| P^{RSC}_{j,(s-1)} \left|\psi_i\right\rangle \right|, \\ \widehat{g}_{i,j,k} &= \left| P^{RSC}_{j,(s-1)} P^{RSC}_{k,s} \neg P^{RSC}_{k+1,s} \left|\psi_i\right\rangle \right|, \\ g_{i,k} &= \left| P^{RSC}_{k,s} \neg P^{RSC}_{k+1,s} \left|\psi_i\right\rangle \right|. \end{aligned}$$

We want to bound $g_{i,k}$, and to do so, we define some convenient notation. We start by defining Π_s, a term that appears in the bound of $g_{i,k}$.

Definition 18. *Let Π_s be defined as follows:*

$$\begin{cases} \Pi_1 = 1 \\ \Pi_2 = 1 \\ \forall s \geq 2, \quad \Pi_{s+1} = 2 \cdot \sqrt{s} \cdot \sqrt{\Pi_s} \end{cases}$$

We define $A_{i,s}$ and $\mu_s(\ell)$ as follows:

Definition 19.

$$A_{i,s} = \sum_{\ell=0}^{i-1} \left(\sqrt{(s-1) \cdot \frac{\mu_s(\ell)}{N}} + \left(\frac{\ell}{N}\right)^{s/2} + \left(\sum_{r=2}^{s} \binom{s}{r} \frac{\ell}{N^r} \right)^{1/2} \right),$$

where

$$\mu_s(\ell) = \max \left\{ \Pi_{s-1} \cdot (2e)^{\frac{2^{s-2}-1}{2^{s-3}}} \frac{\ell^{(2^{s-1}-1)/2^{s-2}}}{N^{(2^{s-2}-1)/2^{s-2}}}, 10 \cdot s^2 \cdot \Pi_{s-1} \cdot N^{1/2^s} \right\}.$$

We can now state the bound on $g_{i,k}$ that we will need to prove Theorem 16:

Lemma 20. *For every $i \in \mathbb{N}$ and every $k \in \mathbb{N}$, we have that:*

$$g_{i,k} \leq \frac{A^k_{i,s+1}}{k!} + O\left(2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}} \right).$$

In order to prove Lemma 20, we first prove a bound on $A_{i,k}$.

Lemma 21. $A_{i,s} \leq (2e)^{\frac{2^{s-2}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \cdot \Pi_s + O\left(s^4 \cdot \Pi_s \cdot N^{-1/(2^s(2^s-2))}\right)$

In the interest of space, we leave the proof of Lemma 21 to Appendix B.4, and we now prove Lemma 20.

Proof of Lemma 20. We prove this theorem by induction. The case $s = 3$ corresponds to the subsection 3.2. Fix $s \geq 3$. We assume that $f_{i,j} \leq \frac{A_{i,s}^j}{j!} + O\left(2^{-s^2 \cdot \Pi_{s-1} \cdot N^{1/2^s}}\right)$ for every $i \in \mathbb{N}$ and $j \in \mathbb{N}$. We will show that $g_{i,k} \leq \frac{A_{i,s+1}^k}{k!} + O\left(2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}\right)$.

Similarly to the previous subsection, we will bound $g_{i,k}$ recursively. Using Lemma 4, we have that:

$$\left|P_{k,s}^{RSC} \left|\psi_i\right\rangle\right| \leq \left|P_{k,s}^{RSC} \left|\phi_i\right\rangle\right| + \left|P_{k,s}^{RSC} cO(I - P_{k,s}^{RSC}) \left|\phi_i\right\rangle\right|,$$

where the second term can be written as:

$$\left| P_{k,s}^{RSC} \sum_{\substack{x,y,z \\ D:(k-1) \text{ distinct } s-RSC}} \frac{1}{\sqrt{N^s}} \sum_{u'} \omega_n^{uu'} \alpha_{x,u,z,D} \left|x, u, z, D \oplus (x, u')\right\rangle \right|.$$

We analyse now the possibilities for achieving a s–RSC, considering the different cases of the inner sum. We have different possible ways to get from D that does not have a s-RSC to $D_{u'} := D \oplus (x, u')$ that has a s-RSC.

- ($x = x_0$) As for the case $s = 2$, we consider the cases where we query x such that we have $x_1, \ldots, x_s$, such that $\forall 1 \leq j \leq s$, $h_s(x) = h_s(x_s)$. For every $1 \leq j \leq s$, only $i - 1$ values of u' will make $D_{u'}$ contain a collision on h_s. Thus there are at most $\frac{(i-1)^s}{N^s}$ values of u' such that $D_{u'}$ contain a new s–RSC in this case.

- ($x = x_s$) Similarly to the case $s = 2$, we consider the case where there exists $x_0, \ldots, x_{s-1}$ such that $x_0, \ldots, x_{s-1}$ is a $(s-1)$–RSC, and we query x such that $h_j(x) = h_j(x_0)$ for some $1 \leq j \leq s$. If we have found ℓ distinct $(s-1)$–RSC in D previously, then l values of u' can make $D_{u'}$ contain a s–RSC, out of the N possible values for the outcome of h_j (notice that the values of $h_i(x)$ for $i \neq j$ are not relevant for this case), and there are s different values for j.

- However, some new terms do not appear in the case of 2–RSC. That would be the case where the query x is equal to $x_{i_1} = x_{i_2} = \cdots = x_{i_r}$ for some $r \in \{2, \ldots, s\}$ in the new s–RSC. We bound these terms as follows: for each r, there is at most $(i-1)$ distinct $(s-r)$–RSC. For each of these $(s-r)$–RSC, there are r collisions missing on some $h_{i_1}, \ldots, h_{i_r}$. And exactly one value of u' will make $D_{u'}$ contain a collision for h_{i_j}. The values of the other hash functions are irrelevant here. Hence using Lemma 3 we can bound the probability of this event by:

  $$\sum_{r=2}^{s} \frac{i-1}{N^r} {g_{i-1,k-1}}^2, \tag{15}$$

  where we bound the amplitude of the database containing at least one $(s-r)$–RSC and $k-1$ distinct s–RSC after $i-1$ quantum queries by $g_{i-1,k-1}$, the amplitude of the databases containing only $k-1$ distinct s–RSC after $i-1$ quantum queries.

Using Lemma 3, we can upper bound $g_{i,k}$ by

$$\begin{aligned} & g_{i-1,k} + \sqrt{s \sum_{\ell \geq 0} \frac{\ell}{N} \widehat{g}^2_{i-1,\ell,k-1}} + \sqrt{\frac{(i-1)^s}{N^s} {g_{i-1,k-1}}^2} + \sqrt{\sum_{r=2}^{s} \frac{i-1}{N^r} {g_{i-1,k-1}}^2} \\ & \leq g_{i-1,k} + \sqrt{s \sum_{\ell \geq 0} \frac{\ell}{N} \widehat{g}^2_{i-1,\ell,k-1}} + \left(\left(\frac{i-1}{N} \right)^{s/2} + \left(\sum_{r=2}^{s} \frac{i-1}{N^r} \right)^{1/2} \right) g_{i-1,k-1}, \end{aligned} \tag{16}$$

where the second term can be split in two, similar to the proof of Lemma 10:

$$\sqrt{s \sum_{\ell \geq 0} \frac{\ell}{N} \widehat{g}^2_{i-1,\ell,k-1}} \leq \sqrt{s \cdot \frac{\mu_{s+1}(i-1)}{N}} g_{i-1,k-1} + \sqrt{s} \cdot f_{i-1,\mu_{s+1}(i-1)}$$

The term $f_{i-1,\mu_{s+1}(i-1)}$ can be bounded by induction hypothesis by:

$$f_{i-1,\mu_{s+1}(i-1)} \leq \frac{A_{i-1,s}^{\mu_{s+1}(i-1)}}{\mu_{s+1}(i-1)!} + O\left(2^{-s^2 \cdot \Pi_{s-1} \cdot N^{1/2^s}}\right),$$

and the first term can be bounded by using Lemma 21 and the definition of $\mu_{s+1}(i-1)$ by:

$$\left(\frac{e(2e)^{\frac{2^{s-2}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \Pi_s + O\left(s^4 \Pi_s N^{-1/(2^s(2^s-2))}\right)}{\max\left\{ (2e)^{\frac{2^{s-1}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \Pi_s, 10(s+1)^2 \Pi_s \cdot N^{1/2^s} \right\}} \right)^{10(s+1)^2 \Pi_s N^{1/2^{s+1}}},$$

which is smaller than

$$\left(\frac{1}{2} + o(1) \right)^{10(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}},$$

which leads to:

$$f_{i-1,\mu_{s+1}(i-1)} < 2^{-9.8 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}.$$

We write

$$B_{\ell,s} = \sqrt{s \cdot \frac{\mu_{s+1}(\ell)}{N}} + \left(\frac{\ell-1}{N} \right)^{s/2} + \left(\sum_{r=2}^{s} \frac{\ell}{N^r} \right)^{1/2},$$

and rewrite Equation (16) as:

$$g_{i,k} \leq g_{i-1,k} + B_{\ell,s} \cdot g_{i-1,k-1} + \sqrt{s} \cdot 2^{-9.8 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}.$$

Then, by expanding the inequality and using the fact that $g_{0,k-1} = 0$, we get:

$$\begin{aligned} g_{i,k} \le & g_{i-1,k} + B_{\ell,s} \cdot g_{i-1,k-1} + \sqrt{s} \cdot 2^{-9.8 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}} \\ & \vdots \\ \le & \sum_{\ell=0}^{i-1} \left(B_{\ell,s} \cdot g_{\ell,k-1} + \sqrt{s} \cdot 2^{-9.8 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}} \right) \\ \le & \left(\sum_{\ell=0}^{i-1} B_{\ell,s} \cdot g_{\ell,k-1} \right) + s \cdot N^{1/2} \cdot \sqrt{s} \cdot 2^{-9.8 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}} \\ \le & \left(\sum_{\ell=0}^{i-1} B_{\ell,s} \cdot g_{\ell,k-1} \right) + s^{3/2} \cdot 2^{-9.5 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}, \end{aligned}$$

where we use the fact that $i \le s \cdot \sqrt{N}$ for the third inequality.

Expanding this inequality, we obtain

$$g_{i,k} \le \frac{A_{i,s+1}^k}{k!} + s^{3/2} \cdot e^{A_{i,s+1}} \cdot 2^{-9.5 \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}. \tag{17}$$

For details on Equation (17), see Appendix B.5.

And because $i \le s \cdot \sqrt{N}$, we have $A_{i,s+1} \le 2e \cdot (s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}$. Using this and the fact that $s^{3/2} \le 2^{\Pi_s \cdot (s+1)^2 \cdot N^{1/2^{s+1}}}$, we conclude:

$$g_{i,k} \le \frac{A_{i,s+1}^k}{k!} + 2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}.$$

□

At last we bound Π_s to conclude the analysis.

Proposition 22. *We have for any $s \in \mathbb{N}$ that:*

$$\Pi_s \le 4s$$

Proof. The statement is true for $s = 1, 2$. Assume it is true for $s \ge 2$. Then,

$$\Pi_{s+1} = 2\sqrt{s} \cdot \sqrt{\Pi_s} \le 2\sqrt{s} \cdot \sqrt{4(s-1)} \le 4s.$$

□

Finally, we can prove Theorem 16:

Proof of Theorem 16. From Lemma 21, we have:

$$A_{i,s} \le (2e)^{\frac{2^{s-2}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \cdot \Pi_s + O\left(s^4 \cdot \Pi_s \cdot N^{-1/(2^s(2^s-2))}\right).$$

Hence we can bound $g_{i,k}$ for any i, k, by:

$$\begin{aligned}
g_{i,k} &\leq \frac{A_{i,s+1}^k}{k!} + O\left(2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}\right) \\
&\leq \left(\frac{e \cdot A_{i,s+1}}{k}\right)^k + O\left(2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}\right) \\
&\leq \left(\frac{e}{k}(2e)^{\frac{2^{s-2}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \cdot \Pi_s + \frac{e}{k} \cdot O\left(s^4 \Pi_s \cdot N^{-1/(2^s(2^s-2))}\right)\right)^k \\
&\quad + O\left(2^{-(s+1)^2 \cdot \Pi_s \cdot N^{1/2^{s+1}}}\right) \\
&\leq \left(\frac{e}{k} \cdot (2e)^{\frac{2^{s-2}-1}{2^{s-2}}} \frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}} \cdot 4s + \frac{e}{k} \cdot O\left(4s^5 \cdot N^{-1/(2^s(2^s-2))}\right)\right)^k \\
&\quad + O\left(2^{-4s(s+1)^2 \cdot N^{1/2^{s+1}}}\right),
\end{aligned}$$

where the first inequality comes from Lemma 20, the third inequality comes from Lemma 21 and the last inequality comes from Proposition 22.

If $i = o\left(s^{-\frac{2^{s-1}}{2^s-1}} \cdot k^{\frac{2^{s-1}}{2^s-1}} \cdot N^{\frac{2^{s-1}-1}{2^s-1}}\right)$, then $g_{i,k} = o(1)$. Hence if we want $g_{i,k}$ to be constant, i.e. not $o(1)$, we must have $i = \Omega\left(s^{-\frac{2^{s-1}}{2^s-1}} \cdot k^{\frac{2^{s-1}}{2^s-1}} \cdot N^{\frac{2^{s-1}-1}{2^s-1}}\right)$. □

4 The (r, k)–subset cover problem

In this section, we prove some upper and lower bounds on the (r, k)–SC problem. As far as we know, there are no quantum algorithm to find a (r, k)–SC problem, except for [14]'s algorithm when $k = r$, and for the harder problem of finding a k–RSC. We first prove a lower bound on the $(1, k)$–SC problem, then design new algorithms for finding a (r, k)–SC.

4.1 Lower bound on finding a $(1, k)$–subset cover

In this subsection, we will prove a lower bound on the $(1, k)$–SC problem. We are given k random functions $h_1, \ldots, h_k$ such that for $i \in [1, k]$, $h_i : \mathcal{X} \to \mathcal{Y}$. We write $N = |\mathcal{Y}|$ and for $x \in \mathcal{X}$, we write $H(x) = \{h(x_i) | i \in [1, k]\}$. The goal of this subsection is to prove the following theorem.

Theorem 23. *Given k random functions $h_1, \ldots, h_k : \mathcal{X} \to \mathcal{Y}$ where $N = |\mathcal{Y}|$, a quantum algorithm needs to make $\Omega\left(C_k^{-1/5} \cdot N^{k/5}\right)$ queries to $h_1, \ldots, h_k$ to find one (1,k)–SC with constant probability, where $C_k = \sum_{j=2}^{k} \frac{k!}{(j-1)!}$.*

To prove Theorem 23, we introduce the problem of finding a *j–repetition* on $h_{i_1}, \ldots, h_{i_j}$, that consist of finding an $x \in \mathcal{X}$ such that $h_{i_1}(x) = \cdots = h_{i_j}(x)$. More formally, we define the following database property:

Definition 24.

$$\forall \ell, j, P_{\ell,j}^{rep} = \left\{ D \in \mathcal{D} \middle| \begin{array}{l} \exists x_1, x_2, \ldots, x_\ell, \forall i, \forall 1 \leq \ell \leq j, h_1(x_i) = h_\ell(x_i) \\ \forall i \neq p, x_i \neq x_p \end{array} \right\}$$

Note that we define the property only for ℓ distinct j–repetition on $h_1, \dots, h_j$, because by symmetry, the probability of finding a j–repetition on $h_1, \dots, h_j$ is the same as finding a j–repetition on $h_{i_1}, \dots, h_{i_\ell}$.

We also define:

1. $\widetilde{f}^{rep}_{i,\ell,j}$ as the amplitude of the D containing *at least* ℓ distinct j–repetitions on $h_1, \dots, h_j$ after i quantum queries.

2. $f^{rep}_{i,\ell,j}$ as the amplitude of the D containing *exactly* ℓ distinct j–repetitions on $h_1, \dots, h_j$ after i quantum queries.

3. $g_{i,k}$ as the amplitude of the D containing at least one $(1,k)$–SC after i quantum queries.

More formally, let $|\psi_i\rangle$ be the state just after the i^{th} query to the oracle, then $\widetilde{f}^{rep}_{i,\ell,j} = \left| P^{rep}_{\ell,j} |\psi_i\rangle \right|$, $f^{rep}_{i,\ell,j} = \left| P^{rep}_{\ell,j} \neg P^{rep}_{\ell+1,j} |\psi_i\rangle \right|$, and $g_{i,k} = \left| P^{SC}_{(1,k)} |\psi_i\rangle \right|$.

Our goal is to bound $g_{i,k}$ and for that we will bound $\widetilde{f}^{rep}_{i,\ell,j}$.

Lemma 25. *For all $i, \ell, j \in \mathbb{N}$, we have that:*

$$\widetilde{f}^{rep}_{i,\ell,j} \leq \left(\frac{e \cdot i}{\ell \cdot N^{\frac{j-1}{2}}} \right)^{\ell}.$$

Proof. Following the proof of Lemma 12, we have that:

$$\begin{aligned}
\widetilde{f}^{rep}_{i,\ell,j} &\leq \widetilde{f}^{rep}_{i-1,\ell,j} + \sqrt{\frac{1}{N^{j-1}} \widetilde{f}^{rep}_{i-1,\ell-1,k}}^{\,2} \\
&\leq \sum_{m=0}^{i-1} \sqrt{\frac{1}{N^{j-1}} \widetilde{f}^{rep}_{m,\ell-1,k}} \\
&\leq \sum_{m_1=0}^{i-1} \sum_{m_2=0}^{m_1} \sqrt{\frac{1}{N^{j-1}}} \sqrt{\frac{1}{N^{j-1}} \widetilde{f}^{rep}_{m_2,\ell-2,k}} \\
&\vdots \\
&\leq \sum_{0 \leq m_\ell < m_{\ell-1} < \cdots < m_1 < i} \left(\frac{1}{N^{j-1}} \right)^{\ell/2} \\
&\leq \frac{i^\ell}{\ell!} \left(\frac{1}{N^{j-1}} \right)^{\ell/2} \\
&\leq \left(\frac{e \cdot i}{\ell \cdot N^{(j-1)/2}} \right)^{\ell}.
\end{aligned}$$

□

We now bound the amplitude $g_{i,k}$ with an inductive formula, as for the RSC problem.

Lemma 26. *For all $i \in \mathbb{N}$ and $k \in \mathbb{N}$, we have that:*

$$g_{i,k} \leq g_{i-1,k} + \left(k^k \frac{i-1}{N^k} \right)^{1/2} + \left(\sum_{j=2}^{k} \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} \cdot \frac{k!}{(j-1)!} {f^{rep}_{i-1,\ell,j}}^2 \right)^{1/2}.$$

Proof. For convenience, we denote $P_k = P^{SC}_{(1,k)}$ the projector on the databases D that contain at least a $(1,k)$–SC. We write $|\phi_i\rangle$ the state just before the i^{th} quantum queries, and $|\psi_i\rangle$ the state just after the i^{th} quantum query.

Using Lemma 4, and writing $D_{u'} := D \oplus (x, u')$ we have that:

$$|P_k\,|\psi_i\rangle| \leq |P_k\,|\phi_i\rangle| + \left| P_k \sum_{\substack{x,y,z \\ D\text{:no (1,k)–SC}}} \frac{1}{\sqrt{N^k}} \sum_{u'} \omega_n^{uu'} \alpha_{x,u,z,D}\,|x,u,z,D_{u'}\rangle \right| .$$

We analyse now the possibilities for achieving a $(1,k)$–SC, considering the different cases of the inner sum. We have multiple possible ways to get from D that does not have a $(1,k)$–SC to $D_{u'}$ that has a $(1,k)$–SC.

- $(x = x_0)$ Here, we consider the case we query x such that $\{h_i(x)\} \subseteq \{h_i(x_1)\}$, where x_1 was queried before. Notice that there are $(i-1)$ possible values of x_1, and for each fixed value of x_1, we have k^k possible values of $H(x)$ that would lead to this value. This leads to $k^k(i-1)$ possible values of u' that would lead to an $(1,k)$-SC.

- $(x = x_1)$ Here, we consider the case we query x such that $\{h_i(x_0)\} \subseteq \{h_i(x)\}$, where x_0 was queried before.

  Let us suppose that x_0 has a j-repetition on $h_{i_1}, \ldots, h_{i_j}$, for some distinct $i_1, ..., i_j$. Notice that in this case, $S := \{h_i(x_0)\}$ has $k - j + 1$ elements and we will count the number of possible $H(x)$ that contains all of these elements. Out of the k functions $h_1, \ldots, h_k$, we have $\binom{k}{k-j+1}$ possible ways of choosing the functions that will be filled with the values in S. When we fix such functions, there are $|S|! = (k-j+1)!$ ways of filling them with the elements of S, and N^{j-1} ways of filling the other functions. Therefore, there are $\binom{k}{k-j+1}(k-j+1)!N^{j-1}$ values of $H(x)$ such that $\{h_i(x_0)\} \subseteq \{h_i(x)\}$.

This gives:

$$|P_k\,|\psi_i\rangle| \leq |P_k\,|\phi_i\rangle| + \left(k^k \frac{i-1}{N^k} \sum_{\substack{x,y,z \\ D\text{:no k–SC}}} |\alpha_{x,y,z,D}|^2 \right)^{1/2}$$
$$+ \left(\sum_{j=2}^{k} \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} \cdot \frac{k!}{(j-1)!} \sum_{\substack{x,y,z \\ D\text{:no k–SC} \\ \ell \text{ distinct } j-repetitions}} |\alpha_{x,y,z,D}|^2 \right)^{1/2} .$$

Using Lemma 3 and our notation, we conclude:

$$g_{i,k} \leq g_{i-1,k} + \left(k^k \frac{i-1}{N^k} \right)^{1/2} + \left(\sum_{j=2}^{k} \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} \cdot \frac{k!}{(j-1)!} {f^{rep}_{i-1,\ell,j}}^2 \right)^{1/2} .$$

□

We now bound $g_{i,k}$ in the following lemma.

Lemma 27. *For every $i \in \mathbb{N}$ and $k \in \mathbb{N}$, we have that:*

$$g_{i,k} \leq k^{k/2} \cdot \frac{i^{3/2}}{N^{k/2}} + \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot i^{5/2}}{N^{k/2}}.$$

Proof. From Lemma 26, we have that:

$$g_{i,k} \leq g_{i-1,k} + \left(k^k \frac{i-1}{N^k}\right)^{1/2} + \left(\sum_{j=2}^{k} \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} \cdot \frac{k!}{(j-1)!} {f_{i-1,\ell,j}^{rep}}^2\right)^{1/2}.$$

We want to bound each term in the sum indexed by j. Fix $j \in \{2, \dots, k\}$. We have that:

$$\sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} \cdot \frac{k!}{(j-1)!} {f_{i-1,\ell,j}^{rep}}^2 = \frac{k!}{(j-1)!} \cdot \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} {f_{i-1,\ell,j}^{rep}}^2.$$

Next, we have that:

$$\begin{aligned}
\sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} {f_{i-1,\ell,j}^{rep}}^2 &\leq \frac{i-1}{N^{k+1-j}} \cdot \sum_{\ell \geq 1} {f_{i-1,\ell,j}^{rep}}^2 \\
&= \frac{i-1}{N^{k+1-j}} \cdot {\widetilde{f}_{i-1,1,j}^{rep}}^2 \\
&\leq \frac{i-1}{N^{k+1-j}} \cdot \left(\frac{e \cdot (i-1)}{N^{\frac{j-1}{2}}}\right)^2 \\
&= \frac{e^2 (i-1)^3}{N^k},
\end{aligned}$$

where $\widetilde{f}_{i-1,1,j}^{rep}$ is the amplitude of the D containing *at least* one j–repetition on $h_1, \dots, h_j$ after $i-1$ quantum queries. The first inequality follows since there cannot be more than $i-1$ distinct j–repetitions on $h_1, \dots, h_j$ after $i-1$ quantum queries. The second inequality comes from the bound on $\widetilde{f}_{i-1,1,j}^{rep}$ in Lemma 25.

This gives:

$$\left(\sum_{j=2}^{k} \frac{k!}{(j-1)!} \sum_{\ell \geq 0} \frac{\ell}{N^{k+1-j}} {f_{i-1,\ell,j}^{rep}}^2\right)^{1/2} \leq \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot (i-1)^{3/2}}{N^{k/2}}.$$

Finally, by developing the recursive terms (using that $g_{0,k} = 0$), we get that:

$$\begin{aligned}
g_{i,k} &\leq g_{i-1,k} + \sqrt{k^k \frac{i-1}{N^k}} + \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot (i-1)^{3/2}}{N^{k/2}} \\
&\vdots \\
&\leq \sum_{\ell=0}^{i-1} \left(\sqrt{k^k \frac{\ell}{N^k}} + \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot \ell^{3/2}}{N^{k/2}}\right) \\
&\leq k^{k/2} \frac{i^{3/2}}{N^{k/2}} + \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot i^{5/2}}{N^{k/2}}.
\end{aligned}$$

□

We can now prove Theorem 23.

Proof of Theorem 23. From Lemma 27, we have that:

$$g_{i,k} \leq k^{k/2} \cdot \frac{i^{3/2}}{N^{k/2}} + \sqrt{\sum_{j=2}^{k} \frac{k!}{(j-1)!}} \cdot \frac{e \cdot i^{5/2}}{N^{k/2}}.$$

Writing $C_k = \sum_{j=2}^{k} \frac{k!}{(j-1)!}$, this rewrites as:

$$g_{i,k} \leq k^{k/2} \cdot \frac{i^{3/2}}{N^{k/2}} + \sqrt{C_k} \cdot \frac{e \cdot i^{5/2}}{N^{k/2}}.$$

If $i = o\left(C_k^{-1/5} \cdot N^{k/5}\right)$, then $g_{i,k} = o(1)$. Hence if we want $g_{i,k}$ to be constant, i.e. not $o(1)$, we must have $i = \Omega\left(C_k^{-1/5} \cdot N^{k/5}\right)$. □

4.2 Algorithm for finding a $(1,k)$–subset cover

We now describe an algorithm that finds a $(1,k)$–SC, assuming $|\mathcal{X}| = |\mathcal{Y}|^k = N^k$.

We first notice that an algorithm that finds a collision on H also finds a $(1,k)$–SC in an expected $O(N^{k/3})$ number of queries. We show now that there is a more efficient algorithm, as stated in the following theorem:

Theorem 28. *There exists a quantum algorithm that finds a $(1,k)$–SC in expected $O\left(N^{k/4}\right)$ quantum queries if k is even, and $O(N^{k/4+1/12})$ if k is odd.*

To prove this theorem, we describe the following algorithm (which takes as parameters j and t, whose values will be chosen later):

Algorithm 1. *Input: $j \in \{2,\dots,k\}$ and $t \in \mathbb{N}$.*

1. *Define $F_1 : \mathcal{X} \to \{0,1\}$ as follows:*

$$F_1(x) = \begin{cases} 1, & \textit{if } h_1(x) = h_2(x) = \cdots = h_j(x) \\ 0, & \textit{otherwise.} \end{cases}$$

   *(Note that an element $x \in \mathcal{X}$ such that $F_1(x) = 1$ is a j–*repetition.*)*

2. *Execute Grover's algorithm t times on F_1 to find t distinct j–repetitions in H. Let $T = \{x_1,\dots,x_t\}$ be the set of these j–repetitions.*

3. *Define $F_2 : \mathcal{X} \to \{0,1\}$ as follows:*

$$F_2(x) = \begin{cases} 1, & \textit{if there exists } x_0 \in T \textit{ such that } h_1(x) = h_1(x_0) \\ & \textit{and for } 1 \leq m \leq k-j, h_{m+1}(x) = h_{j+m}(x_0) \\ 0, & \textit{otherwise.} \end{cases}$$

4. *Execute Grover's algorithm to find an x such that $F_2(x) = 1$*

5. *Find x_0 in T corresponding to x, and output (x, x_0).*

Lemma 29. *Algorithm 1 makes an expected number of* $O\left(N^{(2k-j+1)/6}\right)$ *queries to the oracle when* $j \leq \frac{k+2}{2}$ *for* $t = N^{(k-2j+2)/3}$.

Proof. Notice that if we consider a uniformly random function, we have that $Pr[h_1(x) = \cdots = h_j(x)] = N^{-j+1}$. Therefore, the expected number of elements in $\mathcal{X}$ such that $F_1(x) = 1$ is $N^k \cdot N^{-j+1} = N^{k-j+1}$. We write $X_1, \ldots, X_{N^k}$ the random variables corresponding to F_1's output on each $x \in \mathcal{X}$, X the sum of this variables, $\mu = N^{k-j+1}$ its mean. Chernoff bound tells us that for any $0 \leq \delta \leq 1$,

$$Pr\left(|X - \mu| \geq \mu\delta\right) \leq e^{-\delta^2\mu/3}.$$

With $\delta = 1/2$, we have:

$$Pr\left(|X - \mu| \geq \frac{\mu}{2}\right) \leq e^{-\mu/12}.$$

Thus, unless with probability $e^{-(N^{k-j+1})/12}$, the number of elements $x \in \mathcal{X}$ such that $F_1(x) = 1$ is greater than $N^{k-j+1}/2$.

Hence using Theorem 35, the second step of the algorithm is expected to make $O\left(t \cdot \sqrt{\frac{N^k}{N^{k-j+1}}}\right) = O\left(t \cdot N^{(j-1)/2}\right)$ quantum queries to the oracle.

Notice that for a fixed value x_0, if we consider a uniformly random function, we have that

$$Pr[h_1(x) = h_1(x_0) \wedge h_2(x) = h_{m+1}(x_0) \wedge \cdots \wedge h_{k-j+1}(x) = h_k(x)] = N^{j-k-1}.$$

Therefore, the expected number of elements such that $F_2(x) = 1$ is $t \cdot N^k \cdot N^{j-k-1} = t \cdot N^{j-1}$. Similarly, using Chernoff bound, unless with probability $e^{-(t \cdot N^{j-1})/12}$, the number of elements such that $F_2(x) = 1$ is greater than $t \cdot N^{j-1}/2$. Hence, using Theorem 35, the fourth step of the algorithm is expected to make $O\left(\sqrt{\frac{N^k}{t \cdot N^{j-1}}}\right) = O\left(\frac{N^{(k-j+1)/2}}{\sqrt{t}}\right)$ quantum queries to the oracle.

By picking $t = N^{(k-2j+2)/2}$ with $j \leq \frac{k+2}{2}$, the complexity of the algorithm is $O(N^{(k-2j+2)/3} \cdot N^{(j-1)/2}) = O(N^{(2k-j+1)/6})$. □

We now prove Theorem 23

Proof of Theorem 23. From Lemma 29, the complexity of Algorithm 1 is$O(N^{(2k-j+1)/6})$ when $j \leq \frac{k+2}{2}$.

- If k is even, then we pick $j = \frac{k+2}{2}$ to reach a complexity of $O(N^{k/4})$.
- If k is odd, then we pick $j = \frac{k+1}{2}$ to reach a complexity of $O(N^{k/4+1/12})$.

Note that if $j > \frac{k+1}{2}$, then the second step of the algorithm is expected to make at least $O\left(N^{\frac{k+1}{4}}\right)$ quantum queries, which is worse than $O(N^{k/4+1/12})$. □

Remark 1. *Note that we do not reach the lower bound of Theorem 23, and it would be interesting to see if the gap can be further reduced by either improving our lower bounds or designing a more efficient algorithm.*

A slightly better algorithm We describe a more efficient algorithm when k is not constant. The idea is to take into account the fact that we do not necessarily need the j–repetitions from the previous algorithm to occur on the first j functions $h_1, \dots, h_j$, but they could rather be on any $h_{i_1}, \dots, h_{i_j}$ instead. We also consider permutations of the $h_1, \dots, h_k$ in the fourth step of Algorithm 1.

Theorem 30. *There exists a quantum n algorithm that finds a $(1, k)$–SC in:*

- $O\left(\binom{k}{(k+2)/2}^{-1/2} \cdot N^{k/4}\right)$ *quantum queries if k is even,*
- $O\left(\binom{k}{(k+1)/2}^{-1/2} \cdot N^{k/4+1/12}\right)$ *quantum queries if k is odd.*

The gain that we obtain is a function of k and is therefore not significant if k is constant. However, as we have shown in Theorem 23, the dependence in k can be quite large for the $(1, k)$–SC problem.

To prove this theorem, we describe the algorithm as follows (which takes again as input two integers j and t playing the role of parameters whose optimal values will be determined later):

Algorithm 2. *Input: $j \in \{2, \dots, k\}$ and $t \in \mathbb{N}$.*

1. *Define $F_1 : \mathcal{X} \to \{0, 1\}$ as follows:*

$$F_1(x) = \begin{cases} 1, & \textit{if there exists distinct } i_1, \dots, i_j \in [1, k] \textit{ such that} \\ & h_{i_1}(x) = h_{i_2}(x) = \dots = h_{i_j}(x) \\ 0, & \textit{otherwise.} \end{cases}$$

*(Note that an element $x \in \mathcal{X}$ such that $F_1(x) = 1$ is a j–*repetition.*)*

2. *Execute Grover's algorithm t times on F_1 to find t distinct j–repetitions in H. Let $T = \{x_1, \dots, x_t\}$ be the set of these j–repetitions. We write, for $\ell \in [1, t]$ $I_\ell = \{i_1^\ell, \dots, i_j^\ell\}$ the set of indices such that $h_{i_1^\ell}(x_t) = \dots = h_{i_k^\ell}(x_t)$, and $I'_\ell = [1, k] \backslash I_\ell = \{i_{j+1}^\ell, \dots, i_k^\ell\}$.*

3. *Define $F_2 : \mathcal{X} \to \{0, 1\}$ as follows:*

$$F_2(x) = \begin{cases} 1, & \textit{if there exists distinct } j_0, j_1, \dots, j_{k-j+1} \in [1, k], \\ & \quad \textit{and } \ell \in [1, t] \textit{ s.t. } h_{i_1^\ell}(x_\ell) = h_{j_0}(x) \\ & \quad \textit{and for all } 1 \le m \le k - j, h_{j_m}(x) = h_{i_{j+m}^\ell}(x_\ell) \\ 0, & \textit{otherwise.} \end{cases}$$

4. *Execute Grover's algorithm find an x such that $F_2(x) = 1$*

5. *Find x_0 in T, and output (x, x_0).*

Remark 2. *F_1 (F_2) can be constructed with $O\left(\binom{k}{j}\right)$ (resp. $O\left(\frac{k!}{(j-1)!}\right)$) quantum gates and one query to H.*

Lemma 31. *Algorithm 2 makes an expected number of $O\left(\binom{k}{j}^{-1/2} N^{(2k-j+1)/6}\right)$ queries to the oracle when $j \le \frac{k+2}{2}$ for $t = N^{(k-2j+2)/3}$.*

The proof of Lemma 31 is given in Appendix B.6. We prove now Theorem 30.

Proof of Theorem 30. From Lemma 31, the complexity of Algorithm 2 is $O\left(\binom{k}{j}^{-1/2} \cdot N^{(2k-j+1)/6}\right)$ when $j \leq \frac{k+2}{2}$.

- If k is even, for $j = \frac{k+2}{2}$, we get a complexity of $O\left(\binom{k}{(k+2)/2}^{-1/2} \cdot N^{k/4}\right)$.
- If k is odd, for $j = \frac{k+1}{2}$, we get a complexity of $O\left(\binom{k}{(k+1)/2}^{-1/2} \cdot N^{k/4+1/12}\right)$.

Note that if $j > \frac{k+1}{2}$, then the second step of the algorithm is expected to make at least $O\left(\binom{k}{(k+1)/2}^{-1/2} \cdot N^{\frac{k+1}{4}}\right)$ quantum queries. □

4.3 Algorithm for finding a (r, k)–subset cover

In this section, we describe an algorithm for solving the (r, k)–SC problem. We consider the case where $|\mathcal{X}| = |r \cdot \mathcal{Y}|^k = r^k \cdot N^k$. The result is stated as follows:

Theorem 32. *There exists a quantum algorithm that finds a (r, k)–SC in $O\left(N^{k/(2+2r)}\right)$ quantum queries to H, if k is divisible by $r + 1$, and $O\left(N^{k/(2+2r)+1/2}\right)$ otherwise.*

The idea of the algorithm is basically the same as Algorithm 1 of Section 4.2:

1. we first find t distinct $(r - 1, k')$–SC for some integers t and k';
2. we then find the (r, k)–SC problem.

The first step is done recursively, using the algorithm defined for lower values of k' and $r - 1$. The second step uses Grover's algorithm. The algorithm can be defined for any value of k' and t, and we pick them to optimize the complexity.

More formally, we define the algorithm recursively. Assume that we have an algorithm that can output a $(r - 1, k')$–SC in $O\left(N^{k'/2r}\right)$ queries, for any $k' < k$ such that k' is divisible by r. Then, we can find a (r, k)–SC as follows:

Algorithm 3. *Input: $t \in \mathbb{N}$, $k' \in \mathbb{N}$.*

1. *Execute the $(r - 1, k')$–SC algorithm t times to find t distinct $(r - 1, k')$–SC in H. Let $T = \{(x_{1,0}, x_{1,1}, \ldots, x_{1,r-1}), \ldots, (x_{t,0}, x_{t,1}, \ldots, x_{t,r-1})\}$ be the set of these $(r - 1, k')$–SC.*
2. *Define $F : \mathcal{X} \to \{0, 1\}$ as follows:*

$$F(x) = \begin{cases} 1, & \text{if there exists } (x_{i,0}, x_{i,1}, \ldots, x_{i,r-1}) \in T \text{ such that} \\ & \forall 1 \leq m \leq k - k', h_m(x) = h_{k'+m}(x_{i,0}), \\ 0, & \text{otherwise.} \end{cases}$$

3. *Execute Grover's algorithm to find an x such that $F(x) = 1$*
4. *Find $(x_{i,0}, x_{i,1}, \ldots, x_{i,r-1})$ in T and output $(x_{i,0}, x_{i,1}, \ldots, x_{i,r-1}, x)$.*

Lemma 33. *Algorithm 3 makes an expected number of $O\left(N^{k/(2+2r)}\right)$ queries to the oracle, when k is divisible by r, and $O\left(N^{k/(2+2r)+1/2}\right)$ otherwise.*

We defer the proof of Lemma 33 to Appendix B.7.

References

[1] J.-P. Aumasson and G. Endignoux. Clarifying the subset-resilience problem. Cryptology ePrint Archive, Report 2017/909, 2017. `https://eprint.iacr.org/2017/909`.

[2] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn. SPHINCS: Practical stateless hash-based signatures. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 368–397. Springer, Heidelberg, Apr. 2015.

[3] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe. The SPHINCS$^+$ signature framework. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 2129–2146. ACM Press, Nov. 2019.

[4] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, Dec. 2011.

[5] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, jun 1998.

[6] G. Brassard, P. Høyer, and A. Tapp. Quantum cryptanalysis of hash and claw-free functions. In C. L. Lucchesi and A. V. Moura, editors, *LATIN '98: Theoretical Informatics, Third Latin American Symposium, Campinas, Brazil, April, 20-24, 1998, Proceedings*, volume 1380 of *Lecture Notes in Computer Science*, pages 163–169. Springer, 1998.

[7] K.-M. Chung, S. Fehr, Y.-H. Huang, and T.-N. Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In A. Canteaut and F.-X. Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 598–629. Springer, Heidelberg, Oct. 2021.

[8] L. K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996.

[9] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, Jul 1997.

[10] L. Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, october 1979.

[11] Q. Liu and M. Zhandry. On finding quantum multi-collisions. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 189–218. Springer, Heidelberg, May 2019.

[12] L. Reyzin and N. Reyzin. Better than BiBa: Short one-time signatures with fast signing and verifying. In L. M. Batten and J. Seberry, editors, *ACISP 02*, volume 2384 of *LNCS*, pages 144–153. Springer, Heidelberg, July 2002.

[13] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, oct 1997.

[14] Q. Yuan, M. Tibouchi, and M. Abe. On subset-resilient hash function families. *Designs, Codes and Cryptography*, 90, 03 2022.

[15] M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, Aug. 2019.

A Grover's algorithm and Quantum Fourier Transform

A.1 Grover's algorithm

Here we quickly recall Grover's algorithm. We start by defining the search problem.

Definition 34 (Search problem)**.** *We are given a function $F : \mathcal{X} \to \{0,1\}$. The search problem consists of finding an $x \in \mathcal{X}$ such that $F(x) = 1$, in the least amount of queries to F possible.*

Grover's algorithm solves the search problem in $O\left(\sqrt{\frac{|\mathcal{X}|}{t}}\right)$, where t is the number of x such that $F(x) = 1$. The result is stated as follows:

Theorem 35 ([9][5])**.** *Let $F : \mathcal{X} \to \{0,1\}$ be a function, $t = |\{x|F(x) = 1\}|$, and $N = |\mathcal{X}|$. Then, Grover's algorithm finds an x such that $F(x) = 1$ with constant probability with $O\left(\sqrt{\frac{N}{t}}\right)$ queries to F. Moreover, this algorithm is optimal.*

Remark 3. *When constructing quantum algorithms in the* Quantum Random Oracle Model*, we are given a black box access to a function $H : \mathcal{X} \to \mathcal{Y}$. To use Grover's algorithm in this model, we need to construct the function $F : \mathcal{X} \to \{0,1\}$ from the function H. Then, to count the number of queries to H, it is sufficient to compute the number of queries to F.*

A.2 The Quantum Fourier Transform

Let $Y = \{0,1\}^n$, for some $n \in \mathbb{N}$. We recall that the computational basis is $\{|i\rangle\}_{i \in Y}$. The **Quantum Fourier Transform** is a unitary that, given an input state $|\phi\rangle = \sum_{k=0}^{2^n-1} x_k |k\rangle$, outputs $\sum_{k=0}^{2^n} y_k |k\rangle$ where the y_k's are computed with the following formula:

$$y_k = \frac{1}{2^{n/2}} \sum_{\ell=0}^{2^n-1} x_\ell \omega_n^{k\ell}$$

where $\omega_n = e^{2\pi i/2^n}$ thus ω_n^ℓ is a 2^n-th root of unity.
This unitary can be efficiently implemented, and we write it *QFT*.
Applying the *QFT* to the computational basis yields the *Fourier basis.*

B Technical proofs

B.1 Proof of Equation (6)

Writing $D_{u'} = D \oplus (x, u')$,

$$
\begin{aligned}
&\left| P_2 c O (I - P_2) \left|\phi_i\right\rangle \right| \\
\leq &\Bigg| \sum_{\ell \geq 0} \frac{\ell}{N} \sum_{b \in \{1,2\}} \sum_{\substack{x,y,z \\ D:\neg P_2 \\ \text{exactly } \ell \\ \text{collisions on } h_b}} \sum_{u'} \frac{1}{\sqrt{N^2}} \omega_n^{uu'} \alpha_{x,u,z,D} \left|x, u, z, D_{u'}\right\rangle \\
&+ \frac{(i-1)^2}{N^2} \sum_{\substack{x,y,z \\ D:\neg P_2}} \sum_{u'} \frac{1}{\sqrt{N^2}} \omega_n^{uu'} \alpha_{x,u,z,D} \left|x, y, z, D_{u'}\right\rangle \Bigg| \\
\leq &\left| 2 \cdot \sum_{\ell \geq 0} \frac{\ell}{N} \sum_{\substack{x,y,z \\ D:\neg P_2 \\ \text{exactly } \ell \\ \text{collisions on } h_1}} \sum_{u'} \frac{1}{\sqrt{N^2}} \omega_n^{uu'} \alpha_{x,u,z,D} \left|x, u, z, D_{u'}\right\rangle \right| \\
&+ \left| \frac{(i-1)^2}{N^2} \sum_{\substack{x,y,z \\ D:\neg P_2}} \sum_{u'} \frac{1}{\sqrt{N^2}} \omega_n^{uu'} \alpha_{x,u,z,D} \left|x, y, z, D_{u'}\right\rangle \right| \\
\leq &\left(2 \cdot \sum_{\ell \geq 0} \frac{\ell}{N} \sum_{\substack{x,y,z \\ D:\neg P_2 \\ \text{exactly } \ell \\ \text{collisions on } h_1}} \left|\alpha_{x,u,z,D}\right|^2 \right)^{1/2} + \left(\frac{(i-1)^2}{N^2} \sum_{\substack{x,y,z \\ D:\neg P_2}} \left|\alpha_{x,u,z,D}\right|^2 \right)^{1/2} \\
\leq &\left(2 \cdot \sum_{\ell \geq 0} \frac{\ell}{N} \left| P_{\ell-col-h_1} \left|\phi_i\right\rangle \right|^2 \right)^{1/2} + \frac{(i-1)}{N},
\end{aligned}
$$

where in the second inequality, we used the symmetry of finding collisions on h_1 and collisions on h_2, and used the definition of $\left| P_{\ell-col-h_1} \left|\phi_i\right\rangle \right|^2$ in the last inequality.

B.2 Proof of Lemma 14

Proof. We have that

$$\begin{aligned}
A_i &\leq \sum_{\ell:\mu(\ell)=2e\frac{\ell^{3/2}}{\sqrt{N}}} \sqrt{2}\cdot\frac{\sqrt{2e\ell^{3/2}}}{N^{3/4}} + \sum_{\ell:\mu_3(\ell)=10N^{1/8}} \sqrt{2}\cdot\frac{\sqrt{10N^{1/8}}}{N^{1/2}} + \sum_{\ell=0}^{i-1}\sqrt{2}\cdot\frac{\ell-1}{N} \\
&\leq \sum_{\ell=1}^{i-1}\sqrt{2}\cdot\frac{\sqrt{2e\ell^{3/2}}}{N^{3/4}} + \sum_{\ell:\mu_3(\ell)=10N^{1/8}} \sqrt{2}\cdot\frac{\sqrt{10N^{1/8}}}{N^{1/2}} + \sum_{\ell=0}^{i-1}\sqrt{2}\cdot\frac{\ell-1}{N} \\
&\leq 2\sqrt{e}\frac{i^{7/4}}{N^{3/4}} + \sqrt{2}\cdot\left(\frac{10}{2e}\right)^{2/3}\cdot N^{5/12}\cdot\frac{\sqrt{10N^{1/8}}}{N^{1/2}} + \sqrt{2}\cdot\frac{i^2}{N} \\
&\leq 2\sqrt{e}\cdot\frac{i^{7/4}}{N^{3/4}} + \sqrt{2}\cdot\frac{i^2}{N} + O\left(N^{-1/48}\right),
\end{aligned}$$

where the third inequality comes from counting the number of ℓ such that $\mu_3(\ell) = 10N^{1/8}$, which is equal to the number of ℓ such that $2e\frac{\ell^{3/2}}{\sqrt{N}} \leq 10N^{1/8}$. □

B.3 Proof of Equation (14)

Here, we give a proof of Equation (14). Starting from Equation (13), we have that:

$$\begin{aligned}
g_{i,k} &\leq g_{i-1,k} + \sqrt{2}\left(\sqrt{\frac{\mu_3(i-1)}{N}} + \frac{i-1}{N}\right) g_{i-1,k-1} + \sqrt{2}\cdot f^{col}_{i-1,\mu_3(i-1)} \\
&\vdots \\
&\leq \sqrt{2}\sum_{\ell=0}^{i-1}\left(\left(\sqrt{\frac{\mu_3(\ell)}{N}} + \frac{\ell}{N}\right) g_{\ell,k-1} + f^{col}_{\ell,\mu_3(\ell)}\right) \\
&\leq \sqrt{2}\sum_{\ell=0}^{i-1}\left(\left(\sqrt{\frac{\mu_3(\ell)}{N}} + \frac{\ell}{N}\right) g_{\ell,k-1} + \left(\frac{1}{2}\right)^{10N^{1/8}}\right) \\
&\leq \sum_{\ell=0}^{i-1}\sqrt{2}\left(\sqrt{\frac{\mu_3(\ell)}{N}} + \frac{\ell}{N}\right) g_{\ell,k-1} + \sqrt{2}\cdot 2^{-10N^{1/8}}\cdot N^{1/2} \\
&\leq \sum_{\ell=0}^{i-1}\sqrt{2}\left(\sqrt{\frac{\mu_3(\ell)}{N}} + \frac{\ell}{N}\right) g_{\ell,k-1} + \sqrt{2}\cdot 2^{-9.5N^{1/8}},
\end{aligned}$$

where the second inequality comes from the recursion on the first term $g_{i-1,k}$, and using the fact that $g_{0,k} = 0$. For the third inequality, we used Lemma 7 and the definition of μ_3. Expanding recursively inside the sum, we have:

$$\begin{aligned}
g_{i,k} &\leq \sum_{\ell=0}^{i-1} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell)}{N}} + \frac{\ell}{N}\right) g_{\ell,k-1} + \sqrt{2}\cdot 2^{-9.5N^{1/8}} \\
&\leq \sum_{\ell_1=0}^{i-1} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_1)}{N}} + \frac{\ell_1}{N}\right)\left(\sum_{\ell_2=0}^{\ell_1} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_2)}{N}} + \frac{\ell_2}{N}\right) g_{\ell_2,k-2}\right. \\
&\quad \left. + \sqrt{2}\cdot 2^{-9.5N^{1/8}}\right) + \sqrt{2}\cdot 2^{-9.5N^{1/8}} \\
&\vdots \\
&\leq \sum_{0\leq \ell_k<\ell_{k-1}<\cdots<\ell_1<i} \prod_{j=1}^{k} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_j)}{N}} + \frac{\ell_j}{N}\right) \\
&\quad + \sqrt{2}\cdot 2^{-9.5N^{1/8}} \sum_{t=0}^{k-1} \sum_{0\leq \ell_t<\ell_{t-1}<\cdots<\ell_1<i} \prod_{j=1}^{t} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_j)}{N}} + \frac{\ell_j}{N}\right) \\
&\leq \frac{A_i^k}{k!} + \sqrt{2}\cdot 2^{9.5N^{1/8}} \sum_{t=0}^{k-1} \frac{A_i^t}{t!} \\
&\leq \frac{A_i^k}{k!} + \sqrt{2}\cdot e^{A_i} 2^{9.5N^{1/8}},
\end{aligned}$$

where the third inequality comes from expanding recursively all of the terms $g_{\ell_t,k-t}$, and using the fact that $g_{\ell,0} = 1$. The fourth inequality comes from the fact that:

$$\begin{aligned}
&\sum_{0\leq \ell_k<\ell_{k-1}<\cdots<\ell_1<i} \prod_{j=1}^{k} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_j)}{N}} + \frac{\ell_j}{N}\right) \\
&\leq \frac{1}{k!} \sum_{0\leq \ell_k,\ell_{k-1},\ldots,\ell_1<i} \prod_{j=1}^{k} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_j)}{N}} + \frac{\ell_j}{N}\right) \\
&= \frac{1}{k!} \prod_{j=1}^{k} \sum_{0\leq \ell_j<i} \sqrt{2}\left(\sqrt{\frac{\mu_3(\ell_j)}{N}} + \frac{\ell_j}{N}\right) \\
&= \frac{1}{k!} \prod_{j=1}^{k} A_i \\
&= \frac{A_i^k}{k!}.
\end{aligned}$$

B.4 Proof of Lemma 21

Proof. We have that:

$$\begin{aligned} A_{i,s} &= \sum_{\ell=0}^{i-1} \left(\sqrt{(s-1)\cdot\frac{\mu_s(\ell)}{N}} + \left(\frac{\ell}{N}\right)^{s/2} + \left(\sum_{r=2}^{s}\frac{\ell}{N^r}\right)^{1/2} \right) \\ &= \sqrt{s-1}\sum_{\ell=0}^{i-1}\sqrt{\frac{\mu_s(\ell)}{N}} + \sum_{\ell=0}^{i-1}\left(\frac{\ell}{N}\right)^{s/2} + \sum_{\ell=0}^{i-1}\left(\sum_{r=2}^{s}\frac{\ell}{N^r}\right)^{1/2}. \qquad (18) \end{aligned}$$

Notice that

$$\begin{aligned} &\sum_{\ell=0}^{i-1}\sqrt{\frac{\mu_s(\ell)}{N}} \\ &= \sum_{\ell:\mu_s(\ell)=10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}} \sqrt{\frac{10\cdot s^2\cdot \Pi_{s-1}\cdot N^{1/2^s}}{N}} \qquad (19) \\ &\qquad + \sum_{\ell:\mu_s(\ell)>10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}} \sqrt{\frac{\mu_s(\ell)}{N}} \\ &\le \sum_{\ell:\mu_s(\ell)=10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}} \sqrt{\frac{10\cdot s^2\cdot \Pi_{s-1}\cdot N^{1/2^s}}{N}} \qquad (20) \\ &\qquad + \sum_{\ell=0}^{i-1}(2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\frac{\ell^{(2^{s-1}-1)/2^{s-1}}}{N^{(2^{s-2}-1)/2^{s-1}}}\cdot N^{-1/2}\cdot\sqrt{\Pi_{s-1}}, \end{aligned}$$

where we replaced $\mu_s(\ell)$ by its value, and the inequality comes from the fact that there cannot be more than i values such that $\mu_s(\ell) > 10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}$. The second summation is at most $(2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}}\cdot\sqrt{\Pi_{s-1}}$.

For the first summation of Equation (20), we need to count the values of ℓ such that $\mu_s(l) = 10s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}$. By using the definition of $\mu_s(\ell)$, this quantity corresponds to the number of ℓ that satisfies:

$$\begin{aligned} &\Pi_{s-1}\cdot(2e)^{\frac{2^{s-2}-1}{2^{s-3}}}\frac{\ell^{(2^{s-1}-1)/2^{s-2}}}{N^{(2^{s-2}-1)/2^{s-2}}} \le 10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s} \\ \Leftrightarrow &\ell \le \left(\frac{10}{(2e)^{\frac{2^{s-2}-1}{2^{s-3}}}}\right)^{2^{s-2}/(2^{s-1}-1)} \cdot N^{\left(\frac{1}{2^s}+\frac{2^{s-2}-1}{2^{s-2}}\right)\frac{2^{s-2}}{2^{s-1}-1}}\cdot s^{\frac{2^s}{2^{s-1}-1}} \\ \Leftrightarrow &\ell \le O\left(s^{\frac{2^s}{2^{s-1}-1}}\cdot N^{\left(\frac{1}{2^s}+\frac{2^{s-2}-1}{2^{s-2}}\right)\frac{2^{s-2}}{2^{s-1}-1}}\right). \end{aligned}$$

Thus the first summation of Equation (20) is upper-bounded by:

$$\begin{aligned}
&\sum_{\ell:\mu_s(\ell)=10\cdot\Pi_{s-1}\cdot N^{1/2^s}} \sqrt{\frac{10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}}{N}} = \\
&\sqrt{\frac{10\cdot s^2\cdot\Pi_{s-1}\cdot N^{1/2^s}}{N}}\cdot O\left(s^{\frac{2^s}{2^{s-1}-1}}\cdot N^{\left(\frac{1}{2^s}+\frac{2^{s-2}-1}{2^{s-2}}\right)\frac{2^{s-2}}{2^{s-1}-1}}\right) \\
&\leq O\left(N^{-\frac{1}{2}+\frac{1}{2^{s+1}}+\frac{2^{s-3}}{4(2^{s-1}-1)}}\cdot s^4\cdot\sqrt{\Pi_{s-1}}\right) \\
&\leq O\left(N^{\frac{-2^{2s-1}+2^s+2^{s-1}-1+2^{2s-4}}{2(2^s-2)}}\cdot s^4\cdot\sqrt{\Pi_{s-1}}\right) \\
&\leq O\left(N^{-1/(2^s(2^s-2))}\cdot s^4\cdot\sqrt{\Pi_{s-1}}\right) = O\left(N^{-1/(2^s(2^s-2))}\cdot s^4\cdot\Pi_s\right),
\end{aligned}$$

where for the first inequality we use that $\frac{2^s}{2^{s-1}-1}+1\leq 4$ for all $s\geq 3$.

Therefore, we have that:

$$\sum_{\ell=0}^{i-1}\sqrt{\frac{\mu_s(\ell)}{N}} \leq (2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}}\sqrt{\Pi_{s-1}}+O\left(N^{-1/(2^s(2^s-2))}\cdot s^4\cdot\Pi_s\right). \tag{21}$$

For the second term of Equation (18), we have:

$$\begin{aligned}
\sum_{\ell=0}^{i-1}\left(\frac{\ell}{N}\right)^{s/2} &\leq \sum_{\ell=0}^{i-1}\left(\frac{\ell}{N}\right) \\
&\leq \sum_{\ell=0}^{i-1}\left(\frac{\ell}{N}\right)^{(2^{s-1}-1)/2^{s-1}},
\end{aligned} \tag{22}$$

where we use that $s\geq 3$ and $1\geq(2^{s-1}-1)/2^{s-1}$. And for the third term,

$$\begin{aligned}
\sum_{\ell=0}^{i-1}\left(\sum_{r=2}^{s}\frac{\ell}{N^r}\right)^{1/2} &\leq \sum_{\ell=0}^{i-1}\left((s-1)\frac{\ell}{N^2}\right)^{1/2} \\
&\leq \sum_{\ell=0}^{i-1}\left(\sqrt{s-1}\frac{\ell}{N}\right) \\
&\leq \sum_{\ell=0}^{i-1}\left(\sqrt{s-1}\left(\frac{\ell}{N}\right)^{(2^{s-1}-1)/2^{s-1}}\right),
\end{aligned} \tag{23}$$

where we used that $r\geq 2$ for the second inequality, and that $1\geq(2^{s-1}-1)/2^{s-1}$ for the last inequality. Thus, the sum of the terms of Equation (22) and Equation (23) is smaller than the second term of Equation (20) (since $(2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\geq 2$). Combining Equation (18), Equation (21), Equation (22) and Equation (23) yields that:

$$\begin{aligned}
A_{i,s} &\leq 2\cdot(2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}}\cdot\sqrt{s-1}\cdot\sqrt{\Pi_{s-1}} \\
&\quad + O\left(N^{-1/(2^s(2^s-2))}\cdot s^4\cdot\Pi_s\right) \\
&= (2e)^{\frac{2^{s-2}-1}{2^{s-2}}}\frac{i^{(2^s-1)/2^{s-1}}}{N^{(2^{s-1}-1)/2^{s-1}}}\cdot\Pi_s+O\left(s^4\cdot\Pi_s\cdot N^{-1/(2^s(2^s-2))}\right).
\end{aligned}$$

□

B.5 Proof of Equation (17)

We have

$$\begin{aligned} g_{i,k} &\leq \left(\sum_{\ell=0}^{i-1} B_{\ell,s} \cdot g_{\ell,k-1}\right) + s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}} \\ &\leq \left(\sum_{\ell_1=0}^{i-1} B_{\ell_1,s}\left(\sum_{\ell_2=\ell_1}^{i-1} B_{\ell_2,s} \cdot g_{\ell_2,k-1} + s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}\right)\right) \\ &\qquad + s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}. \end{aligned}$$

We get by induction

$$\begin{aligned} g_{i,k} \leq &\left(\sum_{\ell_1=0}^{i-1} B_{\ell_1,s}\left(\sum_{\ell_2=\ell_1}^{i-1} B_{\ell_2,s}\left(\sum_{\ell_3=\ell_2}^{i-1} B_{\ell_3,s}\cdots\right.\right.\right. \\ &\left.\left.\left.+ s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}\right) + s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}\right)\right) \\ &\qquad + s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}. \end{aligned}$$

We thus obtain

$$\begin{aligned} g_{i,k} \leq &\left(\sum_{0\leq\ell_k<\ell_{k-1}<\cdots<\ell_1<i} \prod_{j=1}^{k} B_{\ell_j,s}\right) \\ &+ s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}} \cdot \sum_{t=0}^{k-1} \sum_{0\leq\ell_t<\ell_{t-1}<\cdots<\ell_1<i} \prod_{j=1}^{t} B_{\ell_j,s}, \end{aligned}$$

and finally

$$\begin{aligned} g_{i,k} \leq &\frac{A_{i,s+1}^k}{k!} + \sum_{\ell=0}^{k-1} \frac{A_{i,s+1}^{\ell}}{\ell!} \cdot s^{3/2} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}} \\ \leq &\frac{A_{i,s+1}^k}{k!} + s^{3/2} \cdot e^{A_{i,s+1}} \cdot 2^{-9.5\cdot(s+1)^2\cdot\Pi_s\cdot N^{1/2^{s+1}}}. \end{aligned}$$

B.6 Proof of Lemma 31

Proof. Similarly to the proof of Lemma 29, we can consider that there are $O\left(N^{k-j+1}\right)$ marked elements in the function F_1. Hence, using Theorem 35, the second step of the algorithm is expected to make

$$O\left(t \cdot \sqrt{\frac{N^k}{N^{k-j+1} \cdot \binom{k}{j}}}\right) = O\left(\frac{t}{\sqrt{\binom{k}{j}}} \cdot N^{(j-1)/2}\right)$$

quantum queries to the oracle.

Similarly to the proof of Lemma 29, we can consider that there are $t \cdot N^{j-1} \cdot \frac{k!}{(j-1)!}$ marked elements in the function F_2. Hence, using Theorem 35, the fourth step of the algorithm is expected to make

$$O\left(\sqrt{\frac{N^k}{t \cdot N^{j-1} \cdot \frac{k!}{(j-1)!}}}\right) = O\left(\frac{N^{(k-j+1)/2}}{\sqrt{t}} \cdot \sqrt{\frac{(j-1)!}{k!}}\right)$$

quantum queries to the oracle.

By picking $t = N^{(k-2j+2)/3}$ with $j \leq \frac{k+2}{2}$, the complexity of the algorithm is

$$O\left(N^{(k-2j+2)/3} \cdot N^{(j-1)/2} \cdot \left(\sqrt{\frac{1}{\binom{k}{j}}} + \sqrt{\frac{j!}{k!}}\right)\right) = O\left(\frac{N^{(2k-j+1)/6}}{\sqrt{\binom{k}{j}}}\right)$$

quantum queries to the oracle. □

B.7 Proof of Lemma 33

Proof. We first prove the result when k is divisible by $r+1$. The result holds for $r = 1$ (using Algorithm 1 and Lemma 29).

Fix $r > 2$, and assume the result holds for $r-1$.

The first step of the algorithm is expected to make $O\left(t \cdot N^{k'/2r}\right)$ quantum queries to the oracle if k' is divisible by r.

Similarly to the proof of Lemma 29, we can consider that there are $t \cdot N^{k'}$ marked elements in the function F_1. Hence, using Theorem 35, the third step of the algorithm is expected to make $O\left(\sqrt{\frac{N^k}{t \cdot N^{k'}}}\right)$ quantum queries to the oracle.

Picking $t = N^{(rk-rk'-k')/3r}$ gives a complexity $O\left(N^{(2rk+(1-2r)k')/6r}\right)$.

By picking $k' = \frac{r}{r+1}k$, k' is an integer since k is divisible by $r+1$. Moreover, k' is divisible by r and the complexity becomes $O\left(N^{k/(2+2r)}\right)$.

If k is not divisible by $r+1$, then there is a k' between k and $k+r+1$ such that k' is divisible by $r+1$. Then, we can use Algorithm 3 to find a (r,k')–SC with the same functions $h_1, \ldots, h_k$ and new random functions $h_{k+1}, \ldots, h_{k'}$. This gives us a (r,k)–SC for the functions $h_1, \ldots, h_k$, and the quantum query complexity is

$$O\left(N^{k'/(2+2r)}\right) \leq O\left(N^{(k+r+1)/(2+2r)}\right).$$

□