Quantum Multi-Solution Bernoulli Search with Applications to Bitcoin's Post-Quantum Security

Alexandru Cojocaru[1], Juan Garay[2], Aggelos Kiayias[3], Fang Song[4], and Petros Wallden[5]

[1]University of Maryland
[2]Texas A&M University
[3]University of Edinburgh and IOHK
[4]Portland State University
[5]University of Edinburgh

A proof of work (PoW) is an important cryptographic construct which enables a party to convince other parties that they have invested some effort in solving a computational task. Arguably, its main impact has been in the setting of cryptocurrencies such as Bitcoin and its underlying blockchain protocol, which have received significant attention in recent years due to its potential for various applications as well as for solving fundamental distributed computing questions in novel threat models. PoWs enable the linking of blocks in the blockchain data structure, and thus the problem of interest is the feasibility of obtaining a sequence ("chain") of such proofs.

At the same time, the rapid development in quantum computing makes the threats to cryptography more and more concerning. In this work, we examine the hardness of finding such *chain* of PoWs against quantum strategies. We prove that the chain of PoWs problem reduces to a problem we call *multi-solution Bernoulli search*, for which we establish its quantum query complexity. Effectively, this is an extension of a threshold direct product theorem to an *average-case* unstructured search problem. Our proof, adding to active recent efforts, simplifies and generalizes the recording technique due to Zhandry [Crypto 2019].

As an application, we revisit the formal treatment of security of the core of the Bitcoin consensus protocol, called the *Bitcoin backbone* [Eurocrypt 2015], in a setting where the adversary has quantum capabilities while the honest parties remain classical, and show that the protocol's security holds under a quantum analogue of the classical "honest majority" assumption that we formulate. Our analysis indicates that the security of the Bitcoin backbone protocol is guaranteed provided that the number of adversarial quantum queries is bounded so that each quantum query is worth $O(p^{-1/2})$ classical ones, where p is the probability of success of a single classical query to the protocol's underlying hash function. Somewhat surprisingly, the wait time for safe settlement of transactions in the case of quantum adversaries matches (up to a constant) the safe settlement time in the classical case.

Alexandru Cojocaru: cojocaru@umd.edu
Juan Garay: garay@cse.tamu.edu
Aggelos Kiayias: akiayias@inf.ed.ac.uk
Fang Song: fang.song@pdx.edu
Petros Wallden: petros.wallden@ed.ac.uk

Contents

1 Introduction **2**
- 1.1 Our Contributions . . . 3
- 1.2 Technical Overview . . . 6
  - 1.2.1 Quantum Query Complexity of Multi-solution Bernoulli Search and Chain-of-PoWs . . . 6
  - 1.2.2 Bitcoin Backbone Security against Quantum Adversaries . . . 7
- 1.3 Related Work . . . 9

2 Preliminaries **9**
- 2.1 The Recording Oracle Technique . . . 9

3 Query Complexity of k-BerSearch **10**
- 3.1 Recording Technique Framework for Bernoulli Distributions . . . 11
- 3.2 Analysis of k-BerSearch via the Bernoulli Recording Technique . . . 13
- 3.3 Bounding the Success Probability with Progress Measure . . . 15
- 3.4 Bounding the Progress Measure . . . 16
- 3.5 Putting Everything Together . . . 18

4 The Bitcoin Backbone against Quantum Adversaries **18**
- 4.1 The Bitcoin Backbone Protocol: Model and Definitions . . . 18
- 4.2 The Chain-of-PoWs Problem . . . 20
- 4.3 Hardness of Chain-of-PoWs . . . 21
- 4.4 Post-Quantum Security of the Bitcoin Backbone Protocol . . . 22

A Comparison with Other Query Complexity Work **30**

B Simplified Bound of Theorem 3.1 **31**

1 Introduction

A *proof of work* (PoW) enables a party to convince other parties that considerable effort has been invested in solving a computational task. First introduced by Dwork and Naor [1], PoWs serve as candidate solutions to thwarting spam emails and denial-of-service attacks. More recently in the setting of cryptocurrencies, PoWs have proven indispensable and manifested profound impact, where they lay the basis for consensus protocols in permissionless blockchains. In particular, Bitcoin [2] and its underlying blockchain protocol have drawn remarkable attention for the potential to resolve fundamental distributed computing problems in various threat models, as well as to enable novel applications.

In the blockchain setting, the objective of a PoW is to confirm new transactions to be included in the blockchain. To successfully create a PoW in Bitcoin, one needs to find a value ("witness") such that evaluating a hash function (SHA-256) on this value together with (the hash of) the last block and new transactions to be incorporated, yields an output below a threshold. A party who produces such a PoW gets to append a new block to the blockchain and is rewarded. A *blockchain* hence consists of a sequence of such *blocks*. Each party maintains such a blockchain, and attempts to extend it via solving a PoW. We refer to the task of creating a chain of multiple blocks as the Chain-of-PoWs problem. In more detail, we define these blockchain-inspired problems as follows.

Definition (Blockchain PoW—Informal). *Given a hash function h, a positive integer T, and a string z representing the hash value of the previous block, the goal is to find a value ctr such that:*

$$h(ctr, z) \leq T\,.$$

Definition (Chain-of-PoWs—Informal). *Given a family of hash functions $h_i : X \times Y \to X$, an initial value $x_0 \in X$, the goal is to output a* chain of length k *consisting of $y_0, \ldots, y_{k-1} \in Y$ such that:*

$$x_{i+1} := h_i(x_i, y_i) \text{ and } x_{i+1} \leq T \text{ for all } i \in \{0, \cdots, k-1\}\,.$$

Intuitively, the hardness of the Chain-of-PoWs problem is crucial to a robust blockchain and trustworthy Bitcoin applications. However, due to the complex nature of these protocols, obtaining a firm security claim in a formal mathematical model turns out to be challenging. At Eurocrypt 2015, Garay *et al.* [3] developed an abstraction of Bitcoin's underlying protocol termed the "Bitcoin backbone," which inspired a series of other formal treatments of Bitcoin (e.g., [4, 5, 6]). In essence, the backbone abstraction of [3] models the hash function of the PoW as a random oracle (RO) [7], and it assumes a *uniform* configuration among a fixed (albeit unknown) number of parties ("miners"), a fraction of which may behave arbitrarily as controlled by an adversary. Namely, the parties are endowed with the same computational power, as measured by the permitted number of queries to the RO per round. Following a modular approach, essential security properties of the backbone protocol, such as *common prefix* and *chain quality*, are formulated, which are proven sufficient to realize blockchain applications, notably a robust public transaction ledger (a.k.a. "Nakamoto consensus"). These properties, relying crucially on the hardness of Chain-of-PoWs, can then be established assuming *honest majority* of computational power.

The formal treatment in the works above assumes that parties are instantiated by "classical" computers. However, we are witnessing a rapid development in quantum computing bringing concerning threats to cryptographic protocols. For example, it is known that quantum computers can cause devastating breaks to both public key cryptography [8] and symmetric-key cryptography [9, 10]. In addition, the unique features of quantum information, such as intrinsic randomness and no-cloning, render many classical security analyses (e.g., rewinding [11, 12, 13]) obsolete, and even the right *modeling* of security can be elusive (e.g., [14, 15, 16]). In particular, existing analyses of blockchain protocols are conducted in the random oracle model. When quantum attackers are present, Boneh *et al.* [17] argued the need for granting the attackers the ability to query the random oracle in *quantum superposition*. This gives rise to the *quantum random oracle* (QRO) model, which is far more challenging to reason about (see Section 1.3 for an account). Several fundamental questions need to be revisited, such as:

> *What is the complexity of* Chain-of-PoWs *in the quantum query model? Can we formally establish the quantum security of the Bitcoin backbone protocol, by, for example, modifying the framework of [3] to work against quantum adversaries in the QRO model?*

In the remainder of this section we summarize our contributions, provide a technical overview, and discuss closely related works. The full presentation of our results then follows.

1.1 Our Contributions

In this work, we prove the quantum query complexity of Chain-of-PoWs, and then formally establish the quantum security of the Bitcoin backbone protocol in the QRO model.

First, consider a family of independent and uniformly random functions $\{h_i : X \times Y \to X\}$, given as quantum unitary black-boxes. Let $p := T/|X|$ be the fraction of target outputs in the co-domain, which in the context of Bitcoin represents the *difficulty* of mining. We prove the following:

Theorem 1.1 (Informal)**.** *For any quantum adversary $\mathcal{A}$ having N quantum queries, the probability that $\mathcal{A}$ solves the* Chain-of-PoWs *problem by outputting a chain of size at least k, is at most*

$$P(N,k) \leq \left(e^2 \left(N/k+1\right)^2 p\right)^k .$$

We show this by considering a variant in which the solution does not need to form a chain. This amounts to searching multiple inputs in a non-uniform random function, and its quantum query hardness immediately implies the one for Chain-of-PoWs, which we derive precisely. More specifically, we have a Boolean function whose output is sampled according to a Bernoulli distribution *independently* for each input, and we wish to find multiple preimages of 1 (call it k-BerSearch). We prove the quantum query complexity of this search problem by extending Zhandry's elegant recording technique [18] to point-wise independent *non-uniform* distributions. In fact, this establishes a quantum *average-case* strong direct-product theorem (SDPT) for this family of non-uniform search problems[1].

Turning to the quantum security of the Bitcoin backbone protocol, we examine all the components of the [3] framework in the presence of classical honest parties against quantum adversaries in the QRO model, and our guiding principle is to "lift" it to the quantum setting with *as few* changes as possible. One crucial change in our analysis is to reformulate a central analytical tool in [3] named *typical execution*, since in the presence of quantum adversaries it becomes ill-defined. Assisted by our quantum complexity bound for Chain-of-PoWs, we are able to identify a quantum analogue of the *honest majority* condition, under which the desired security properties of the Bitcoin backbone protocol follow. As a side benefit, the resulting analysis makes the reduction to Chain-of-PoWs explicit, which further improves the degree of modularity of the classical analysis.

We elaborate on our results and how they are obtained in the technical overview (Section 1.2). Before that, we highlight some interesting aspects of our results and put them in perspective.

Quantum query complexity of Chain-of-PoWs and k-BerSearch. Roughly speaking, the formal query bounds we show match the intuitive argument based on Grover's quantum search algorithm, which at first glance may not look surprising. Instead, we view the following values potentially more impactful.

- Our proof provides another working example demonstrating the power and potential of the recording technique. While the conceptual idea behind the original technique is remarkably neat, the *execution* is often a lot more complex, which in turn dictates the success or failure of applying the technique. This is also the case in many recent generalizations. Our execution of a particular non-uniform recording technique is carefully specified and "packaged" in clean modules, and could be treated as a template to derive other proofs based on it.

  We note that the basic idea of non-uniform extensions is natural and has already been investigated (e.g., [20]). In particular, in independent work, Hamoudi and Magniez [21] consider essentially the same search problem as we do (k-BerSearch), proving a quantum

[1] It is actually a *threshold* SDPT, which is usually stronger. The distinction is not essential to our discussion, and interested readers are referred to [19].

query bound by extending Zhandry's technique[2]. In fact, if the sole goal is to show a query bound for the search problem, not necessarily as tight as what we obtain, it is plausible that one can simulate non-uniform functions by uniform functions and then resort to the original uniform recording technique (cf. [22]). Again, the merit would lie in the execution, and in any case a stand-alone treatment of a non-uniform case remains beneficial.

- As also noted in [21], the quantum query complexity of the search problem establishes an *average-case* strong direct product theorem (SDPT) via the recording technique, which is combinatorial in nature. SDPTs are highly desirable in a host of theoretical models [23, 24, 25, 19], and it basically assert that the success probability of solving multiple instances of a problem drops exponentially unless investing equal multiples of the resource for solving one instance. In the context of quantum query complexity, SDPT has been extensively investigated, where the *polynomial method* [26] and the *(generalized) adversary method* [27] are the two dominating approaches to prove it. The polynomial method builds upon *analytical* properties of polynomials, while the ("modern") adversary method often resorts to advanced *algebraic* tools such as representation theory. In contrast, the combinatorial approach based on the recording technique, which usually follows more elementary and intuitive arguments, could open a new route to establishing a SDPT. Besides, existing examples of SDPT proven by either method (polynomial or adversary method) are typically for *worst-case* problems and usually difficult to extend to the *average-case*. This for instance is the case for many examples of SDPT for the standard unstructured search [28, 29, 30, 31]. Our new approach via the recording technique seems especially well suited to reason about *average-case* problems and also to induce hardness bounds needed in the cryptographic setting.

Quantum security of the Bitcoin backbone protocol. Once the quantum hardness of Chain-of-PoWs is established, it would be tempting to "plug" it into the classical framework of [3] in order to derive its quantum security. However, we stress that, as more and more works demonstrate, when quantum adversaries are present every link of the provable-security framework needs to be reexamined, including the attack models, security goals, and security reductions.

Specifically, it can be justified that the security goals of the Bitcoin backbone protocol (e.g., common prefix, chain quality) may stay unchanged fortunately. But as soon as the attacks come into play, we observe a crucial simplification assumption in [3] that bears no clear quantum analogue. In a nutshell, classically it is without loss of generality to assume that an adversary, just as honest users, solves every PoW simply by querying the random oracle on a sequence of inputs. As a result, the adversarial behavior is fixed and the protocol can be described by an explicit random experiment, in which we can conduct probabilistic analysis of various events such as those indicating secure configurations of the system. This is how the analysis proceeds in [3], by formulating a central notion of *typical executions*, which enables establishing the desired security properties under an honest-majority condition. In our case, since we can no longer assume a specific adversarial strategy that includes all possible quantum attacks, the above simplification becomes unsound.

Our solution introduces a simple remedy which allows inheriting the classical analysis to a large extent. It boils down to an alternative and stronger characterization of typical executions for the quantum setting. Although this characterization is not as intuitive as in the classical case, it enables the analysis of quantum adversaries and proof that the backbone protocol's security

[2]The formulation of the search problem in [21] is in a slightly restricted form, but their analysis should in principle work for the general version we consider here. We also need strengthened bounds to derive tighter security for the Bitcoin backbone protocol. See Appendix A for further discussion.

goals under a post-quantum honest-majority condition we identify. In addition, as we mentioned earlier, the security of the protocol is explicitly reduced to the hardness of Chain-of-PoWs. In turn, this leads to a more modular analysis even in the classical setting, and might help in the quantum security analysis of other blockchain-based protocols [5, 32, 33].

Finally, we remark that the focus of this work is the simplified scenario in which only the adversary is quantum-capable, while the honest parties remain classical. We consider this analysis a first step towards the more general scenario, that also has independent interest in its own right since it address a realistic practical situation that the near-term development of quantum technologies may bring us. Specifically, imagine that large quantum computers are not available to the public, due to their price and required technologies, hence most users participating in the Backbone Bitcoin would only own classical computing devices. Large quantum computers would only be held by a few big companies or states, where most of them would not be willing or interested in participating in the bitcoin blockchain. The question we address is what would happen if one of those very few but powerful players decided to attempt to compromise the security of a PoW-based blockchain. The ultimate aim would be to address the full quantum setting, in which there could be an arbitrary number of quantum-capable players (both honest and adversarial), and this remains an important direction for future research.

1.2 Technical Overview

1.2.1 Quantum Query Complexity of Multi-solution Bernoulli Search and Chain-of-PoWs

To show the quantum query complexity of the Chain-of-PoWs problem (Theorem 1.1), we consider a simpler variant (for the adversary), called k-BerSearch, which relaxes the requirement that the PoWs need to form a chain. Bounding the adversary's performance in the simpler problem readily provides a bound for the Chain-of-PoWs problem.

Hence, we focus on the quantum query complexity of k-BerSearch, and prove the main bound on the success probability for solving it.

Definition (k-BerSearch). *Given a Boolean function f as a black-box, such that each input x is independently assigned the value $f(x) = 1$ with probability p, find k distinct preimages of 1 for f.*

Theorem 1.2 (Informal)**.** *For any quantum adversary $\mathcal{A}$ having N quantum queries, the probability that $\mathcal{A}$ solves k-BerSearch is bounded by:*

$$p_k^N \leq 4(1-p)p^k \left(\sum_{i=0}^{k} (\sqrt{1-p})^i \cdot \binom{N}{i} \right)^2 \leq \frac{1}{k} \cdot \left(\frac{eN\sqrt{p}}{k} \right)^{2k}. \tag{1.1}$$

Our proof of Theorem 1.2 relies on the recording technique of quantum queries in the quantum random oracle model due to Zhandry [18], designed for uniformly sampled functions. We modify the framework to accommodate functions sampled according to Bernoulli distributions. In Zhandry's recording technique, the key observation is that the oracle holds a private register with the (uniform) superposition over all possible functions. Then if we view the oracle register in the Fourier domain (i.e., applying a Fourier transformation on it), each query modifies *exactly one position* (from 0 to 1, or the other way around).

In our setting, the oracle's state will now be a Bernoulli superposition of the form $\sum_f \sqrt{\alpha_f} \, |f\rangle$, where α_f is the probability to sample the function f according to a Bernoulli distribution. The view between the standard and Fourier domains in the uniform does not hold anymore. Instead, we consider a different dual domain, and extend the primal-dual perspective to Bernoulli

distributions. Switching between the primal and dual domain is based on a unitary U_p, which essentially rotates around the Z axis depending on the probability parameter p, which specializes to the Hadamard transformation in the uniform case when $p = 1/2$. As a result, in the dual domain we start with the oracle state being initialized in the all-0 state, and each adversarial quantum query will rotate exactly one of its positions (in superposition) using the U_p transformation. Consequently, after N quantum queries we will have at most N rotated qubits in the oracle register and the rest of the qubits will remain 0. Intuitively, this gives us a bound on the adversary's knowledge by examining how many non-zero entries are in the oracle's register.

Having established a relation between solving the k-BerSearch problem and the oracle's property of having at least k non-zero entries in its register, we just need to bound the probability of the latter event. The final step is achieved by bounding the "progress" made in enhancing the probability amplitude of the desired oracle's states after each query.

From the obtained bound on the k-BerSearch problem, we can also derive a bound on the harder variant, the Chain-of-PoWs problem.

Lemma 1.3 (Informal)**.** *The probability of solving Chain-of-PoWs using N quantum queries is at most the probability of solving k-BerSearch using $N + k$ quantum queries.*

1.2.2 Bitcoin Backbone Security against Quantum Adversaries

We assume a single quantum adversary who runs a computation up to depth Q per round, i.e., that the adversary makes at most Q superposition queries sequentially in each round to the QRO. When we consider s consecutive rounds, we denote $N = sQ$ the total number of queries to the QRO in s rounds, which is also the depth of the quantum computation in s rounds[3]. We "lift" the classical analytical framework of [3] to the quantum setting in the following steps:

1. Define the notion of *post-quantum typical execution* of a blockchain in the presence of honest players and a quantum adversary. This is a critical step towards modularizing the analysis;
2. show that under a post-quantum typical execution, the two desired properties, common prefix and chain quality, follow for suitable choices of parameters;
3. identify a post-quantum *honest majority* condition under which a post-quantum typical execution occurs with high probability.

From our main theorem regarding the query complexity of the Chain-of-PoWs problem (Theorem 1.1), we can bound two relevant quantities: (i) the expected length of an adversarial chain, and (ii) the (overwhelming) probability that this bound holds. The former leads to a post-quantum "honest majority" condition, where we require that the total number of quantum queries Q per round of the attacker, has to be less than the total number of classical queries of all the honest parties divided by an extra $O(p^{-1/2})$ factor, where p is the probability of success of a single query and, informally, represents the difficulty level of the PoW. To get this condition, we examine closely the requirements for a typical execution and ensure that the honest chain exceeds by a small constant factor the largest chain that adversaries can make with non-negligible probability.

The latter determines the wait time for safe settlement, which intuitively represents the number of rounds s that need to pass in order to ensure that the common prefix and chain quality properties hold except with the same negligible probability as in the classical case. Our analysis indicates that the required number of rounds matches, up to a constant, the number of rounds required in the classical adversarial setting. This somewhat surprising result implies

[3]One can also think of a scenario where the adversary controls multiple quantum processors and the overall depth is smaller (see end of Section 4.4).

that for post-quantum security, the number of "block confirmations" necessary for a transaction to be accepted in order to protect against double-spending, is almost the same as in the classical setting.

Finally, we emphasize that our work, apart from setting different parameters, preserves the generality of the classical analysis of [3]. For instance, it captures strategies correlated with honest-parties' actions as well as long-term attacks (such as selfish mining [34]).

Interpretation of results for Bitcoin security. We can now provide a comparison between the analysis of the Bitcoin backbone protocol against classical adversaries of [3] and the current analysis against general quantum adversaries. First off, let us summarize the parameters in the Bitcoin backbone analysis:

n: # honest parties	q: # honest classical queries per round
Q: # adversarial quantum queries per round	f: prob. at least one honest party generates a PoW in a round
ϵ: concentration quality of random variables	κ: security parameter
k: # blocks for common prefix	μ: chain quality parameter
s: # rounds (We refer to Section 4.1 for the definition of round.)	p: prob. of success of a single classical query

Table 1: *Parameters used in our analysis.*

Table 2 shows the comparison of results obtained in the two adversarial settings. The relevant conditions and quantities are:

- "Honest Majority," which expresses the relation between the honest hashing power and the (classical or quantum) hashing power of the adversary; the expected number of adversarial blocks in s consecutive rounds;
- the probability of a "typical execution" (i.e., the probability that the required bounds on the number of adversarial queries hold); and
- the number of rounds required to reach the same level of security as in the classical adversaries setting.

	Classical Adversary	General Quantum Adv. (this work)
Honest Majority	$\frac{t}{n-t} < 1 - 3(f+\epsilon)$ [3] t = number (classical) adv.	$Q \leq \frac{(1-\epsilon)f(1-f)}{(1+\epsilon)e\sqrt{p}}$
Maximum Expectation of Adversarial PoWs	$pqt \cdot s$ [3]	$\leq (1+\epsilon)\sqrt{e^2 p} \cdot Q \cdot s$
Probability of Concentration	$P_{\text{cl}} = 1 - e^{-\Omega(\epsilon^2 f s)}$ [3]	$P_{\text{q}} = 1 - e^{-\Omega((1-\epsilon)f(1-f)s}$
Number of Rounds	s_{cl}	$s_{\text{q}} = O\left(\frac{\epsilon^2}{(1-\epsilon)(1-f)} \cdot s_{cl}\right)$

Table 2: *Classical vs quantum adversaries' bounds.*

From Table 2 we highlight two main aspects. First, noting that the probability that at least one honest party generates a PoW in a round is $f = npq$, (we emphasize that p is much smaller

than the honest hashing power, implying that $f = npq < 1$), the current post-quantum honest majority condition can be expressed as $Q \lessapprox n \cdot q \cdot p^{1/2} \cdot O(1)$, meaning that each quantum query is worth $p^{-1/2}$ classical queries. Second, the number of rounds for safe settlement is (up to a constant) the same as in the classical case.

1.3 Related Work

A first step towards understanding Bitcoin's vulnerabilities against quantum attacks was taken by Aggarwal *et al.* [35]. They pointed out the imminent break of the elliptic-curve-based signature scheme in Bitcoin, and argued that in contrast the PoW is relatively resistant to near-term quantum computations due to their slow clock speed and large overhead of quantum error correction. The Bitcoin protocol in a setting where honest parties are also quantum was considered in [36, 37]. While these papers offer interesting observations, they fall short of formal security guarantees. Applications of smart contracts in the quantum world are explored in [38], where the authors propose a classical-quantum payment system which can scale better than blockchains by relying on quantum money techniques.

A number of proof techniques have been developed for the QRO over the years. For example, one can simulate a quantum random oracle [39, 40, 41], program it under a variety of circumstances [42, 43], establish generic security of hash functions [44, 45, 46, 22]. These techniques enable proving the quantum security of many cryptographic schemes in the QRO model [47, 48, 49, 50, 51]. Zhandry's recording technique has inspired many follow-up with various improvements and new applications (e.g., [52, 53, 54, 55]).

A related but different problem than Chain-of-PoWs, called proofs of sequential work, was analyzed in [56, 57]. A central task there is to find a q-chain, which is a sequence $(x_0, x_1, ..., x_q)$ such that $x_i = H(x_{i-1})$ for $1 \leq i \leq q$, using fewer than q queries. This strict relation between the number of quantum queries and the size of the solution marks a drastic distinction from our search problem in the Bitcoin setting.

An interesting generalization of the recording technique to a parallel query model is investigated in the two papers above [56, 57] as well as in [58]. We want to point out that this generalization however would not be sufficient in the Bitcoin context in its current form. One immediate challenge is that the parties can act in an adaptive and heterogeneous way, and one needs to take into account classical communication between parallel parties.

As a final note, in a previous version of this work [59], we analyzed a restricted family of quantum-attack strategies. In the current version, we fully resolve this limitation using completely different techniques. A more thorough discussion is deferred to Appendix A.

2 Preliminaries

2.1 The Recording Oracle Technique

The recording oracle technique of Zhandry [18] is a powerful tool that allows keeping track of a quantum adversary's knowledge when interacting with a *uniform* random oracle.

Assume that the underlying uniformly sampled function of the quantum random oracle is $f : \{0,1\}^m \to \{0,1\}$. This is typically modelled as queries to an oracle performing the map: $|x\rangle |y\rangle \to |x\rangle |y \oplus f(x)\rangle$.
Alternatively, this is equivalent to queries to a "phase" oracle performing the map: $|x\rangle |y\rangle \to (-1)^{y \cdot f(x)} |x\rangle |y\rangle$.

The starting point is the following essential observation: an adversary querying the quantum oracle cannot distinguish between the following two settings. In the first setting the function is

fixed and sampled uniformly at random. In the second setting, there exists an extra function register, prepared in uniform superposition over all possible functions $\frac{1}{\sqrt{2^{2^m}}}\sum_f |f\rangle$, and all further actions/operations are conditional on this register[4]. The function register is not accessible to the adversary and, mathematically, we can trace-it out, resulting exactly to a uniform (classical) mixture over all possible functions.

Using this observation, we will consider that when the adversary performs a query $\sum_{x,y} a_{x,y}|x,y\rangle$, the entire state of the system before the query (composed of adversary's and oracle's state) is of the form: $\sum_{x,y} a_{x,y}|x,y\rangle \otimes \frac{1}{\sqrt{2^{2^m}}}\sum_f |f\rangle$. Then the state of the system after the query (using the phase oracle) becomes: $|\phi_1\rangle := \sum_{x,y} a_{x,y}|x,y\rangle \otimes \frac{1}{\sqrt{2^{2^m}}}\sum_f |f\rangle (-1)^{y\cdot f(x)}$. This shows that the phase resulted from the query can equally be viewed as affecting the oracle's state.

Specifically, we will next analyze how each query affects the oracle's state. We first denote the 2^m-bit string $P_{(x,y)}$ as the string having value y on position x and 0 everywhere else. Using this notation, we can express the state after the query: $|\phi_1\rangle := \sum_{x,y} a_{x,y}|x,y\rangle \otimes \frac{1}{\sqrt{2^{2^m}}}\sum_f |f\rangle (-1)^{\langle f, P_{(x,y)}\rangle}$, where $\langle f, P_{(x,y)}\rangle$ denotes the inner product of the two strings.
This can also be generalized to obtain the state of the system after t queries is of the form:
$|\phi_t\rangle := \sum_{x_1,\ldots,x_t,y_1,\ldots,y_t} a_{x_1,\ldots,x_t,y_1,\ldots,y_t}|\psi_{x_1,\ldots,x_t,y_1,\ldots,y_t}\rangle \otimes \frac{1}{\sqrt{2^{2^m}}}\sum_f |f\rangle (-1)^{\langle f,(P_{(x_1,y_1)}\oplus\ldots\oplus P_{(x_t,y_t)})\rangle}$.
But now, by applying the Hadamard operation to all the qubits of the final register (oracle's state), the oracle state would become: $\left|P_{(x_1,y_1)} \oplus \ldots \oplus P_{(x_t,y_t)}\right\rangle$.
Consequently, we notice that in the Fourier domain the evolution of the adversary's queries would be the following: We start with the all-zero quantum state in the oracle register; namely, the state of the oracle is denoted with $D = 0^{2^m}$. Then, with each adversarial query we will XOR $P_{(x,y)}$ to the string D. Finally, we notice that as a result, after t queries the oracle's state D will have at most t bits equal to 1.

3 Query Complexity of k-BerSearch

In this section we prove the quantum query complexity of the *randomized* search problem denoted as k-BerSearch. This generalizes existing direct product theorems (cf. [30, 24, 25, 19]) to an average-case setting.

Problem k-BerSearch: k solutions randomized search

Given: $f : \{0,1\}^m \to \{0,1\}$ generated by Samp: for each $x \in \{0,1\}^m$, *independently* set

$$f(x) = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{otherwise} \end{cases}$$

Goal: Find $x_1, \ldots, x_k$ such that $f(x_i) = 1$ for all $i \in [k]$.

Theorem 3.1. *For any quantum adversary making at most N queries, the probability p_k^N of solving k-BerSearch satisfies:*

$$p_k^N \le 4(1-p)p^k \left(\sum_{i=0}^{k} (\sqrt{1-p})^i \cdot \binom{N}{i}\right)^2 \le \frac{1}{\pi k} \cdot \left(e^2 (N/k)^2 p\right)^k .$$

[4]Here, we abuse the notation and denote by f the 2^m-bit string representing the truth table of the function f.

The expression following the second inequality comes from some elementary algebraic simplification, whose proof can be found in Appendix B. In the following sections we prove the bound in the first expression. We first develop a framework of the recording technique for functions drawn according to a point-wise independent Bernoulli distribution, and then apply it to k-BerSearch.

3.1 Recording Technique Framework for Bernoulli Distributions

At a high level, we proceed in the following steps:

1. Extend the primal-dual framework to functions sampled according to Bernoulli distributions:
    1. First define a proper dual domain to accommodate for functions sampled according to this distribution;
    2. Determine the mapping between the primal and dual domains;
    3. Show how each quantum query affects the dual domain;
    4. Finally, map back to the standard domain and analyse what is the form of the system state in the standard domain;
2. For the specific k-BerSearch problem, relate the probability of finding a solution to the structure of the final state in the standard domain:
    1. Relate solving k-BerSearch to causing the desired property in the oracle (i.e., at least k non-zero entries in the visited positions).
    2. Bound the probability of the occurrence of desired property in the oracle for any N-query algorithm.

We recall from Section 2.1 that the key observation in the uniform setting is to keep the random function coherent, i.e., a uniform superposition of the truth table of all possible functions $\sum_{f\in\{0,1\}^{2^m}} |f\rangle$. Then if we look at it in the Fourier domain, this superposition becomes simply $|0^{2^m}\rangle$. In this viewpoint, a query $|x, y\rangle$ to the oracle has the effect that it modifies (in superposition) one position x of the truth table. Along the course of the query algorithm, the truth table gets updated and an algorithm's knowledge of the oracle translates to combinatorial properties of the truth table.

In our case, we extend this primal-dual framework to a non-uniform distribution. Loosely speaking, we will see that solving k-BerSearch amounts to that at least k of the entries in the truth table being rotated towards $|1\rangle$.

Definition 3.2 (Bernoulli random function and Bernoulli superposition)**.** *We call $f : \{0,1\}^m \to \{0,1\}$ a Bernoulli random function, if for every input x, $f(x)$ is sampled according to a Bernoulli distribution. We will denote by α_f the probability of f being chosen according to our Bernoulli sampling procedure. For each f, we denote its truth table also by f, which is a $M := 2^m$ bit string, where the x'th bit corresponds to the function value $f(x)$. We can hence prepare a coherent oracle state $\sum_f \sqrt{\alpha_f}\, |f\rangle_F$, also called a Bernoulli superposition, in a function register F, which we call the standard Bernoulli oracle (StdBO). Clearly, measuring it in the standard basis would sample a function f according to our Bernoulli distribution.*

Recall in the standard query model, an oracle query is modelled as: $|x, y\rangle \mapsto (-1)^{y\cdot f(x)} |x\rangle |y\rangle$. The standard Bernoulli oracle action can be described as follows:

Definition 3.3 (Standard Bernoulli Oracle Query)**.** *An oracle query in StdBO is modelled as:*

$$\mathsf{StdBO} : |x, y\rangle \otimes \sum_f \sqrt{\alpha_f}\, |f\rangle_F \mapsto |x, y\rangle \otimes \sum_f \sqrt{\alpha_f}(-1)^{y\cdot f(x)} |f\rangle_F \tag{3.1}$$

From the perspective of any quantum query algorithm A, it is indistinguishable whether it accesses StdBO or a sampled Bernoulli random function f. Namely:

$$\Pr\left[A^f(\cdot) = 1 : f \leftarrow \mathsf{Samp}\right] = \Pr\left[A^{\mathsf{StdBO}}(\cdot) = 1\right].$$

We describe a map which will allow us to switch between the standard and dual domains. Note that in the special case of a uniform random function, it becomes the Hadamard gate ($p = 1/2$).

Definition 3.4 (Map Standard-Dual domains)**.** *For general p, we consider the following unitary U_p acting on a single qubit:*

$$U_p : |b\rangle \mapsto \sqrt{1-p}\,|b\rangle + (-1)^b\sqrt{p}\,|b \oplus 1\rangle \tag{3.2}$$

Then let $\mathcal{U}_p := \otimes_{x\in\{0,1\}^m} U_p^x$ where $U_p^x = U_p$ for all x. We observe that:

$$\mathcal{U}_p \left|0^M\right\rangle_F = \sum_f \sqrt{\alpha_f}\,|f\rangle_F\ . \tag{3.3}$$

We can now examine the knowledge of an algorithm about the underlying oracle. Consider running any query algorithm A with respect to the standard domain. Intuitively, by our earlier observation, at the end of the execution, the function register will contain superposition of strings D of bounded Hamming weight, representing modified truth tables in the dual domain. The non-zero positions (in these strings D) would entail what the query algorithm can infer about the oracle. In particular, once we convert the non-zero entries back to the primal domain through the unitary U_p, i.e., assigning Bernoulli function values, the likelihood of getting k or more 1s essentially bounds the probability of successfully solving k-BerSearch.

Definition 3.5 (System state)**.** *For any query algorithm A, the state of the entire system in the standard domain, can be described by:*

$$|\psi\rangle = \sum_{\mathbf{x},\mathbf{y},z,D} \alpha_{\mathbf{x},\mathbf{y},z,D}\,|\mathbf{x},\mathbf{y},z\rangle \otimes |D\rangle\ , \tag{3.4}$$

where $\mathbf{x}$ and $\mathbf{y}$ are tuples consisting of k inputs and outputs respectively and z represents the "workspace" register of the algorithm.

Intuitively, the tuple $\mathbf{x}$ represents the solution (of size k) the adversary outputs for the k-BerSearch problem and these k inputs may or may not be queried. In this way, after each query, if we were to measure the first registers $\mathbf{x}$ and $\mathbf{y}$, we can determine the success probability that the tuple represents a solution for our search problem. We will also decompose $|D\rangle$ as:

$$|D\rangle = |D_{\mathbf{x}}\rangle \otimes |D_{\bar{\mathbf{x}}}\rangle\ ,$$

where $D_{\mathbf{x}}$ contains the entries that coincide with $\mathbf{x}$, and $D_{\bar{\mathbf{x}}}$ contains the oracle's state on the rest of the inputs.

Using the definitions of the standard-dual map and of the system state, we can now describe how the system evolves after each query in the dual domain:

Definition 3.6 (Query Operator in the Dual Domain)**.** *For the dual domain we define the following query operator:*

$$\tilde{O} := (I \otimes \mathcal{U}_p^\dagger) \cdot \mathsf{StdBO} \cdot (I \otimes \mathcal{U}_p).$$

where $\mathcal{U}_p^\dagger$ and $\mathcal{U}_p$ only act on the last register (containing the strings D).

Standard-Dual domains view. We are now ready to fully describe the Standard-Dual view of the recording technique for Bernoulli functions:

- Standard domain:
  1. The oracle query is the standard function oracle StdBO;
  2. The initial state of the oracle is the Bernoulli superposition $\sum_f \sqrt{\alpha_f}\,|f\rangle_F$;
  3. We denote the state of the entire system in the standard domain after t queries with $|\psi^t\rangle$.
- Dual domain:
  1. The oracle query is $\tilde{O} := (I \otimes \mathcal{U}_p^\dagger) \cdot \mathsf{StdBO} \cdot (I \otimes \mathcal{U}_p)$;
  2. The initial state of the oracle is the state $\left|0^M\right\rangle_F$;
  3. Denote the state of the entire system in the dual domain after t queries with $|\phi^t\rangle$.

It can easily be seen that the two domains are equivalent up to a final $\mathcal{U}_p$ unitary applied on the state of the system corresponding to the dual domain.

Corollary 3.7 (Relation between states in standard-dual domains)**.** *After any t queries, we have the relation:*

$$\left|\psi^t\right\rangle = I \otimes \mathcal{U}_p \left|\phi^t\right\rangle \quad \text{where } \mathcal{U}_p \text{ only acts on the last function register} \tag{3.5}$$

Evolution of the query algorithm in Dual domain. We keep track of the adversary's knowledge in the dual domain, where for the evolution of the system state, we notice the following:

- After each query, only one position from the function register of each of the states in the superposition of the system state is modified, by being rotated using the unitary U_p, the rest of the positions remain unchanged.
  Therefore, after N queries there are at most N positions containing rotated qubits and the rest of them remain the $|0\rangle$ state.
- However, after all N queries, we need to return to the standard domain, by applying a final $\mathcal{U}_p$ on all qubits of the function register and then measure in the computational basis the function register. Note that there is a small chance that the positions that were never queried (corresponding to $|0\rangle$ state) will also collapse to 1.

Once we have established this framework for the recording technique for Bernoulli random functions, we will show how to apply it for our target problem k-BerSearch.

3.2 Analysis of k-BerSearch via the Bernoulli Recording Technique

To determine the success probability of the adversary to solve the k-BerSearch problem using N quantum queries, denoted as p_k^N, we use the oracle's state as follows:

1. We look at the vector $\mathbf{x}$ and at the measurement outcome of the function register (previously described), let us call this measurement $\tilde{f}$.
2. Then, we check if $\tilde{f}(x_i) = 1$ for all $x_i \in \mathbf{x}$.

Importantly, the probability of success is defined with respect to the standard domain, hence after the final $\mathcal{U}_p$ application.

Relation between success probability and oracle state. In order to relate the success probability p_k^N with the evolution of the oracle we need to define the following set of projectors. Firstly, we introduce two families of projectors P and Π, acting on the states of the standard domain (Eq. 3.4):

Definition 3.8 (Projection Family P)**.** *For any integer k, consider the following family of projectors P acting in the standard domain:*

- P_k*: defined by all basis states $|\mathbf{x}, \mathbf{y}, z\rangle |D\rangle$ such that D contains exactly k ones;*
- $P_{\leq k}$*: defined by all basis states $|\mathbf{x}, \mathbf{y}, z\rangle |D\rangle$ such that D contains at most k ones, $P_{\leq k} = \sum_{i=0}^{k} P_i$;*
- $P_{\geq k}$*: defined by all basis states $|\mathbf{x}, \mathbf{y}, z\rangle |D\rangle$ such that D contains at least k ones, $P_{\geq k} = \sum_{i \geq k} P_i$.*

In addition, related to P_k, we will also define the projectors:

- P_k^0 *defined by the basis states $|\mathbf{x}, \mathbf{y}, z\rangle |D\rangle$ such that D contains exactly k ones, $y = 1$ and $f(x) = 0$. For this, we can consider that $|x, y\rangle$ is an extra register containing a single input x and a single bit y (different from $|\mathbf{x}, \mathbf{y}\rangle$) which is used for the queries to the function oracle* StdBO*.*
- P_k^1 *defined by the basis states $|\mathbf{x}, \mathbf{y}, z\rangle |D\rangle$ such that D contains exactly k ones, $y = 1$ and $f(x) = 1$.*

Definition 3.9 (Projection Π)**.** *The projector Π is acting in the standard domain and is defined by the basis states $|\mathbf{x}, \mathbf{y}, z, D_{\mathbf{x}}, D_{\bar{\mathbf{x}}}\rangle$ such that $D_{\mathbf{x}}$ has hamming weight k.*

Finally, we introduce the family of projectors Ξ_i, which will act on the dual domain (hence before the final application of $\mathcal{U}_p$):

Definition 3.10 (Projection family Ξ)**.** *The projectors $\{\Xi_i\}_i$ are acting in the dual domain and are defined by the basis states having exactly i ones qubits in $D_{\mathbf{x}}$.*

Now, we will see how to employ all these projectors in order to bound the success probability of the adversary to solve the k-BerSearch problem when having N available quantum queries.

For the success event of finding k preimages of 1, we must have k ones in $D_{\mathbf{x}}$ for $\left|\psi^N\right\rangle$ - the state in the standard domain. In other words, the adversary's success probability will be bounded by:

$$p_k^N \leq \left\|\Pi \left|\psi^N\right\rangle\right\|^2 = \left\|\Pi \left(\mathcal{U}_p \left|\phi^N\right\rangle\right)\right\|^2. \tag{3.6}$$

Deriving the bound on success probability. Next, we will determine the bound on p_k^N by following the next two steps:

1. Derive a bound on the norm of the projection Π using the norms of the projection Ξ_i and subsequently bound this norm with the norms of $P_{\geq i}$.
2. Bound the norm of $P_{\geq i}$. This will determine the progress after N queries as:

$$a_{N,k} := \left\|P_{\geq k} \left|\phi^N\right\rangle\right\| \tag{3.7}$$

For the first step we show the following result:

Lemma 3.11. *The norm of projection Π can be bounded using the progress measure after N queries as follows:*

$$\left\|\Pi\left|\psi^N\right\rangle\right\| \leq \sum_{i=0}^{k}(\sqrt{p})^{k-i}(\sqrt{1-p})^i\left\|\Xi_i\left|\phi^N\right\rangle\right\| \leq \sum_{i=0}^{k}(\sqrt{p})^{k-i}(\sqrt{1-p})^i\left\|P_{\geq i}\left|\phi^N\right\rangle\right\| \tag{3.8}$$

For the second step we will show that:

Lemma 3.12. *For any N and any k, given the state $\left|\phi^N\right\rangle$ of the dual domain after N queries, we have the following bound:*

$$a_{N,k} := \left\|P_{\geq k}\left|\phi^N\right\rangle\right\| \leq 2\sqrt{1-p}(\sqrt{p})^k\binom{N}{k}. \tag{3.9}$$

In the following sections we will show how to derive these two results and finally we will combine them to obtain the final bound stated in Theorem 3.1.

3.3 Bounding the Success Probability with Progress Measure

We now show how to bound the success probability using the progress measure, as stated in Lemma 3.11.

Proof of Lemma 3.11. Firstly, using the definition of $\left|\psi^N\right\rangle$ we have:

$$\left\|\Pi\left|\psi^N\right\rangle\right\| = \left\|\Pi\mathcal{U}_p\left|\phi^N\right\rangle\right\| \tag{3.10}$$

Observe that $\sum_{i=0}^{k}\Xi_i = I$, hence we obtain:

$$\left\|\Pi\left|\psi^N\right\rangle\right\| = \left\|\Pi\left(\mathcal{U}_p\left(\sum_{i=0}^{k}\Xi_i\right)\left(\left|\phi^N\right\rangle\right)\right)\right\| \leq \sum_{i=0}^{k}\left\|\Pi\left(\mathcal{U}_p\Xi_i\left(\left|\phi^N\right\rangle\right)\right)\right\| \tag{3.11}$$

Hence to complete the proof, we need to show that:

$$\left\|\Pi\left(\mathcal{U}_p\Xi_i\left(\left|\phi^N\right\rangle\right)\right)\right\| \leq (\sqrt{1-p})^i(\sqrt{p})^{k-i}\left\|\Xi_i\left|\phi^N\right\rangle\right\| \tag{3.12}$$

Now let us consider separately the state $|\rho_i\rangle := \Xi_i\left(\left|\phi^N\right\rangle\right)$ for any quantum state $\left|\phi^N\right\rangle$. Then, from the definition of Ξ_i, $|\rho_i\rangle$ has exactly i ones and $k-i$ zeros in the register $D_{\mathbf{x}}$. Let us denote the set of allowed $D_{\mathbf{x}}$ (having exactly i ones) by $\mathcal{D}_{\mathbf{x}}$, where $|\mathcal{D}_{\mathbf{x}}| = \binom{k}{i}$; and denote the set of possible configurations of $D_{\bar{\mathbf{x}}}$ by $\mathcal{D}_{\bar{\mathbf{x}}}$, where $|\mathcal{D}_{\bar{\mathbf{x}}}| = 2^{M-k}$. Then we can write:

$$|\rho_i\rangle = \sum_{\substack{\mathbf{x},\mathbf{y},z \\ D_{\mathbf{x}}\in\mathcal{D}_{\mathbf{x}}, D_{\bar{\mathbf{x}}}\in\mathcal{D}_{\bar{\mathbf{x}}}}} \beta_{\mathbf{x},\mathbf{y},z,D_{\mathbf{x}},D_{\bar{\mathbf{x}}}}\,|\mathbf{x}\rangle\,|\mathbf{y}\rangle\,|z\rangle\,|D_{\mathbf{x}}\rangle\otimes|D_{\bar{\mathbf{x}}}\rangle\,. \tag{3.13}$$

For simplicity we drop the $|\mathbf{y}\rangle\,|z\rangle$ register as they will not be affected, and instead denote $|\rho\rangle$ as:

$$|\rho_i\rangle = \sum_{\substack{\mathbf{x},\mathbf{y},z \\ D_{\mathbf{x}}\in\mathcal{D}_{\mathbf{x}}, D_{\bar{\mathbf{x}}}\in\mathcal{D}_{\bar{\mathbf{x}}}}} \beta_{\mathbf{x},\mathbf{y},z,D_{\mathbf{x}},D_{\bar{\mathbf{x}}}}\,|\mathbf{x}\rangle\,|D_{\mathbf{x}}\rangle\otimes|D_{\bar{\mathbf{x}}}\rangle\,. \tag{3.14}$$

Now let us consider applying Π to a single state from this superposition, namely $|\mathbf{x}\rangle\,|D_{\mathbf{x}}\rangle\otimes|D_{\bar{\mathbf{x}}}\rangle$:

$$\|\Pi\,|\mathbf{x}\rangle\,(\mathcal{U}_p\,|D_{\mathbf{x}}\rangle\otimes|D_{\bar{\mathbf{x}}}\rangle)\| \quad = \quad ||\Pi\,|\mathbf{x}\rangle\,(\sqrt{p}\,|0\rangle - \sqrt{1-p}\,|1\rangle)^{\otimes i}\otimes(\sqrt{1-p}\,|0\rangle +$$

$$
\begin{aligned}
&+ \quad \sqrt{p}\,|1\rangle)^{\otimes k-i} \otimes U_p^{\otimes M-k}\,|D_{\bar{\mathbf{x}}}\rangle\,|| \\
&= \quad (\sqrt{1-p})^i(\sqrt{p})^{k-i}\left\||\mathbf{x}\rangle \otimes |1_{D_{\mathbf{x}}}\rangle \otimes U_p^{\otimes M-k}\,|D_{\bar{\mathbf{x}}}\rangle\right\| \\
&= \quad (\sqrt{1-p})^i(\sqrt{p})^{k-i} \qquad (3.15)
\end{aligned}
$$

Therefore, we have that:

$$
\|\Pi\mathcal{U}_p\,|\rho_i\rangle\| = (\sqrt{1-p})^i(\sqrt{p})^{k-i}\left\|\sum_{\substack{\mathbf{x},\mathbf{y},z\\ D_{\mathbf{x}}\in\mathcal{D}_{\mathbf{x}},D_{\bar{\mathbf{x}}}\in\mathcal{D}_{\bar{\mathbf{x}}}}} \beta_{\mathbf{x},\mathbf{y},z,D_{\mathbf{x}},D_{\bar{\mathbf{x}}}}\,|\mathbf{x}\rangle \otimes |1_{D_{\mathbf{x}}}\rangle \otimes \left(U_p^{\otimes M-k}\,|D_{\bar{\mathbf{x}}}\rangle\right)\right\| \tag{3.16}
$$

$$
= (\sqrt{1-p})^i(\sqrt{p})^{k-i}\cdot\left\|\sum_{\substack{\mathbf{x},\mathbf{y},z\\ D_{\mathbf{x}}\in\mathcal{D}_{\mathbf{x}},D_{\bar{\mathbf{x}}}\in\mathcal{D}_{\bar{\mathbf{x}}}}} \beta_{\mathbf{x},\mathbf{y},z,D_{\mathbf{x}},D_{\bar{\mathbf{x}}}}\,|\mathbf{x}\rangle \otimes |1_{D_{\mathbf{x}}}\rangle \otimes |D_{\bar{\mathbf{x}}}\rangle\right\| \tag{3.17}
$$

$$
= (\sqrt{1-p})^i(\sqrt{p})^{k-i}\cdot\sqrt{\sum_{\substack{\mathbf{x},\mathbf{y},z\\ D_{\mathbf{x}}\in\mathcal{D}_{\mathbf{x}},D_{\bar{\mathbf{x}}}\in\mathcal{D}_{\bar{\mathbf{x}}}}} |\beta_{\mathbf{x},\mathbf{y},z,D_{\mathbf{x}},D_{\bar{\mathbf{x}}}}|^2} \tag{3.18}
$$

$$
= (\sqrt{1-p})^i(\sqrt{p})^{k-i}\cdot\||\rho_i\rangle\| \tag{3.19}
$$

□

3.4 Bounding the Progress Measure

We now show how to bound the progress measure $a_{N,k} := \left\|P_{\geq k}\left|\phi^N\right\rangle\right\|$, as stated in Lemma 3.12.

Proof of Lemma 3.12. First, we want to define a recurring relation for the progress measure sequence. Specifically, for any t, we have:

$$
\begin{aligned}
a_{t+1,k} &= \left\|P_{\geq k}\tilde{O}\left|\phi^t\right\rangle\right\| = \left\|P_{\geq k}\tilde{O}(P_{\geq k}+P_{k-1}+P_{\leq k-2})\left|\phi^t\right\rangle\right\| \\
&\leq \left\|P_{\geq k}\tilde{O}P_{\geq k}\left|\phi^t\right\rangle\right\| + \left\|P_{\geq k}\tilde{O}P_{k-1}\left|\phi^t\right\rangle\right\| \\
&\leq a_{t,k} + \left\|P_{\geq k}\tilde{O}P_{k-1}\left|\phi^t\right\rangle\right\|
\end{aligned} \tag{3.20}
$$

Now let us consider separately the term $\left\|P_{\geq k}\tilde{O}P_{k-1}\,|\phi^t\rangle\right\|$. A basis state $|\mathbf{x},\mathbf{y},z\rangle\,|D\rangle$ contributes to $a_{t+1,k}$ if D has exactly $k-1$ ones, but also if $y=1$ and $f(x)\neq 1$.

What would actually contribute to $a_{t+1,k}$ is the state $P_{\geq k}\tilde{O}P^0_{k-1}\,|\phi^t\rangle$, hence we have:

$$
a_{t+1,k} \leq a_{t,k} + \left\|P_{\geq k}\tilde{O}P^0_{k-1}\left|\phi^t\right\rangle\right\| \tag{3.21}
$$

Now, to bound the second term, for any basis state $|\mathbf{x},\mathbf{y},z\rangle\,|D\rangle$ in the support of P^0_{k-1}, we will make use of the following observation.

Using the definition of $\tilde{O}$, we notice that each query $|x,y\rangle$ in the dual domain has the following effect on the zero entries of the D strings (oracle state) - when $f(x)=0$:

$$
\begin{aligned}
U_p^\dagger\cdot\mathsf{StdBO}\cdot U_p\,|0\rangle &= U_p^\dagger[(-1)^{y\cdot 0}\sqrt{1-p}\,|0\rangle + (-1)^{y\cdot 1}\sqrt{p}\,|1\rangle] = U_p^{\dagger 2y}\,|0\rangle \\
&= \begin{cases} |0\rangle\,, & \text{if } y=0 \\ (1-2p)\,|0\rangle - 2\sqrt{p(1-p)}\,|1\rangle\,, & \text{if } y=1 \end{cases}
\end{aligned} \tag{3.22}
$$

This leads us to:

$$P_{\geq k}\tilde{O}\left|\mathbf{x},\mathbf{y},z\right\rangle \otimes \left|D\right\rangle = -2\sqrt{p(1-p)}\left|\mathbf{x},\mathbf{y},z\right\rangle \otimes D_{\mathbf{x}-\{x\}} \otimes \left|1\right\rangle_x \otimes \left|D_{\bar{\mathbf{x}}}\right\rangle) \tag{3.23}$$

where (x, y) is the query to the function oracle StdBO, as in Definition 3.8.
As a result, we have:

$$\left\|P_{\geq k}\tilde{O}P^0_{k-1}\left|\phi^t\right\rangle\right\| = 2\sqrt{p(1-p)}||P^0_{k-1}\left|\phi^t\right\rangle || \tag{3.24}$$

Therefore, we obtain:

$$a_{t+1,k} \leq a_{t,k} + 2\sqrt{p(1-p)}||P^0_{k-1}\left|\phi^t\right\rangle || \tag{3.25}$$

Applying this relation iteratively for any $t \in \{N-1, ..., 0\}$, leads us to the following bound on the progress measure after N queries:

$$a_{N,k} \leq 2\sqrt{p(1-p)}\sum_{i=0}^{N-1}\left\|P^0_{k-1}\left|\phi^i\right\rangle\right\| \tag{3.26}$$

By using Lemma 3.13, we obtain the bound on $a_{N,k}$:

$$a_{N,k} \leq 2\sqrt{1-p}(\sqrt{p})^k\binom{N}{k}. \tag{3.27}$$

□

Lemma 3.13. *Let $\left|\phi^i\right\rangle$ be the state before the $i+1$'th query and $\beta^i_{k-1} := \left\|P^0_{k-1}\left|\phi^i\right\rangle\right\|$. Then $S_{k,N} := \sum_{i=0}^{N-1}\beta^i_{k-1} \leq \binom{N}{k}(\sqrt{p})^{k-1}$.*

Proof. We analyze β^i_{k-1} case by case:

- If $i < k-1$, then $\beta^i_{k-1} = 0$. This is because unless at least $k-1$ queries have been made, the number of non-zero entries in D is below $k-1$ and falls out of the support of P^0_{k-1}.
- If $i = k-1$, then $\beta^i_{k-1} \leq (\sqrt{p})^{k-1}$. By applying $k-1$ queries the magnitude of all resulting in $|1\rangle$ is $\sqrt{p}^{k-1}$.
- If $i \geq k$, then $\beta^i_{k-1} \leq \binom{i}{k-1}(\sqrt{p})^{k-1}(\sqrt{1-p})^{i-k+1}$. This follows from a combinatorial argument, reminiscent of binomial coefficients. Think of taking a random walk on a line for i steps, we need to move forward $k-1$ times. There are $\binom{i}{k-1}$ "routes", and each route contributes a weight of at most $(\sqrt{p})^{k-1}(\sqrt{1-p})^{i-k+1}$.

Hence we have:

$$S_{k,N} := \sum_{i=0}^{N-1}\beta^i_{k-1} \leq (\sqrt{p})^{k-1}\sum_{i=k-1}^{N-1}\binom{i}{k-1} = (\sqrt{p})^{k-1}\binom{N}{k}. \tag{3.28}$$

where for the last equality we used the identity: $\sum_{n=t}^{m}\binom{n}{t} = \binom{m+1}{t+1}$. □

3.5 Putting Everything Together

We now can combine the results to obtain the bound on p_k^N:

Theorem 3.14. *The success probability p_k^N of solving k-BerSearch satisfies*

$$p_k^N \leq 4(1-p)p^k \left(\sum_{i=0}^{k} (\sqrt{1-p})^i \cdot \binom{N}{i} \right)^2 . \tag{3.29}$$

Proof. First by Lemma 3.11 we have

$$p_k^N \leq \left\| \Pi \left| \psi^N \right\rangle \right\|^2 \leq \sum_{i=0}^{k} (\sqrt{p})^{k-i} (\sqrt{1-p})^i \left\| P_{\geq i} \left| \phi^N \right\rangle \right\| . \tag{3.30}$$

We know from Lemma 3.12 that

$$a_{N,i} := \left\| P_{\geq i} \left| \phi^N \right\rangle \right\| \leq 2\sqrt{1-p} (\sqrt{p})^i \binom{N}{i} \tag{3.31}$$

Hence we have:

$$p_k^N \leq \left(\sum_{i=0}^{k} (\sqrt{p})^{k-i} (\sqrt{1-p})^i a_{N,i} \right)^2 \leq 4(1-p)p^k \left(\sum_{i=0}^{k} (\sqrt{1-p})^i \cdot \binom{N}{i} \right)^2 . \tag{3.32}$$

□

4 The Bitcoin Backbone against Quantum Adversaries

4.1 The Bitcoin Backbone Protocol: Model and Definitions

We will analyze our post-quantum version of the Bitcoin backbone protocol, where the honest parties are classical but the adversary is not, in the network model considered in [3], namely, a synchronous communication network which is based on Canetti's formulation of "real world" execution for multi-party cryptographic protocols [60, 61]).

In such a network model, the protocol execution proceeds in *rounds*; in each round, parties receive messages from the network through their communication interface (e.g., input tape or register), perform some computation, and send messages to the network, which are delivered at the beginning of the next round. The inputs to the computation performed by the parties are provided by an environment program denoted by $\mathcal{Z}$. For further details about the synchronous network model, refer to [60].

The execution is assumed to have a polynomial (in the security parameter) time bound. The actual message delivery is provided by a "diffusion" mechanism that is guaranteed to deliver all messages, without however preserving their order and allowing the adversary to arbitrarily inject its own messages. Importantly, the parties are not guaranteed to have the same view of the messages delivered in each round, except for the fact that all the messages sent by the honest parties in the previous round are delivered. Furthermore, and in line with the cryptographic protocols literature, we assume the presence of a single adversary (which may in turn "corrupt" many parties), albeit equipped with quantum computing power. The adversary is allowed to "spoof" messages by changing the source information in them (i.e., communication is not authenticated).

The Bitcoin backbone protocol. First, we introduce some blockchain notation, following [3]. A *block* is any triple of the form $B = \langle s, x, ctr \rangle$ where $s \in \{0,1\}^\kappa, x \in \{0,1\}^*, ctr \in \mathbb{N}$ are such that satisfy predicate $\mathsf{validblock}_q^D(B)$ defined as:

$$(H(ctr, G(s,x)) < T) \wedge (ctr \leq q), \tag{4.1}$$

where H, G are cryptographic hash functions (e.g., SHA-256) modelled as random oracles. The parameter $T \in \mathbb{N}$ is also called the block's *difficulty level*. We then define $p = T/2^\kappa$ to be the probability that a single classical query solves a PoW. The parameter $q \in \mathbb{N}$ is a bound that in the Bitcoin implementation determines the size of the register ctr; in our treatment we allow this to be arbitrary, and use it to denote the maximum allowed number of hash queries performed by the (classical) parties in a round.

A *blockchain*, or simply a *chain* is a sequence of *blocks*. The rightmost block is the *head* of the chain, denoted $\mathrm{head}(\mathcal{C})$. Note that the empty string ε is also a chain; by convention we set $\mathrm{head}(\varepsilon) = \varepsilon$. A chain $\mathcal{C}$ with $\mathrm{head}(\mathcal{C}) = \langle s', x', ctr' \rangle$ can be extended to a longer chain by appending a valid block $B = \langle s, x, ctr \rangle$ that satisfies $s = H(ctr', G(s', x'))$. In case $\mathcal{C} = \varepsilon$, by convention any valid block of the form $\langle s, x, ctr \rangle$ may extend it. In either case we have an extended chain $\mathcal{C}_{\mathsf{new}} = \mathcal{C}B$ that satisfies $\mathrm{head}(\mathcal{C}_{\mathsf{new}}) = B$. Consider a chain $\mathcal{C}$ of length m (written as $len(\mathcal{C}) = m$) and any non-negative integer k. We denote by $\mathcal{C}^{\lceil k}$ the chain resulting from the "pruning" of the k rightmost blocks. Note that for $k \geq len(\mathcal{C})$, $\mathcal{C}^{\lceil k} = \varepsilon$. If $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$ we write $\mathcal{C}_1 \preceq \mathcal{C}_2$.

The Bitcoin backbone protocol is executed by an arbitrary number of parties over an unauthenticated network, as described above. It is assumed in [3] that the number of parties running the protocol is fixed, however, parties need not be aware of this number when they execute the protocol. In our analysis we will have n honest parties and a single quantum adversary. Also as mentioned above, communication over the network is achieved by utilizing a send-to-all Diffuse functionality that is available to all parties (and may be abused by the adversary in the sense of delivering different messages to different parties).

Each party maintains a blockchain, as defined above, starting from the empty chain and mining a block that contains the value $s = 0$ (by convention this is the "genesis block"). If in a given round, an honest party is successful in generating a PoW (i.e. satisfying conjunction 4.1), it diffuses it to the network. At each round, each party chooses the longest chain amongst the one he received, and tries to extend it by computing (mining) another block. In such a process, each party's chain may be different, but under certain well-defined conditions, it is shown in [3] that the chains of honest parties will share a large common prefix (see below).

In the backbone protocol, the type of values that parties try to insert in the chain is intentionally left unspecified, as well as the type of chain validation they perform (beyond checking for its structural properties with respect to the hash functions $G(\cdot), H(\cdot)$), and the way they interpret the chain. Instead, these actions are abstracted by the external functions $V(\cdot)$ (the *content validation predicate*), $I(\cdot)$ (the *input contribution function*), and $R(\cdot)$ (the *chain reading function*), which are specified by the application that runs "on top" of the backbone protocol (e.g., a transaction ledger).

Basic security properties of the blockchain. It is shown in [3] that the blockchain data structure built by the Bitcoin backbone protocol satisfies a number of basic properties. At a high level, the first property, called *common prefix*, has to do with the existence, as well as persistence in time, of a common prefix of blocks among the chains of honest parties. The second, called *chain quality*, stipulates the proportion of honest blocks in any portion of some honest party's chain.

Definition 4.1 (Common Prefix)**.** *The* common prefix *property with parameter $k \in \mathbb{N}$, states that for any pair of honest players P_1, P_2 adopting chains $\mathcal{C}_1, \mathcal{C}_2$ at rounds $r_1 \leq r_2$, it holds that $\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$ (the chain resulting from pruning the k rightmost blocks of $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$).*

Definition 4.2 (Chain Quality)**.** *The* chain quality *property with parameters $\mu \in \mathbb{R}$ and $l \in \mathbb{N}$, states that for any honest party P with chain $\mathcal{C}$, it holds that for any l consecutive blocks of $\mathcal{C}$, the ratio of blocks created by honest players is at least μ.*

Parameters and Random Variables. Next, we recall some important notions in the Bitcoin backbone protocol setting.

- n denotes the number of honest parties;
- q represents the number of classical queries of each honest party per round;
- Q denotes the number of adversarial quantum queries per round;
- f is the probability that at least one honest party generates a PoW (i.e. satisfy conjunction 4.1) in a round;
- ϵ will be used for the concentration quality of random variables while κ for the security parameter;
- k denotes the number of blocks for common prefix property and μ denotes the chain quality parameter;
- s refers to the total number of rounds;
- p is the probability of success of generating a PoW (conjunction 4.1) using a single classical query;
- $X(s)$ and $Y(s)$ represent within s rounds, the numbers of rounds at least one honest player and *exactly* one honest player solves a PoW respectively.
- $Z(s)$ represents the number of PoWs solved by an adversary in s consecutive rounds.
- f denotes the probability that at least one honest player generates a PoW (conjunction 4.1) in a single round (e.g., in the Bitcoin system, f is about $2-3\%$).
- In condition (c), an *insertion* denotes the event that given a chain $\mathcal{C}$ with two consecutive blocks B and B', a block B^* created after B' is such that B, B^*, B' form three consecutive blocks of a valid chain.
- A *copy* occurs if the same block exists in two different positions.
- A *prediction* occurs when a block extends one which was computed at a later round.

4.2 The Chain-of-PoWs Problem

In the Bitcoin backbone protocol, an adversary aims to produce a chain of blocks that is longer than the honest chain. We formalize it as the Chain-of-PoWs search problem below.

> Problem Π_G: Chain-of-PoWs
> **Given**: N, $x_0 \in X$ and $h_0, \ldots, h_{N-1}$ as (quantum) random oracles, where each $h_i : X \times Y \to X$ is independently sampled.
> **Goal**: Using N total queries find a sequence $y_0, \ldots, y_{k-1}$ such that $x_{i+1} := h_i(x_i, y_i)$ and $x_{i+1} \leq T \ \forall \ i \in \{0, \cdots, k-1\}$ such that the length of the sequence $k \leq N$ is the maximum that can be achieved. T is a fixed positive integer.

Note that the output of this problem is the maximum length k and the corresponding sequence $(y_0, ..., y_{k-1})$. For ease of notations, we will omit $h_0, \ldots, h_{N-1}$ from the input of Π_G. We call any pair (x, y) s.t. $h(x, y) \leq T$ a PoW.

A possible approach is to prove a composition theorem for the query complexity of such a search problem. This however appears beyond the scope of existing results and techniques in quantum query complexity. Instead, we show its hardness by reducing to the k-BerSearch problem we analyzed before.

Theorem 4.3 (Main Theorem)**.** *For any quantum adversary $\mathcal{A}$ having N quantum queries, the probability that $\mathcal{A}$ solves the Chain-of-PoWs problem, by outputting a solution of size at least k is at most:*

$$P(N,k) \leq \exp\left(-2k \cdot \ln\left(\frac{k}{e(N+k)} \cdot \frac{1}{\sqrt{p}}\right)\right) \tag{4.2}$$

where $p := \frac{T}{2^\kappa}$ is the probability of success of a single query to the random oracle.

4.3 Hardness of Chain-of-PoWs

We relate Chain-of-PoWs to k-BerSearch formally below.

Lemma 4.4. *If for any quantum adversary $\mathcal{A}$ having N quantum queries the probability that $\mathcal{A}$ obtains a solution of size k for k-BerSearch is at most $p_1(N,k)$, then for any quantum adversary $\mathcal{A}'$ having N quantum queries, the probability that $\mathcal{A}'$ obtains a solution of size k for Chain-of-PoWs is at most $p_2(N,k)$, satisfying: $p_2(N,k) \leq p_1(N+k,k)$.*

Proof. We first state k-BerSearch in an equivalent form, which we call Bag-of-PoWs. That is, we consider N functions $h_0, \ldots, h_{N-1}$ as oracles, where each $h_i : X \times Y \to X$ is independently sampled, and the goal is to find a set of pairs $\{(x_{i_1}, y_{i_1})_1, \ldots, (x_{i_k}, y_{i_k})_k\}$ so that $h_{i_l}(x_{i_l}, y_{i_l}) \leq T$, for all $l \in \{1, \ldots, k\}$. This is clearly equivalent to k-BerSearch, by considering the random boolean function $f : [N] \times X \times Y \to \{0,1\}$: $f(i,x,y) = 1$ if and only if $h_i(x,y) \leq T$.

Now suppose for the sake of contradiction that there exists a quantum adversary $\mathcal{A}'$ equipped with N queries such that:

$$p_2 := \Pr[\mathcal{A}'(N) \text{ gives a solution of size } k \text{ for Chain-of-PoWs}] > p_1(N+k,k)\,.$$

We construct an algorithm $\mathcal{A}''$ that using $N+k$ queries obtains a solution of size k for the Bag-of-PoWs problem, and hence a solution also for the k-BerSearch problem, with the same probability p_2.

$\mathcal{A}''$ first samples at random an element x_0 (from the domain of any oracle h_i). Then, it will run $\mathcal{A}'$ on input (N, x_0), in order to obtain the sequence $y_0, \ldots, y_{k-1}$ with probability p_2 spending N queries. Next, starting from y_0, $\mathcal{A}''$ computes $x_i = h_{i-1}(x_{i-1}, y_{i-1})$ for all $i \in \{1, \cdots, k-1\}$ (which is guaranteed to hold from the setting of Chain-of-PoWs) spending one extra query per element of the sequence, i.e. extra k queries. Then the algorithm $\mathcal{A}''$ outputs the pairs $(x_0, y_0), (x_1, y_1), \ldots\ (x_{k-1}, y_{k-1})$ having used $N+k$ queries and will succeed with probability $p_2 > p_1(N+k,k)$.

But from the initial assumption, the probability that $\mathcal{A}''$ using $N+k$ queries obtains a solution of size k for the k-BerSearch is at most $p_1(N+k,k)$.

□

By combining Lemma 4.4 together with the bound on the success probability for the k-BerSearch problem from Theorem 3.1, we can obtain a bound on the success probability for the Chain-of-PoWs problem:

Lemma 4.5. *Any quantum adversary having N queries to the QRO, can obtain a chain of PoWs of length k, with probability at most:*

$$P(N,k) := \frac{2(1-p)}{\pi k}\left[\frac{(N+k)e\sqrt{p}}{k}\right]^{2k} \tag{4.3}$$

where p is the probability of a successful PoW with a single query.

4.4 Post-Quantum Security of the Bitcoin Backbone Protocol

Now that we have established the hardness of the Chain-of-PoWs problem, we will use it to identify the conditions under which the central properties of blockchains, *common prefix* and *chain quality*, can be satisfied in the presence of quantum adversaries. To do so, it is helpful to take a closer look at the classical analysis [3], which proceeds in three steps.

1. A notion of *typical execution* of a blockchain is defined in the presence of honest players and adversaries. This is a critical vehicle that modularizes the analysis.
2. It is proven that under a typical execution, the desired properties will follow for suitable parameters.
3. An *honest majority* condition is identified under which a typical execution occurs with high probability.

We would like to "lift" this analytical framework to the quantum setting with as few changes as possible. However, we run into a roadblock immediately since the definition of a typical execution dose not make sense any more. To see this, let us recall the classical definition.

Definition 4.6 (Classical typical execution [3])**.** *An execution is (ϵ, s)-typical (or just typical), for $\epsilon \in (0,1)$, if for any set S of at least s consecutive rounds, we have:*

(a) $(1-\epsilon)fs < X(s) < (1+\epsilon)fs$ *and* $(1-\epsilon)\mathbb{E}[Y(s)] < Y(s)$.

(b) $Z(s) < \mathbb{E}[Z(s)] + \epsilon\mathbb{E}[X(s)]$.

(c) No insertions, no copies, and no predictions occurred.

For completeness, let us first revisit now four main results of [3] which would help us extract the necessary conditions on the number of adversarial PoWs required to satisfy the security of the backbone protocol.

Lemma 4.7 ([3])**.** *For any $s \geq 2/f$ rounds, we have that with probability $1 - e^{-\Omega(\epsilon^2 sf)}$, the following hold:*

$$(1-\epsilon)fs < X(s) < (1+\epsilon)fs \ ; \ (1-\epsilon)\mathbb{E}[Y(s)] < Y(s) \ ; \ (1-\epsilon)f(1-f)s < Y(s).$$

Lemma 4.8 ([3])**.** *Assuming that $X(s) + Z(s) < 2fs$, then in a typical execution, any $k \geq 2fs$ consecutive blocks of a chain have been computed in more than $k/2f$ consecutive rounds.*

Lemma 4.9 (Common Prefix Lemma ([3]))**.** *Assuming that $Z(s) < Y(s)$ and that at round r of a typical execution an honest party has a chain $\mathcal{C}_1$, while a chain $\mathcal{C}_2$ of length at least $len(\mathcal{C}_1)$ is adopted by an honest party. Then $\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$ and $\mathcal{C}_2^{\lceil k} \preceq \mathcal{C}_1$ for $k \geq 2fs$ and $s \geq \frac{2}{f}$.*

Lemma 4.10 (Chain Quality ([3]))**.** *In a typical execution, the chain quality property (Definition 4.2) holds with parameters $l \geq 2sf$ and ration of honest blocks μ such that the following condition holds: $Z(s) < (1-\mu)X(s)$.*

Post-quantum typical executions. As it turns out, the definition of classical typical executions (Def. 4.6) is not adequate for our post-quantum analysis, for the following main reason. Note that in the classical random oracle model, in terms of solving a PoW, everyone including malicious players runs the same procedure. Hence, there is a *universal* well-defined distribution on $Z(s)$, regardless what an adversary does otherwise, based on which we can discuss its expectation. When we consider a quantum adversary, however, it is unclear how to formulate an appropriate "expectation" that is independent of the specific adversary's quantum strategy.

To address this, we observe that some alternative quantitative characterizations of a typical execution in [3] actually admit simple counterparts in the quantum setting. Hence we adopt these as our notion of typical executions below. As a result, we can lift the classical analysis in step 2 almost verbatim to show that common prefix and chain quality hold under this new definition of typical execution.

Finally, we will use the hardness of Chain-of-PoWs to derive a quantum analogue of a honest-majority condition, under which typical execution occurs with overwhelming probability.

Definition 4.11 (Post-quantum typical execution)**.** *An execution is post-quantum (ε, s)-typical (or just post-quantum typical), for $\varepsilon \in (0,1)$ and $sf \geq 2$, if for any set S of at least s consecutive rounds, the following hold:*

(a) $(1-\epsilon)fs < X(s) < (1+\epsilon)fs$ *and* $(1-\epsilon)\mathbb{E}[Y(s)] < Y(s)$.

(b) $Z(s) < (1-\epsilon)f(1-f)s$.

(c) No insertions, no copies, and no predictions occurred.

Note that Conditions (a) and (c) remain unchanged, and in particular, (a) concerns honest parties only. As in the classical setting, under our new definition of a typical execution, the desired properties of a blockchain follow easily. The second condition follows from the derived bounds on the hardness of the Chain-of-PoWs problem.

Lemma 4.12. *In a post-quantum typical execution in the presence of any quantum adversary,*

- *The common prefix property of the Bitcoin backbone protocol holds with parameter $k \geq 2sf$, for any $s \geq \frac{2}{f}$ consecutive rounds;*
- *The chain quality property holds with parameter $l \geq 2sf$ and ratio of honest blocks μ with $\mu = f$.*

Proof. Using Lemma 4.8 we know that we can ensure that any $k \geq 2fs$ consecutive blocks of a chain have been computed in $s \geq \frac{k}{2f}$ consecutive rounds, as long as we can impose the following condition: for any quantum adversary $\mathcal{A}$ and for any $s \geq \frac{2}{f}$, we have: $X(s) + Z(s) < 2fs$. Then, in a typical execution (Def. 4.11) we have that $X(s) < (1+\epsilon)fs$ which implies that we must have $\frac{Z(s)}{s} < (1-\epsilon)f$.

Secondly, the condition between $Z(s)$ and $Y(s)$ comes from Lemma 4.9, which then implies the common prefix property. To apply Lemma 4.9, what we must guarantee is that for any quantum adversary $\mathcal{A}$ and for $s \geq \frac{2}{f}$, we have: $Z(s) < Y(s)$. Therefore, in order to prove that the common prefix property holds with parameter $k \geq 2sf$, it is sufficient to impose on the quantum adversary the following two conditions for any $s \geq \frac{2}{f}$ consecutive rounds:

$$X(s) + Z(s) < 2fs \quad ; \quad Z(s) < Y(s) \tag{4.4}$$

Using the bounds on the honest players variables $X(s)$ and $Y(s)$ from Lemma 4.7, the sufficient conditions become:

$$\frac{Z(s)}{s} < (1-\epsilon)f \quad ; \quad \frac{Z(s)}{s} < (1-\epsilon)f(1-f) \tag{4.5}$$

Given that $\min\{(1-\epsilon)f(1-f),(1-\epsilon)f\} = (1-\epsilon)f(1-f)$, this leads to:

$$\frac{Z(s)}{s} < (1-\epsilon)f(1-f) \tag{4.6}$$

which is exactly the second condition of the post-quantum typical execution (Def. 4.11).
For the chain quality property, the statement follows directly from the proof of chain quality in Lemma 4.10.

□

Finally we are just left to find out an appropriate quantum analogue of the honest-majority condition, and show that it will ensure a post-quantum typical execution occurs with high probability.

Definition 4.13 (Post-quantum honest majority)**.** *We say that the post-quantum honest majority condition*[5] *holds if:*

$$Q \leq \frac{(1-\epsilon)f(1-f)}{(1+\epsilon)e\sqrt{p}} \tag{4.7}$$

where Q denotes the total number of quantum queries performed by the adversary per round.

Lemma 4.14. *Under the post-quantum honest majority condition (Def. 4.13), the probability of a post-quantum typical execution is:*

$$P_q = 1 - e^{-\Omega((1-\epsilon)f(1-f)s)}. \tag{4.8}$$

Proof. We examine the bound we obtained in Lemma 4.5 for the Chain-of-PoWs problem. A simplified form of the bound on this probability success (derived by using $\frac{2(1-p)}{\pi k} < 1$), can be expressed as:

$$P(N,k) \leq \exp\left(-2k \cdot \ln\left(\frac{k}{e(sQ+k)} \cdot \frac{1}{\sqrt{p}}\right)\right) \tag{4.9}$$

It is clear that this expression decays as a function of the number of rounds s only if $\frac{k}{e(sQ+k)} \cdot \frac{1}{\sqrt{p}} > 1$. Therefore, asymptotically (where $N \gg k$), an adversary can solve at most $k < s \cdot Q \cdot e\sqrt{p}$ PoWs. [6].

Next, we define the following function for the number of chained PoWs, as a function of the number of consecutive rounds s:

$$k_0(s) := s \cdot (1+\epsilon)eQ\sqrt{p} \tag{4.10}$$

where we recall that ϵ refers to the concentration quality. Now, for this target function $k_0(s)$, we first observe that the probability the adversary achieves this number of chained PoWs is:

$$P(N,k_0) \leq \exp\left(-2e(1+\epsilon)sQ\sqrt{p}\ln\left(\frac{1+\epsilon}{1+e(1+\epsilon)\sqrt{p}}\right)\right) \tag{4.11}$$

Hence, we notice that $P(N,k_0)$ decays exponentially as the number of rounds s increases (for any choice of $\epsilon > \frac{e\sqrt{p}}{1-e\sqrt{p}}$).

[5]See the proof of Theorem 4.14 for intuition on how this expression arises.

[6]Note that this is very close to the expression we obtained in [59], and is actually a slightly tighter bound since $c \approx 8 < e^2$.

Then, it is sufficient to obtain an honest majority condition by imposing that the honest players can achieve $k_0(s)$ chained PoWs. As a result, by using Definition 4.11 , the sufficient condition becomes:

$$k_0(s) \leq (1-\epsilon)f(1-f)s \tag{4.12}$$

Consequently, the honest majority condition can be described by the following bound on the quantum adversarial hashing power:

$$Q \leq \frac{(1-\epsilon)f(1-f)}{(1+\epsilon)e\sqrt{p}} \tag{4.13}$$

Computing the probability P_{q} of the post-quantum honest majority follows directly by using Lemma 4.5 under the choice of parameters: $N = sQ$ and $k = (1-\epsilon)f(1-f)s$. □

Corollary 4.15. *The required number of rounds for safe settlement against quantum adversaries is*

$$s_q = O\left(\frac{\epsilon^2}{(1-\epsilon)(1-f)} \cdot s_{cl}\right) \tag{4.14}$$

where s_{cl} is the required number of rounds in the classical setting [3].

On parallel quantum processors. A final note is to highlight an assumption we made about the depth of the quantum computation of the adversary in s rounds which is sQ. This assumption overestimates the power of the quantum adversary, since one can imagine an adversary that controls two or more quantum processors, so the overall depth of the computation is smaller than the total number of queries. It is known that search algorithms are not parallelizable, meaning that we have been over-pessimistic on the honest majority condition we derived. On the one hand, as we have mentioned in Section 1.3, the recording techniques cannot generalize to this setting. On the other hand, based on simple calculations of restricted parallel quantum adversaries, the factor $p^{-1/2}$ remains, reduced at best by a constant factor. Thus, qualitatively, our results persist in that case.

Acknowledgements

PW acknowledges support by the UK Hub in Quantum Computing and Simulation, part of the UK National Quantum Technologies Programme with funding from UKRI EPSRC grant EP/T001062/1. AC acknowledges support from the National Science Foundation grant CCF-1813814 and from the AFOSR under Award Number FA9550-20-1-0108. FS was supported by the US National Science Foundation grants CCF-2042414 and CCF-2054758 (CAREER). The work of JG was supported by National Science Foundation grants CNS-2001082 and CNS-2055694.

References

[1] Cynthia Dwork and Moni Naor. "Pricing via processing or combatting junk mail". In Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. Volume 740 of Lecture Notes in Computer Science, pages 139–147. Springer (1992). 2

[2] Satoshi Nakamoto. "Bitcoin open source implementation of p2p currency". http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source (2009). http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source. 2

[3] Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications". In Elisabeth Oswald and Marc Fischlin, editors, Advances in Cryptology - EUROCRYPT 2015. Pages 281–310. Berlin, Heidelberg (2015). Springer Berlin Heidelberg. 3, 4, 5, 7, 8, 18, 19, 22, 23, 25

[4] Rafael Pass, Lior Seeman, and Abhi Shelat. "Analysis of the blockchain protocol in asynchronous networks". In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, Advances in Cryptology - EUROCRYPT 2017. Volume 10211 of Lecture Notes in Computer Science. (2017). 3

[5] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. "The bitcoin backbone protocol with chains of variable difficulty". In Jonathan Katz and Hovav Shacham, editors, Advances in Cryptology – CRYPTO 2017. Pages 291–323. Cham (2017). Springer International Publishing. 3, 6

[6] Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. "Bitcoin as a transaction ledger: A composable treatment". In Jonathan Katz and Hovav Shacham, editors, Advances in Cryptology – CRYPTO 2017. Pages 324–356. Cham (2017). Springer International Publishing. 3

[7] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols". In CCS '93. Pages 62–73. (1993). 3

[8] Peter W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". SIAM J. Comput. **26**, 1484–1509 (1997). 3

[9] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. "Breaking symmetric cryptosystems using quantum period finding". In Matthew Robshaw and Jonathan Katz, editors, Advances in Cryptology – CRYPTO 2016. Pages 207–237. Berlin, Heidelberg (2016). Springer Berlin Heidelberg. 3

[10] Thomas Santoli and Christian Schaffner. "Using simon's algorithm to attack symmetric-key cryptographic primitives". Quantum Information and Computation **17**, 65–78 (2017). 3

[11] Jeroen Van De Graaf. "Towards a formal definition of security for quantum protocols". PhD thesis. Universite de Montreal. CAN (1998). 3

[12] John Watrous. "Zero-knowledge against quantum attacks". In Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing. Page 296–305. STOC '06New York, NY, USA (2006). Association for Computing Machinery. 3

[13] Dominique Unruh. "Quantum proofs of knowledge". In David Pointcheval and Thomas Johansson, editors, Advances in Cryptology – EUROCRYPT 2012. Pages 135–152. Berlin, Heidelberg (2012). Springer Berlin Heidelberg. 3

[14] Sean Hallgren, Adam Smith, and Fang Song. "Classical cryptographic protocols in a quantum world". In Phillip Rogaway, editor, Advances in Cryptology – CRYPTO 2011. Pages 411–428. Berlin, Heidelberg (2011). Springer Berlin Heidelberg. 3

[15] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. "Unforgeable quantum encryption". In Jesper Buus Nielsen and Vincent Rijmen, editors, Advances in Cryptology – EUROCRYPT 2018. Pages 489–519. Cham (2018). Springer International Publishing. 3

[16] Dan Boneh and Mark Zhandry. "Quantum-secure message authentication codes". In Thomas Johansson and Phong Q. Nguyen, editors, Advances in Cryptology – EUROCRYPT 2013. Pages 592–608. Berlin, Heidelberg (2013). Springer Berlin Heidelberg. 3

[17] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world". In Dong Hoon Lee and Xiaoyun Wang, editors, Advances in Cryptology – ASIACRYPT 2011. Pages 41–69. Berlin, Heidelberg (2011). Springer Berlin Heidelberg. 3

[18] Mark Zhandry. "How to record quantum queries, and applications to quantum indifferentiability". In Alexandra Boldyreva and Daniele Micciancio, editors, Advances in Cryptology – CRYPTO 2019. Pages 239–268. Cham (2019). Springer International Publishing. 4, 6, 9

[19] Troy Lee and Jérémie Roland. "A strong direct product theorem for quantum query complexity". computational complexity **22**, 429–462 (2013). 4, 5, 10

[20] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. "Quantum-secure message authentication via blind-unforgeability". In Advances in Cryptology – EUROCRYPT 2020. Springer (2020). 4

[21] Yassine Hamoudi and Frédéric Magniez. "Quantum Time-Space Tradeoff for Finding Multiple Collision Pairs". In Min-Hsiu Hsieh, editor, 16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021). Volume 197 of Leibniz International Proceedings in Informatics (LIPIcs), pages 1:1–1:21. Dagstuhl, Germany (2021). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 4, 5, 30

[22] Qipeng Liu and Mark Zhandry. "On finding quantum multi-collisions". In Yuval Ishai and Vincent Rijmen, editors, Advances in Cryptology – EUROCRYPT 2019. Pages 189–218. Cham (2019). Springer International Publishing. 5, 9

[23] Falk Unger. "A probabilistic inequality with applications to threshold direct-product theorems". In 2009 50th Annual IEEE Symposium on Foundations of Computer Science. Pages 221–229. IEEE (2009). 5

[24] H. Klauck, R. de Wolf, and R. Špalek. "Quantum and classical strong direct product theorems and optimal time-space tradeoffs". In 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. Pages 12–21. Los Alamitos, CA, USA (2004). IEEE Computer Society. 5, 10

[25] Alexander A Sherstov. "Strong direct product theorems for quantum communication and query complexity". SIAM Journal on Computing **41**, 1122–1165 (2012). 5, 10

[26] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. "Quantum lower bounds by polynomials". J. ACM **48**, 778–797 (2001). 5

[27] Andris Ambainis. "Quantum lower bounds by quantum arguments". J. Comput. Syst. Sci. **64**, 750–767 (2002). 5

[28] Christof Zalka. "Grover's quantum searching algorithm is optimal". Phys. Rev. A **60**, 2746–2751 (1999). 5

[29] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. "Tight bounds on quantum searching". Fortschritte der Physik **46**, 493–505 (1998). 5

[30] Andris Ambainis, Robert Špalek, and Ronald de Wolf. "A new quantum lower bound method, with applications to direct product theorems and time-space tradeoffs". Algorithmica **55**, 422–461 (2009). 5, 10

[31] Andris Ambainis. "A new quantum lower bound method, with an application to a strong direct product theorem for quantum search". Theory of Computing **6**, 1–25 (2010). 5

[32] Juan A. Garay, Aggelos Kiayias, Nikos Leonardos, and Giorgos Panagiotakos. "Bootstrapping the blockchain, with applications to consensus and fast pki setup". In Michel Abdalla and Ricardo Dahab, editors, Public-Key Cryptography – PKC 2018. Pages 465–495. Cham (2018). Springer International Publishing. 6

[33] Juan A. Garay, Aggelos Kiayias, and Giorgos Panagiotakos. "Iterated search problems and blockchain security under falsifiable assumptions". Cryptology ePrint Archive, Report 2019/315 (2019). https://eprint.iacr.org/2019/315. 6

[34] Ittay Eyal and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable". In Nicolas Christin and Reihaneh Safavi-Naini, editors, Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers. Volume 8437 of Lecture Notes in Computer Science, pages 436–454. Springer (2014). 8

[35] Divesh Aggarwal, Gavin Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. "Quantum attacks on bitcoin, and how to protect against them". Ledger**3** (2018). 9

[36] Troy Lee, Maharshi Ray, and Miklos Santha. "Strategies for Quantum Races". In Avrim Blum, editor, 10th Innovations in Theoretical Computer Science Conference (ITCS 2019). Volume 124 of Leibniz International Proceedings in Informatics (LIPIcs), pages 51:1–51:21. Dagstuhl, Germany (2018). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 9

[37] Or Sattath. "On the insecurity of quantum bitcoin mining". Int. J. Inf. Secur. **19**, 291–302 (2020). 9

[38] Andrea Coladangelo and Or Sattath. "A Quantum Money Solution to the Blockchain Scalability Problem". Quantum **4**, 297 (2020). 9

[39] Mark Zhandry. "How to construct quantum random functions". In 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science. Pages 679–687. (2012). 9

[40] Mark Zhandry. "Secure identity-based encryption in the quantum random oracle model". In Reihaneh Safavi-Naini and Ran Canetti, editors, Advances in Cryptology – CRYPTO 2012. Pages 758–775. Berlin, Heidelberg (2012). Springer Berlin Heidelberg. 9

[41] Fang Song and Aaram Yun. "Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks". In Jonathan Katz and Hovav Shacham, editors, Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II. Volume 10402 of Lecture Notes in Computer Science, pages 283–309. Springer (2017). 9

[42] Edward Eaton and Fang Song. "Making existential-unforgeable signatures strongly unforgeable in the quantum random-oracle model". In Salman Beigi and Robert König, editors, 10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015, May 20-22, 2015, Brussels, Belgium. Volume 44 of LIPIcs, pages 147–162. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2015). 9

[43] Dominique Unruh. "Non-interactive zero-knowledge proofs in the quantum random oracle model". In Elisabeth Oswald and Marc Fischlin, editors, Advances in Cryptology - EUROCRYPT 2015. Pages 755–784. Berlin, Heidelberg (2015). Springer Berlin Heidelberg. 9

[44] Andreas Hülsing, Joost Rijneveld, and Fang Song. "Mitigating multi-target attacks in hash-based signatures". In Proceedings, Part I, of the 19th IACR International Conference on Public-Key Cryptography — PKC 2016 - Volume 9614. Pages 387–416. Berlin, Heidelberg (2016). Springer-Verlag. 9

[45] Marko Balogh, Edward Eaton, and Fang Song. "Quantum collision-finding in non-uniform random functions". In Tanja Lange and Rainer Steinwandt, editors, Post-Quantum Cryptography. Pages 467–486. Cham (2018). Springer International Publishing. 9

[46] Ben Hamlin and Fang Song. "Quantum security of hash functions and property-preservation of iterated hashing". In Jintai Ding and Rainer Steinwandt, editors, Post-Quantum Cryptography. Pages 329–349. Cham (2019). Springer International Publishing. 9

[47] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A modular analysis of the fujisaki-okamoto transformation". In Yael Kalai and Leonid Reyzin, editors, Theory of Cryptography. Pages 341–371. Cham (2017). Springer International Publishing. 9

[48] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. "Tightly-secure key-encapsulation mechanism in the quantum random oracle model". In Jesper Buus Nielsen and Vincent Rijmen, editors, Advances in Cryptology – EUROCRYPT 2018. Pages 520–551. Cham (2018). Springer International Publishing. 9

[49] Andris Ambainis, Mike Hamburg, and Dominique Unruh. "Quantum security proofs using semi-classical oracles". In Alexandra Boldyreva and Daniele Micciancio, editors, Advances in Cryptology – CRYPTO 2019. Pages 269–295. Cham (2019). Springer International Publishing. 9

[50] Qipeng Liu and Mark Zhandry. "Revisiting post-quantum fiat-shamir". In Alexandra Boldyreva and Daniele Micciancio, editors, Advances in Cryptology – CRYPTO 2019. Pages 326–355. Cham (2019). Springer International Publishing. 9

[51] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the fiat-shamir transformation in the quantum random-oracle model". In Alexandra Boldyreva and Daniele Micciancio, editors, Advances in Cryptology – CRYPTO 2019. Pages 356–383. Cham (2019). Springer International Publishing. 9

[52] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shi-Feng Sun. "Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and cca security". In Annual International Conference on the Theory and Applications of Cryptographic Techniques. Pages 703–728. Springer (2020). 9

[53] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. "Tight quantum time-space tradeoffs for function inversion". In 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS). Pages 673–684. IEEE (2020). 9

[54] Shuichi Katsumata, Kris Kwiatkowski, Federico Pintore, and Thomas Prest. "Scalable ciphertext compression techniques for post-quantum kems and their applications". In International Conference on the Theory and Application of Cryptology and Information Security. Pages 289–320. Springer (2020). 9

[55] Jan Czajkowski. "Quantum indifferentiability of sha-3". Cryptology ePrint Archive, Report 2021/192 (2021). https://ia.cr/2021/192. 9

[56] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. "On the compressed-oracle technique, and post-quantum security of proofs of sequential work". In Anne Canteaut and François-Xavier Standaert, editors, Advances in Cryptology – EUROCRYPT 2021. Pages 598–629. Cham (2021). Springer International Publishing. 9

[57] Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. "On the Security of Proofs of Sequential Work in a Post-Quantum World". In Stefano Tessaro, editor, 2nd Conference on Information-Theoretic Cryptography (ITC 2021). Volume 199 of Leibniz International Proceedings in Informatics (LIPIcs), pages 22:1–22:27. Dagstuhl, Germany (2021). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 9

[58] Dominique Unruh. "Compressed permutation oracles (and the collision-resistance of sponge/sha3)". Cryptology ePrint Archive, Report 2021/062 (2021). https://eprint.iacr.org/2021/062. 9

[59] Alexandru Cojocaru, Juan Garay, Aggelos Kiayias, Fang Song, and Petros Wallden. "The bitcoin backbone protocol against quantum adversaries". Cryptology ePrint Archive, Paper 2019/1150 (2019). https://eprint.iacr.org/2019/1150. 9, 24

[60] Ran Canetti. "Security and composition of multiparty cryptographic protocols". J. Cryptology **13**, 143–202 (2000). 18

[61] Ran Canetti. "Universally composable security: A new paradigm for cryptographic protocols". In 42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA. Pages 136–145. IEEE Computer Society (2001). 18

A Comparison with Other Query Complexity Work

In this section we compare three different approaches to the query complexity analysis of the k-BerSearch problem, in terms of their performance (not the techniques to derive the bounds). We will denote the three different bounds of the works we are comparing as follows:

- Gen_1 - analysis performed in the work of [21] leading to success probability $\bar{p}_k^N$;
- Gen_2 - analysis performed in the current work leading to success probability p_k^N;
- NonS - analysis in our previous work leading to success probability P_{NonS};

We note that Gen_1 and Gen_2 deal with the most general (powerful) quantum adversaries, while in NonS the adversary is restricted in treating the number of queries to the oracle as a classical variable. The comparison can be summarized as follows:

	NonS	Gen_1	Gen_2
Quantum Adversary	Restricted	Most General	Most General
Success Probability	For $k = (1+\epsilon)\sqrt{cp}N$ $P_{\mathsf{NonS}} := e^{-(cp)^{\frac{2}{3}} N \cdot g_2(\epsilon)}$	$\bar{p}_k^N \leq 2\left(\frac{8eN\sqrt{p}}{k}\right)^k + \left(\frac{1}{2}\right)^k$	$p_k^N \leq \frac{2(1-p)}{\pi k} \cdot \left(\frac{eN\sqrt{p}}{k}\right)^{2k}$
Expected Optimal[7]	$\sqrt{cp}N$	$8e\sqrt{p}N$	$e\sqrt{p}N$
Convergence Around Optimal	$\exp\left(-N \cdot O(p^{2/3})\right)$	$\exp\left(-N \cdot O(p^{1/2})\right)$	$\exp\left(-N \cdot O(p^{1/2})\right)$

Table 3: Analysis of the k-BerSearch Problem

The first thing to observe is that for the analysis of Gen_1 and Gen_2, their results can be directly compared in terms of probability of success as function of the number of available queries and size of solution for the k-BerSearch problem. Hence, we can notice from the two

[7] Note, that by "Expected Optimal", we mean the value of k above which the given expressions begin to converge (decay). To see how quickly the tails converges, we simply need to consider $k = (1+\epsilon) \times$ Expected Optimal.

functions $\bar{p}_k^N$ and p_k^N, that our current work provides a tighter bound (besides the extra term $1/2^k$, their function has for the first term exponent k, while our expression has almost the same term with exponent $2k$). Hence, we remark that while the result for expected optimal length is only a constant better in our case compared to Gen_1, the bound on the probability of success we obtained is much stronger than the one obtained in Gen_1 (roughly our bound is the square of the bound in Gen_1).

Second thing to note is that from the form of the success probability we can deduce the smallest value of k, above which each expression decays. This can be viewed as a bound on the expected optimal quantum strategy, since these bounds rule-out any larger value of k (on average). Here we see that the best (tightest) bound on the best average quantum strategy is given by this current work and next is our previous work, with looser bound given in Gen_1: $e\sqrt{p}N < \sqrt{cp}N < 8e\sqrt{p}N$. Recall that c is given by bounds on the query complexity of standard search and is known to be $c \sim 8$, making our new and old results very close.

A direct consequence is that the value of k that Gen_1 starts decaying is eight times bigger than the value that Gen_2 starts decaying. This means that Gen_1 overestimates the capabilities of what a quantum party (adversary) can achieve eightfold compared to Gen_2. In the language of the backbone Bitcoin protocol, this means that Gen_1 leads to much worse honest majority.

The third thing to consider, is the "speed" of convergence to zero, as each of the cases considered moves away from its own average value. Here we see that Gen_2 and Gen_1 both perform better (equally well) by having a factor of $\exp\left(-NO(p^{1/2})\right)$, than our earlier work NonS that has a factor of $\exp\left(-NO(p^{2/3})\right)$. In the language of the backbone Bitcoin protocol, this means that $\mathsf{Gen}_1, \mathsf{Gen}_2$ have smaller settlement time (actually essentially the same with the classical case), while our earlier work NonS has greater settlement time by a factor of $O(p^{-1/6})$. Of course, the settlement time is determined by assuming that the adversary controls the maximum allowed, by the honest majority, computational power. Given that Gen_1 requires that the honest parties must have much bigger advantage (many more queries) than in NonS, in practice even if such honest majority is assumed, NonS might perform better.

B Simplified Bound of Theorem 3.1

Proof. We start with denoting the sum: $S_{k,N} := \sum_{i=0}^{k}(\sqrt{1-p})^i\binom{N}{i}$. Firstly we drop the $\sqrt{1-p}^i$ factors ($\sqrt{1-p}^i < 1$), so instead we will work with the sums: $S'_{N,k} := \sum_{i=0}^{k}\binom{N}{i}$.
We will bound the ratio:

$$R_{N,k} := \frac{S'_{N,k}}{S'_{N,k-1}} = \frac{S'_{N,k-1} + \binom{N}{k}}{S'_{N,k-1}} = 1 + \frac{\binom{N}{k}}{S'_{N,k-1}} \tag{B.1}$$

To upper bound the ratio $R_{N,k}$, we will lower bound the sum $S'_{N,k-1}$ with the last two terms in the sum: $S'_{N,k-1} > \binom{N}{k-1} + \binom{N}{k-2} = \binom{N+1}{k-1}$. Hence we will upper bound the ratio as:

$$R_{N,k} \leq 1 + \frac{\binom{N}{k}}{\binom{N+1}{k-1}} = \frac{k(N+1) + (N-k+1)(N-k+2)}{k(N+1)} \tag{B.2}$$

From this we obtain that:

$$R_{N,k} \leq \frac{N}{k} \tag{B.3}$$

which holds as long as $4 \leq k \leq N$.
This implies that: $S'_{N,k} < \frac{N}{k}S'_{N,k-1}$. By applying this iteratively, and by using that $S'_{N,0} = 1$,

we obtain:

$$S'_{N,k} \leq \frac{N^k}{k!} \tag{B.4}$$

Hence, as $p_k^N \leq 4(1-p)p^k {S'_{N,k}}^2$, we have:

$$p_k^N \leq 4\left[\sqrt{1-p} \cdot \sqrt{p}^k \frac{N^k}{k!}\right]^2 \tag{B.5}$$

Finally, using Stirling approximation for: $k! > \sqrt{2\pi k} \cdot \frac{k^k}{e^k}$, we get:

$$p_k^N \leq 4\left[\sqrt{1-p} \cdot \sqrt{p}^k \left(\frac{Ne}{k}\right)^k \cdot \frac{1}{\sqrt{2\pi k}}\right]^2 \tag{B.6}$$

□