





## III.2 Copies of Representative Publications

1. Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. “Oblivious transfer is in minicrypt”. In: *Advances in Cryptology – EUROCRYPT 2021*. Springer, 2021, pp. 531–561. DOI: [10.1007/978-3-030-77886-6\\_18](https://doi.org/10.1007/978-3-030-77886-6_18)
2. Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. “Zero-Knowledge Proof Systems for QMA”. in: *SIAM J. Comput.* 49.2 (2020), pp. 245–283. DOI: [10.1137/18M1193530](https://doi.org/10.1137/18M1193530)
3. Zhengfeng Ji, Yi-Kai Liu, and Fang Song. “Pseudorandom Quantum States”. In: *Advances in Cryptology – CRYPTO 2018*. Springer, 2018, pp. 126–152. DOI: [10.1007/978-3-319-96878-0\\_5](https://doi.org/10.1007/978-3-319-96878-0_5)
4. Fang Song and Aaram Yun. “Quantum Security of NMAC and Related Constructions - PRF Domain Extension Against Quantum attacks”. In: *Advances in Cryptology - CRYPTO 2017*. Springer, 2017, pp. 283–309. DOI: [10.1007/978-3-319-63715-0\\_10](https://doi.org/10.1007/978-3-319-63715-0_10)
5. Jean-François Biasse and Fang Song. “Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields”. In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*. SIAM, 2016, pp. 893–902. DOI: [10.1137/1.9781611974331.ch64](https://doi.org/10.1137/1.9781611974331.ch64)
6. Kirsten Eisenträger, Sean Hallgren, Alexei Y. Kitaev, and Fang Song. “A quantum algorithm for computing the unit group of an arbitrary degree number field”. In: *Proceedings of the forty-sixth annual ACM Symposium on Theory of Computing, STOC 2014*. ACM, 2014, pp. 293–302. DOI: [10.1145/2591796.2591860](https://doi.org/10.1145/2591796.2591860)



# Oblivious Transfer Is in MiniQCrypt

Alex B. Grilo<sup>1</sup>() , Huijia Lin<sup>2</sup>() , Fang Song<sup>3</sup>() ,  
and Vinod Vaikuntanathan<sup>4</sup>()

<sup>1</sup> CNRS, LIP6, Sorbonne Université, Paris, France

`Alex.Bredariol-Grilo@lip6.fr`

<sup>2</sup> University of Washington, Seattle, WA, USA

`rachel@cs.washington.edu`

<sup>3</sup> Portland State University, Portland, OR, USA

`fsong@pdx.edu`

<sup>4</sup> MIT, Cambridge, MA, USA

`vinodv@csail.mit.edu`

**Abstract.** MiniQCrypt is a world where quantum-secure one-way functions exist, and quantum communication is possible. We construct an oblivious transfer (OT) protocol in MiniQCrypt that achieves simulation-security in the plain model against malicious quantum polynomial-time adversaries, building on the foundational work of Crépeau and Kilian (FOCS 1988) and Bennett, Brassard, Crépeau and Skubiszewska (CRYPTO 1991). Combining the OT protocol with prior works, we obtain secure two-party and multi-party computation protocols also in MiniQCrypt. This is in contrast to the classical world, where it is widely believed that one-way functions alone do not give us OT.

In the common random string model, we achieve a *constant-round* universally composable (UC) OT protocol.

## 1 Introduction

Quantum computing and modern cryptography have enjoyed a highly productive relationship for many decades ever since the conception of both fields. On the one hand, (large-scale) quantum computers can be used to break many widely used cryptosystems based on the hardness of factoring and discrete logarithms, thanks to Shor’s algorithm [60]. On the other hand, quantum information and computation have helped us realize cryptographic tasks that are otherwise impossible, for example quantum money [65] and generating certifiable randomness [13, 17, 63].

Yet another crown jewel in quantum cryptography is the discovery, by Bennett and Brassard [8], of a key exchange protocol whose security is unconditional. That is, they achieve information-theoretic security for a cryptographic task that classically necessarily has to rely on unproven computational assumptions. In a nutshell, they accomplish this using the uncloneability of quantum states, a bedrock principle of quantum mechanics. What’s even more remarkable is the

---

A full version of this paper appears on ePrint Archive Report 2020/1500 [35].

© International Association for Cryptologic Research 2021

A. Canteaut and F.-X. Standaert (Eds.): EUROCRYPT 2021, LNCS 12697, pp. 531–561, 2021.

[https://doi.org/10.1007/978-3-030-77886-6\\_18](https://doi.org/10.1007/978-3-030-77886-6_18)

fact that their protocol makes minimalistic use of quantum resources, and consequently, has been implemented in practice over very large distances [23, 45]. This should be seen in contrast to large scale quantum *computation* whose possibility is still being actively debated.

Bennett and Brassard’s groundbreaking work raised a *tantalizing* possibility for the field of cryptography:

*Could every cryptographic primitive  
be realized unconditionally using quantum information?*

A natural next target is oblivious transfer (OT), a versatile cryptographic primitive which, curiously, had its origins in Wiesner’s work in the 1970s on quantum information [65] before being rediscovered in cryptography by Rabin [56] in the 1980s. Oblivious transfer (more specifically, 1-out-of-2 OT) is a two-party functionality where a receiver Bob wishes to obtain one out of two bits that the sender Alice owns. The OT protocol must ensure that Alice does not learn which of the two bits Bob received, and that Bob learns only one of Alice’s bits and no information about the other. Oblivious transfer lies at the foundation of secure computation, allowing us to construct protocols for the secure multiparty computation (MPC) of any polynomial-time computable function [33, 42, 43].

Crépeau and Killian [19] and Bennett, Brassard, Crépeau and Skubiszewska [9] constructed an OT protocol given an *ideal* bit commitment protocol and quantum communication. In fact, the only quantum communication in their protocol consisted of Alice sending several so-called “BB84 states” to Bob. Unfortunately, *unconditionally secure* commitment [49, 53] and *unconditionally secure* OT [16, 48] were soon shown to be impossible even with quantum resources.

However, given that bit commitment can be constructed from one-way functions (OWF) [37, 54], the hope remains that OT, and therefore a large swathe of cryptography, can be based on only *OWF* together with (practically feasible) quantum communication. Drawing our inspiration from Impagliazzo’s five worlds in cryptography [39], we call such a world, where post-quantum secure one-way functions (pqOWF) exist and quantum computation and communication are possible, MiniQCrypt. The question that motivates this paper is:

*Do OT and MPC exist in MiniQCrypt?*

Without the quantum power, this is widely believed to be impossible. That is, given only OWFs, there are no *black-box* constructions of OT or even key exchange protocols [40, 57]. The fact that [8] overcome this barrier and construct a key exchange protocol with quantum communication (even without the help of OWFs) reinvigorates our hope to do the same for OT.

**Aren’t We Done Already?** At this point, the reader may wonder why we do not have an affirmative answer to this question already, by combining the OT protocol of [9, 19] based on bit commitments, with a construction of bit commitments from pqOWF [37, 54]. Although this possibility was mentioned already in [9], where they note that “...computational complexity based quantum cryptography is interesting since it allows to build oblivious transfer around one-way functions.”, attaining this goal remains elusive as we explain below.

First, proving the security of the [9,19] OT protocol (regardless of the assumptions) turns out to be a marathon. After early proofs against limited adversaries [52,66], it is relatively recently that we have a clear picture with formal proofs against arbitrary quantum polynomial-time adversaries [12,20,21,61]. Based on these results, we can summarize the state of the art as follows.

- *Using Ideal Commitments:* If we assume an *ideal* commitment protocol, formalized as universally composable (UC) commitment, then the quantum OT protocol can be proven secure in strong simulation-based models, in particular the quantum UC model that admits sequential composition or even concurrent composition in a network setting [12,20,30,61]. However, UC commitments, in contrast to vanilla computationally-hiding and statistically-binding commitments, are powerful objects that do not live in Minicrypt. In particular, UC commitments give us key exchange protocols and are therefore black-box separated from Minicrypt.<sup>1</sup>
- *Using Vanilla Commitments:* If in the [9,19] quantum OT protocol we use a *vanilla* statistically-binding and computationally hiding commitment scheme, which exists assuming a pqOWF, the existing proofs, for example [12], fall short in two respects.

First, for a malicious receiver, the proof of [12] constructs only an *inefficient* simulator. Roughly speaking, this is because the OT receiver in [9,19] acts as a committer, and vanilla commitments are not extractable. Hence, we need an inefficient simulator to extract the committed value by brute force. Inefficient simulation makes it hard, if not impossible, to use the OT protocol to build other protocols (even if we are willing to let the resulting protocol have inefficient simulation). Our work will focus on achieving the standard ideal/real notion of security [32] with efficient simulators.

Secondly, it is unclear how to construct a simulator (even ignoring efficiency) for a malicious sender. Roughly speaking, the issue is that simulation seems to require that the commitment scheme used in [9,19] be secure against selective opening attacks, which vanilla commitments do not guarantee [6].

- *Using Extractable Commitments:* It turns out that the first difficulty above can be addressed if we assume a commitment protocol that allows *efficient extraction* of the committed value – called extractable commitments. Constructing extractable commitments is surprisingly challenging in the quantum world because of the hardness of rewinding. Moreover, to plug into the quantum OT protocol, we need a strong version of extractable commitments from which the committed values can be extracted efficiently *without destroying or*

---

<sup>1</sup> The key exchange protocol between Alice and Bob works as follows. Bob, playing the simulator for a malicious sender in the UC commitment protocol, chooses a common reference string (CRS) with a trapdoor  $TD$  and sends the CRS to Alice. Alice, playing the sender in the commitment scheme, chooses a random  $K$  and runs the committer algorithm. Bob runs the straight-line simulator-extractor (guaranteed by UC simulation) using the  $TD$  to get  $K$ , thus ensuring that Alice and Bob have a common key. An eavesdropper Eve should not learn  $K$  since the above simulated execution is indistinguishable from an honest execution, where  $K$  is hidden.

*even disturbing the quantum states of the malicious committer*,<sup>2</sup> a property that is at odds with quantum unclonability and rules out several extraction techniques used for achieving arguments of knowledge such as in [62]. In particular, we are not aware of a construction of such extractable commitments without resorting to strong assumptions such as (unleveled) quantum FHE and LWE [2, 10], which takes us out of minicrypt. Another standard way to construct extractable commitments is using public-key encryption in the CRS model, which unfortunately again takes us out of minicrypt.

To summarize, we would like to stress that before our work, the claims that quantum OT protocols can be constructed from pqOWFs [9, 28] were rooted in misconceptions.

**Why MiniQCrypt.** Minicrypt is one of five Impagliazzo’s worlds [39] where OWFs exist, but public-key encryption schemes do not. In Cryptomania, on the other hand, public-key encryption schemes do exist.

Minicrypt is robust *and* efficient. It is robust because there is an abundance of candidates for OWFs that draw from a variety of sources of hardness, and most do not fall to quantum attacks. Two examples are (OWFs that can be constructed from) the advanced encryption standard (AES) and the secure hash standard (SHA). They are “structureless” and hence typically do not have any subexponential attacks either. In contrast, cryptomania seems fragile and, to some skeptics, even endangered due to the abundance of subexponential and quantum attacks, except for a handful of candidates. It is efficient because the operations are combinatorial in nature and amenable to very fast implementations; and the key lengths are relatively small owing to OWFs against which the best known attacks are essentially brute-force key search. We refer the reader to a survey by Barak [3] for a deeper perspective.

Consequently, much research in (applied) cryptography has been devoted to minimizing the use of public-key primitives in advanced cryptographic protocols [5, 41]. However, complete elimination seems hard. In the classical world, in the absence of quantum communication, we can construct pseudorandom generators and digital signatures in Minicrypt, but not key exchange, public-key encryption, oblivious transfer or secure computation protocols. With quantum *communication* becoming a reality not just academically [23, 38, 55] but also commercially [45], we have the ability to reap the benefits of robustness and efficiency that Minicrypt affords us, *and* construct powerful primitives such as oblivious transfer and secure computation that were so far out of reach.

**Our Results.** In this paper, we finally show that the longstanding (but previously unproved) claim is true.

**Theorem 1.1 (Informal).** *Oblivious transfer protocols in the plain model that are simulation-secure against malicious quantum polynomial-time adversaries*

---

<sup>2</sup> This is because when using extractable commitment in a bigger protocol, the proof needs to extract the committed value and continue the execution with the adversary.

*exist assuming that post-quantum one-way functions exist and that quantum communication is possible.*

Our main technical contribution consists of showing a construction of an extractable commitment scheme based solely on pqOWFs and using quantum communication. Our construction involves three ingredients. The first is vanilla post-quantum commitment schemes which exist assuming that pqOWFs exist [54]. The second is post-quantum zero-knowledge protocols which also exist assuming that pqOWFs exist [64]. The third and final ingredient is a special multiparty computation protocol called conditional disclosure of secrets (CDS) constructing which in turns requires OT. This might seem circular as this whole effort was to construct an OT protocol to begin with! Our key observation is that the CDS protocol is only required to have a mild type of security, namely *unbounded simulation*, which *can* be achieved with a slight variant of the [9, 19] protocol. Numerous difficulties arise in our construction, and in particular proving consistency of a protocol execution involving quantum communication appears difficult: how do we even write down an statement (e.g., NP or QMA) that encodes consistency? Overcoming these difficulties constitutes the bulk of our technical work. We provide a more detailed discussion on the technical contribution of our work in Sect. 1.1.

We remark that understanding our protocol requires only limited knowledge of quantum computation. Thanks to the composition theorems for (stand-alone) simulation-secure quantum protocols [36], much of our protocol can be viewed as a *classical* protocol in the (unbounded simulation) OT-hybrid model. The only quantumness resides in the instantiation of the OT hybrid with [9, 19].

We notice that just as in [8, 9, 19], the honest execution of our protocols does not need strong quantum computational power, since one only needs to create, send and measure “BB84” states, which can be performed with current quantum technology.<sup>3</sup> Most notably, creating the states does not involve creating or maintaining long-range correlations between qubits.

In turn, plugging our OT protocol into the protocols of [24, 27, 42, 61] (and using the sequential composition theorem [36]) gives us secure two-party computation and multi-party computation (with a dishonest majority) protocols, even for quantum channels.

**Theorem 1.2 (Informal).** *Assuming that post-quantum one-way functions exist and quantum communication is possible, for every classical two-party and multi-party functionality  $\mathcal{F}$ , there is a quantum protocol in the plain model that is simulation-secure against malicious quantum polynomial-time adversaries. Under the same assumptions, there is a quantum two-party and multi-party protocol for any quantum circuit  $Q$ .*

Finally, we note that our OT protocol runs in  $\text{poly}(\lambda)$  number of rounds, where  $\lambda$  is a security parameter, and that is only because of the zero-knowledge

<sup>3</sup> A BB84 state is a single-qubit state that is chosen uniformly at random from  $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ . Alternatively, it can be prepared by computing  $H^h X^x |0\rangle$  where  $X$  is the bit-flip gate,  $H$  is the Hadamard gate, and  $h, x \in \{0, 1\}$  are random bits.



proof. Watrous' ZK proof system [64] involves repeating a classical ZK proof (such as that graph coloring ZK proof [34] or the Hamiltonicity proof [11]) *sequentially*. A recent work of Bitansky and Shmueli [10] for the first time constructs a *constant-round* quantum ZK protocol (using only classical resources) but they rely on a strong assumption, namely (unleveled) quantum FHE and quantum hardness of LWE, which does not live in minicrypt. Nevertheless, in the common random string (CRS) model, we can instantiate the zero-knowledge protocol using a WI protocol and a Pseudo-Random Generator (PRG) with additive  $\lambda$  bit stretch as follows: To prove a statement  $x$ , the prover proves using the WI protocol that either  $x$  is in the language or the common random string is in the image of the PRG. To simulate a proof, the simulator samples the CRS as a random image of the PRG, and proves using the WI protocol that it belongs to the image in a straight-line. Moreover, this modification allows us to achieve *straight-line simulators*, leading to *universally-composable* (UC) security [15]. Therefore, this modification would give us the following statement.

**Theorem 1.3 (Informal).** *Constant-round oblivious transfer protocols in the common random string (CRS) model that are UC-simulation-secure against malicious quantum poly-time adversaries exist assuming that post-quantum one-way functions exist and that quantum communication is possible.*

Plugging the above UC-simulation-secure OT into the protocol of [42] gives constant-round multi-party computation protocols for classical computation in the common random string model that are UC-simulation-secure against malicious quantum poly-time adversaries.

**Going Below MiniQCrypt?** We notice that all of the primitives that we implement in our work *cannot* be implemented unconditionally, even in the quantum setting [16, 48, 49, 53]. Basing their construction on pqOWFs seems to be the next best thing, but it does leave with the intriguing question if they could be based on weaker assumptions. More concretely, assume a world with quantum communication as we do in this paper. Does the existence of quantum OT protocols imply the existence of pqOWFs? Or, does a weaker *quantum* notion of one-way functions suffice? We leave the exploration of other possible cryptographic worlds below MiniQCrypt to future work.

**Other Related Work.** Inspired by the quantum OT protocol [9, 19], a family of primitives, named *k-bit cut-and-choose*, has been shown to be sufficient to realize OT statistically by quantum protocols [25, 29] which is provably impossible by classical protocols alone [51]. These offer further examples demonstrating the power of quantum cryptographic protocols.

There has also been extensive effort on designing quantum protocols OT and the closely related primitive of *one-time-memories* under *physical* rather than *computational* assumptions, such as the bounded-storage model, noisy-storage model, and isolated-qubit model, which restrict the quantum memory or admissible operations of the adversary [21, 22, 44, 46, 47, 58]. They provide important alternatives, but the composability of these protocols are not well understood.

Meanwhile, there is strengthening on the impossibility for quantum protocols to realize secure computation statistically from scratch [14, 59].

We note that there exist classical protocols for two-party and multi-party computation that are quantum-secure assuming strong assumptions such as post-quantum dense encryption and superpolynomial quantum hardness of the learning-with-errors problem [1, 36, 50]. And prior to the result in [24], there is a long line of work on secure multi-party *quantum* computation (Cf. [7, 18, 26, 27]).

We remark that the idea to use OT and ZK for obtaining extractable commitment was also used (at least implicitly) in [10, 36, 50].

Finally, we notice that [4] have independently and concurrently proposed a quantum protocol for extractable and equivocal commitments, which can be used in the protocol of [9, 19] to achieve OT (and secure multi-party computation) in MiniQCrypt. In comparison, their extractable and equivocal commitment scheme is statistically hiding, which leads to one-sided statistical security in their OT protocols. Furthermore, their commitment and OT protocols make black-box use of the underlying one-way function. Our protocols do not have these properties. On the other hand, our commitment scheme is statistically binding, and we give constant-round UC-secure protocols in the *reusable* CRS model. We also believe that our notion of verifiable CDS is of independent interest.

## 1.1 Technical Overview

We give an overview of our construction of post-quantum OT protocol in the plain model from post-quantum one-way functions. In this overview, we assume some familiarity with post-quantum MPC in the stand-alone, sequential composition, and UC models, and basic functionalities such as  $\mathcal{F}_{\text{ot}}$  and  $\mathcal{F}_{\text{com}}$ . We will also consider *parallel versions* of them, denoted as  $\mathcal{F}_{\text{p-ot}}$  and  $\mathcal{F}_{\text{so-com}}$ . The parallel OT functionality  $\mathcal{F}_{\text{p-ot}}$  enables the sender to send some polynomial number of pairs of strings  $\{s_0^i, s_1^i\}_i$  and the receiver to choose one per pair to obtain  $s_{c_i}^i$  in parallel. The commitment with selective opening functionality  $\mathcal{F}_{\text{so-com}}$  enables a sender to commit to a string  $m$  while hiding it, and a receiver to request opening of a subset of bits at locations  $T \subseteq [|m|]$  and obtain  $m_T = (m_i)_{i \in T}$ . We refer the reader to Sect. 2 for formal definitions of these functionalities.

**BBCS OT in the  $\mathcal{F}_{\text{so-com}}$ -Hybrid Model.** We start by describing the quantum OT protocol of [9] in the  $\mathcal{F}_{\text{so-com}}$  hybrid model.

**BBCS OT protocol:** The sender  $\text{ot.S}$  has strings  $s_0, s_1 \in \{0, 1\}^\ell$ , the receiver  $\text{ot.R}$  has a choice bit  $c \in \{0, 1\}$ .

1. **Preamble.**  $\text{ot.S}$  sends  $n \gg \ell$  BB94 qubits  $|x^A\rangle_{\theta^A}$  prepared using random bits  $x^A \in_R \{0, 1\}^n$  and random basis  $\theta^A \in_R \{+, \times\}^n$ .  
 $\text{ot.R}$  measures these qubits in randomly chosen bases  $\theta^B \in_R \{+, \times\}^n$  and commits to the measured bits together with the choice of the bases, that is  $\{\theta_i^B, x_i^B\}_i$ , using  $\mathcal{F}_{\text{so-com}}$ .
2. **Cut and Choose.**  $\text{ot.S}$  requests to open a random subset  $T$  of locations, of size say  $n/2$ , and gets  $\{\theta_i^B, x_i^B\}_{i \in T}$  from  $\mathcal{F}_{\text{so-com}}$ .  
 Importantly, it aborts if for any  $i$   $\theta_i^B = \theta_i^A$  but  $x_i^B \neq x_i^A$ . Roughly speaking, this is because it's an indication that the receiver has not reported honest measurement outcomes.



3. **Partition Index Set.**  $\text{ot.S}$  reveals  $\theta_T^A$  for the unchecked locations  $\bar{T}$ .  $\text{ot.R}$  partitions  $\bar{T}$  into a subset of locations where it measured in the same bases as the sender  $I_c := \{i \in \bar{T} : \theta_i^A = \theta_i^B\}$  and the rest  $I_{1-c} := \bar{T} - I_c$ , and sends  $(I_0, I_1)$  to the sender.
4. **Secret Transferring.**  $\text{ot.S}$  hides the two strings  $s_i$  for  $i = 0, 1$  using randomness extracted from  $x_{I_i}^A$  via a universal hash function  $f$  and sends  $m_i := s_i \oplus f(x_{I_i}^A)$ , from which  $\text{ot.R}$  recovers  $s := m_c \oplus f(x_{I_c}^B)$ .

Correctness follows from that for every  $i \in I_c$ ,  $\theta_i^A = \theta_i^B$  and  $x_{I_c}^A = x_{I_c}^B$ , hence the receiver decodes  $s_c$  correctly.

The security of the BBCS OT protocol relies crucially on two important properties of the  $\mathcal{F}_{\text{so-com}}$  commitments, namely extractability and equivocability, which any protocol implementing the  $\mathcal{F}_{\text{so-com}}$  functionality must satisfy.

*Equivocability:* To show the receiver's privacy, we need to efficiently simulate the execution with a malicious sender  $\text{ot.S}^*$  without knowing the choice bit  $c$  and extract both sender's strings  $s_0, s_1$ . To do so, the simulator  $\text{ot.SimS}$  would like to measure at these unchecked locations  $\bar{T}$  using exactly the same bases  $\theta_{\bar{T}}^A$  as  $\text{ot.S}^*$  sends in Step 3. In an honest execution, this is impossible as the receiver must commit to its bases  $\theta^B$  and pass the checking step. However, in simulation, this can be done by invoking the equivocability of  $\mathcal{F}_{\text{so-com}}$ . In particular,  $\text{ot.SimS}$  can *simulate* the receiver's commitments in the preamble phase without committing to any value. When it is challenged to open locations at  $T$ , it measures qubits at  $T$  in random bases, and *equivocates* commitments at  $T$  to the measured outcomes and bases. Only after  $\text{ot.S}^*$  reveals its bases  $\theta_{\bar{T}}^A$  for the unchecked locations, does  $\text{ot.SimS}$  measure qubits at  $\bar{T}$  in exactly these bases. This ensures that it learns both  $x_{I_0}^A$  and  $x_{I_1}^A$  and hence can recover both  $s_0$  and  $s_1$ .

*Extractability:* To show the sender's privacy, we need to efficiently extract the choice bit  $c$  from a malicious receiver  $\text{ot.R}^*$  and simulate the sender's messages using only  $s_c$ . To do so, the simulator  $\text{ot.SimR}$  needs to extract efficiently from the  $\mathcal{F}_{\text{so-com}}$  commitments all the bases  $\theta^B$ , so that, later given  $I_0, I_1$  it can figure out which subset  $I_c$  contains more locations  $i$  where the bases match  $\theta_i^B = \theta_i^A$ , and use the index of that set as the extracted choice bit. Observe that it is important that extraction does not “disturb” the quantum state of  $\text{ot.R}^*$  at all, so that  $\text{ot.SimR}$  can continue simulation with  $\text{ot.R}^*$ . This is easily achieved using  $\mathcal{F}_{\text{so-com}}$  as extraction is done in a straight-line fashion, but challenging to achieve in the plain model as rewinding a quantum adversary is tricky. Indeed, the argument of knowledge protocol of [62] can extract a witness but disturbs the state of the quantum adversary due to measurement. Such strong extractable commitment is only known in the plain model under stronger assumptions [2, 10, 36] or assuming public key encryption in the CRS model.

It turns out that equivocability *can* be achieved using zero-knowledge protocols, which gives a post-quantum OT protocol with an inefficient simulator  $\text{ot.SimR}$  against malicious receivers (and efficient  $\text{ot.SimS}$ ). Our main technical contribution lies in achieving efficient extractability while assuming only post-quantum one-way functions. In particular, we will use the OT with unbounded simulation as a tool for this. We proceed to describing these steps in more detail.

**Achieving Equivocability Using Zero-Knowledge.** The idea is to let the committer commit  $c = \text{com}(\mu; \rho)$  to a string  $\mu \in \{0, 1\}^n$  using any statistically binding computationally hiding commitment scheme  $\text{com}$  whose decommitment can be verified classically, for instance, Naor’s commitment scheme [54] from post-quantum one-way functions. For now in this overview, think of  $\text{com}$  as non-interactive. (Jumping ahead, later we will also instantiate this commitment with a multi-round extractable commitment scheme that we construct.)

Any computationally hiding commitment can be simulated by simply committing to zero,  $\tilde{c} = \text{com}(0; \rho)$ . The question is how to equivocate  $\tilde{c}$  to any string  $\mu'$  later in the decommitment phase. With a post-quantum ZK protocol, instead of asking the committer to reveal its randomness  $\rho$  which would statistically bind  $\tilde{c}$  to the zero string, we can ask the committer to send  $\mu'$  and give a zero-knowledge proof that  $\tilde{c}$  indeed commits to  $\mu'$ . As such, the simulator can cheat and successfully open to any value  $\mu'$  by simulating the zero-knowledge argument to the receiver.

**Equivocable Commitment:** The sender  $\text{com.S}$  has a string  $\mu \in \{0, 1\}^n$ , the receiver  $\text{com.R}$  has a subset  $T \subseteq [n]$ .

1. **Commit Phase.**  $\text{com.S}$  commits to  $\mu$  using a statistically binding commitment scheme  $\text{com}$  using randomness  $\rho$ . Let  $c$  be the produced commitment.  
NOTE: *Simulation against malicious receivers commits to  $0^n$ . Simulation against malicious senders is inefficient to extract  $\mu$  by brute force.*
2. **Decommit Phase.** Upon  $\text{com.R}$  requesting to open a subset  $T$  of locations,  $\text{com.S}$  sends  $\mu'$  and gives a single zero knowledge argument that  $c$  commits to  $\mu$  such that  $\mu' = \mu_T$ .  
NOTE: *To equivocate to  $\mu' \neq \mu_T$ , the simulator sends  $\mu'$  and simulates the zero-knowledge argument (of the false statement).*

The above commitment protocol implements  $\mathcal{F}_{\text{so-com}}$  with efficient simulation against malicious receivers, but inefficient simulation against malicious senders. Plugging it into BBCS OT protocol, we obtain the following corollary:

**Corollary 1.1 (Informal).** *Assume post-quantum one-way functions. In the plain model, there is:*

- a protocol that securely implements the OT functionality  $\mathcal{F}_{\text{ot}}$ , and
- a protocol that securely implements the parallel OT functionality  $\mathcal{F}_{\text{p-ot}}$ ,

*in the sequential composition setting, and with efficient simulation against malicious senders but inefficient simulation against malicious receivers.*

The second bullet requires some additional steps, as parallel composition does not automatically apply in the stand-alone (as opposed to UC) setting (e.g., the ZK protocol of [64] is not simulatable in parallel due to rewinding). Instead, we first observe that the BBCS OT UC-implements  $\mathcal{F}_{\text{ot}}$  in the  $\mathcal{F}_{\text{so-com}}$  hybrid model, and hence parallel invocation of BBCS OT UC-implements  $\mathcal{F}_{\text{p-ot}}$  in the  $\mathcal{F}_{\text{so-com}}$  hybrid model. Note that parallel invocation of BBCS OT invokes  $\mathcal{F}_{\text{so-com}}$  in parallel, which in fact can be merged into a single invocation to  $\mathcal{F}_{\text{so-com}}$ . Therefore, plugging in the above commitment protocol gives an OT protocol that

implements  $\mathcal{F}_{\text{p-ot}}$ . In particular, digging deeper into the protocol, this ensures that we are invoking a *single* ZK protocol for all the parallel copies of the parallel OT, binding the executions together.

**Achieving Extractability Using OT with Unbounded Simulation.** Interestingly, we show that OT with (even 2-sided) unbounded simulation plus zero-knowledge is sufficient for constructing extractable commitments, which when combined with zero-knowledge again as above gives an implementation of  $\mathcal{F}_{\text{so-com}}$  in the sequential composition setting in the plain model.

The initial idea is to convert the power of simulation into the power of extraction via two-party computation, and sketched below.

**Initial Idea for Extractable Commitment:** The sender  $\text{com.S}$  has  $\mu \in \{0, 1\}^n$ .

1. **Trapdoor setup:** The receiver  $\text{com.R}$  sends a commitment  $c$  of a statistically binding commitment scheme  $\text{com}$ , and gives a zero-knowledge proof that  $c$  commits to 0.
2. **Conditional Disclosure of Secret (CDS):**  $\text{com.S}$  and  $\text{com.R}$  run a two-party computation protocol implementing the CDS functionality  $\mathcal{F}_{\text{cds}}$  for the language  $\mathcal{L}_{\text{com}} = \{(c', b') : \exists r' \text{ s.t. } c' = \text{com}(b'; r')\}$ , where the CDS functionality  $\mathcal{F}_{\text{cds}}$  for  $\mathcal{L}_{\text{com}}$  is defined as below:

$\mathcal{F}_{\text{cds}}$  : Sender input  $(x, \mu)$ , Receiver input  $w$

Sender has no output, Receiver outputs  $x$  and  $\mu' = \begin{cases} \mu & \text{if } \mathcal{R}_{\mathcal{L}_{\text{com}}}(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$

$\text{com.S}$  acts as the CDS sender using input  $(x = (c, 1), \mu)$  while  $\text{com.R}$  acts as the CDS receiver using witness  $w = 0$ .

It may seem paradoxical that we try to implement commitments using the much more powerful tool of two-party computation. The *key observation* is that the hiding and extractability of the above commitment protocol only relies on the *input-indistinguishability property* of the CDS protocol, which is *implied by unbounded simulation*.

- Hiding: A commitment to  $\mu$  can be simulated by simply committing to  $0^n$  honestly, that is, using  $(x = (c, 1), 0^n)$  as the input to the CDS. The simulation is indistinguishable as the soundness of ZK argument guarantees that  $c$  must be a commitment to 0 and hence the CDS statement  $(c, 1)$  is false and should always produce  $\mu' = \perp$ . Therefore, the unbounded-simulation security of the CDS protocol implies that it is indistinguishable to switch the sender's input from  $\mu$  to  $0^n$ .
- Extraction: To efficiently extract from a malicious sender  $\text{com.S}^*$ , the idea (which however suffers from a problem described below) is to let the simulator-extractor  $\text{com.SimS}$  set up a trapdoor by committing to 1 (instead of 0) and simulate the ZK argument; it can then use the decommitment (call it  $r$ ) to 1 as a valid witness to obtain the committed value from the output of the CDS protocol. Here, the unbounded-simulation security of CDS again implies that

interaction with an honest receiver who uses  $w = 0$  is indistinguishable from that with `com.SimS` who uses  $w = r$  as `com.S`<sup>\*</sup> receives no output via CDS.

The advantage of CDS with unbounded simulation is that it can be implemented using OT with unbounded simulation: Following the work of [42, 43, 61], post-quantum MPC protocols exist in the  $\mathcal{F}_{\text{ot}}$ -hybrid model, and instantiating them with the unbounded-simulation OT yields unbounded simulation MPC and therefore CDS.

NP-VERIFIABILITY AND THE LACK OF IT. Unfortunately, the above attempt has several problems: how do we show that the commitment is binding? how to decommit? and how to guarantee that the extracted value agrees with the value that can be decommitted to? We can achieve binding by having the sender additionally commit to  $\mu$  using a statistically binding commitment scheme `com`, and send the corresponding decommitment in the decommitment phase. However, to guarantee that the extractor would extract the same string  $\mu$  from CDS, we need a way to verify that the same  $\mu$  is indeed used by the CDS sender. Towards this, we formalize a verifiability property of a CDS protocol:

*A CDS protocol is verifiable if*

- The honest CDS sender `cds.S` additionally outputs  $(x, \mu)$  and a “proof”  $\pi$  (on a special output tape) at the end of the execution.
- There is an efficient *classical* verification algorithm  $\text{Ver}(\tau, x, \mu, \pi)$  that verifies the proof, w.r.t. the transcript  $\tau$  of the *classical* messages exchanged in the CDS protocol.
- Binding: No malicious sender `cds.S`<sup>\*</sup> after interacting with an honest receiver `cds.R(w)` can output  $(x, \mu, \pi)$ , such that the following holds simultaneously: (a)  $\text{Ver}(\tau, x, \mu, \pi) = 1$ , (b) `cds.R` did not abort, and (c) `cds.R` outputs  $\mu'$  inconsistent with the inputs  $(x, \mu)$  and  $w$ , that is,  $\mu' \neq \begin{cases} \mu & \text{if } \mathcal{R}_{\mathcal{L}}(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$

We observe first that classical protocols with perfect correctness have verifiability for free: The proof  $\pi$  is simply the sender’s random coins  $r$ , and the verification checks if the honest sender algorithm with input  $(x, \mu)$  and random coins  $r$  produces the same messages as in the transcript  $\tau$ . If so, perfect correctness guarantees that the output of the receiver must be consistent with  $x, \mu$ . However, verifiability cannot be taken for granted in the  $\mathcal{F}_{\text{ot}}$  hybrid model or in the quantum setting. In the  $\mathcal{F}_{\text{ot}}$  hybrid model, it is difficult to write down an NP-statement that captures consistency as the OT input is *not* contained in the protocol transcript and is unconstrained by it. In the quantum setting, protocols use quantum communication, and consistency cannot be expressed as an NP-statement. Take the BBCS protocol as an example, the OT receiver receives from the sender  $\ell$  qubits and measures them locally; there is no way to “verify” this step in NP.

**Implementing Verifiable CDS.** To overcome the above challenge, we implement a verifiable CDS protocol in the  $\mathcal{F}_{\text{p-ot}}$  hybrid model assuming only post-quantum one-way functions. We develop this protocol in a few steps below.

Let's start by understanding why the standard two-party computation protocol is not verifiable. The protocol proceeds as follows: First, the sender **cds.S** locally garbles a circuit computing the following function into  $\widehat{G}$  with labels  $\{\ell_b^j\}_{j \in [m], b \in \{0,1\}}$  where  $m = |w|$ :

$$G_{x,\mu}(w) = \mu' = \begin{cases} \mu & \text{if } \mathcal{R}_{\mathcal{L}}(x, w) = 1 \\ \perp & \text{otherwise} \end{cases} \quad (1)$$

Second, **cds.S** sends the pairs of labels  $\{\ell_0^j, \ell_1^j\}_j$  via  $\mathcal{F}_{\text{p-ot}}$ . The receiver **cds.R** on the other hand chooses  $\{w_j\}_j$  to obtain  $\{\tilde{\ell}_{w_j}^j\}_j$ , and evaluates  $\widehat{G}$  with these labels to obtain  $\mu'$ . This protocol is not NP-verifiable because consistency between the labels of the garbled circuit and the sender's inputs to  $\mathcal{F}_{\text{p-ot}}$  cannot be expressed as a NP statement.

To fix the problem, we devise a way for the receiver to verify the OT sender's strings. Let **cds.S** additionally commit to all the labels  $\{c_b^j = \text{com}(\ell_b^j; r_b^j)\}_{j,b}$  and the message  $c = \text{com}(\mu; r)$  and prove in ZK that  $\widehat{G}$  is consistent with the labels and message committed in the commitments, as well as the statement  $x$ . Moreover, the sender sends both the labels and decommitments  $\{(\ell_0^j, r_0^j), (\ell_1^j, r_1^j)\}_j$  via  $\mathcal{F}_{\text{p-ot}}$ . The receiver after receiving  $\{\tilde{\ell}_{w_j}^j, \tilde{r}_{w_j}^j\}_j$  can now verify their correctness by verifying the decommitment w.r.t.  $c_{w_j}^j$ , and aborts if verification fails. This gives the following new protocol:

**A Verifiable but Insecure CDS Protocol:** The sender **cds.S** has  $(x, \mu)$  and the receiver **cds.R** has  $w$ .

1. **Sender's Local Preparation:** **cds.S** generate a garbled circuits  $\widehat{G}$  for the circuit computing  $G_{x,\mu}$  (Equation (1)), with labels  $\{\ell_b^{i,j}\}_{j,b}$ . Moreover, it generates commitments  $c = \text{com}(\mu, r)$  and  $c_b^j = \text{com}(\ell_b^j; r_b^j)$  for every  $j, b$ .
2. **OT:** **cds.S** and **cds.R** invoke  $\mathcal{F}_{\text{p-ot}}$ . For every  $j$ , the sender sends  $(\ell_0^j, r_0^j), (\ell_1^j, r_1^j)$ , and the receiver chooses  $w_j$  and obtains  $(\tilde{\ell}_{w_j}^j, \tilde{r}_{w_j}^j)$ .
3. **Send Garbled Circuit and Commitments:** **cds.S** sends  $\widehat{G}$ ,  $c$ , and  $\{c_b^j\}_{j,b}$  and proves via a ZK protocol that they are all generated consistently w.r.t. each other and  $x$ .
4. **Receiver's Checks:** **cds.R** aborts if ZK is not accepting, or if for some  $j$ ,  $c_{w_j}^j \neq \text{com}(\tilde{\ell}_{w_j}^j, \tilde{r}_{w_j}^j)$ . Otherwise, it evaluates  $\widehat{G}$  with the labels and obtain  $\mu' = G_{x,\mu}(w)$ .

We argue that this protocol is NP-verifiable. The sender's proof is simply the decommitment  $r$  of  $c$ , and  $\text{Ver}(\tau, (x, \mu), r) = 1$  iff  $r$  is a valid decommitment to  $\mu$  of the commitment  $c$  contained in the transcript  $\tau$ . To show the binding property, consider an interaction between a cheating sender **cds.S\*** and **cds.R(w)**. Suppose **cds.R** does not abort, it means that 1) the ZK argument is accepting and hence  $\widehat{G}$  must be consistent with  $x, \{c_b^j\}, c$ , and 2) the receiver obtains the labels committed in  $c_{w_j}^j$ 's. Therefore, evaluating the garbled circuit with these labels must produce  $\mu' = G_{x,\mu}(w)$  for the  $\mu$  committed to in  $c$ .

Unfortunately, the checks that the receiver performs render the protocol insecure. A malicious sender **com.S\*** can launch the so-called selective abort attack

to learn information of  $w$ . For instance, to test if  $w_1 = 0$  or not, it replaces  $\ell_0^1$  with zeros. If  $w_1 = 0$  the honest receiver would abort; otherwise, it proceeds normally.

**THE FINAL PROTOCOL.** To circumvent the selective abort attack, we need a way to check the validity of sender's strings that is independent of  $w$ . Our idea is to use a variant of cut-and-choose. Let  $\text{cds.S}$  create  $2\lambda$  copies of garbled circuits and commitments to their labels,  $\{\widehat{G}^i\}_{i \in [2\lambda]}$  and  $\{c_b^{i,j} = \text{com}(\ell_b^{i,j}; r_b^{i,j})\}_{i,j,b}$  and prove via a ZK protocol that they are all correctly generated w.r.t. the same  $c$  and  $x$ . Again,  $\text{cds.S}$  sends the labels and decommitment via  $\mathcal{F}_{\text{p-ot}}$ , but  $\text{cds.R}$  does not choose  $w$  universally in all copies. Instead, it secretly samples a random subset  $\Lambda \in [2\lambda]$  by including each  $i$  with probability  $1/2$ ; for copy  $i \in \Lambda$ , it chooses random string  $s^i \leftarrow \{0, 1\}^m$  and obtains  $\{\tilde{\ell}_{s_j^i}^{i,j}, \tilde{r}_{s_j^i}^{i,j}\}_j$ , whereas for copy  $i \notin \Lambda$ , it chooses  $w$  and obtains  $\{\tilde{\ell}_{w_j}^{i,j}, \tilde{r}_{w_j}^{i,j}\}_j$ . Now, in the checking step,  $\text{cds.R}$  only verifies the validity of  $\{\tilde{\ell}_{s_j^i}^{i,j}, \tilde{r}_{s_j^i}^{i,j}\}_{i \in \Lambda, j}$  received in copies in  $\Lambda$ . Since the check is now completely independent of  $w$ , it circumvents the selective abort attack.

Furthermore, NP-verifiability still holds. The key point is that if the decommitments  $\text{cds.R}$  receives in copies in  $\Lambda$  are all valid, with overwhelming probability, the number of *bad copies* where the OT sender's strings are not completely valid is bounded by  $\lambda/4$ . Hence, there must exist a copy  $i \notin \Lambda$  where  $\text{cds.R}$  receives the right labels  $\ell_{w_j}^{i,j}$  committed to in  $c_{w_j}^{i,j}$ .  $\text{cds.R}$  can then evaluate  $\widehat{G}^i$  to obtain  $\mu'$ . By the same argument as above,  $\mu'$  must be consistent with the  $(x, \mu)$  and  $w$ , for  $\mu$  committed in  $c$ , and NP-verifiability follows. The final protocol is described in Fig. 3.

**Organization of the Paper.** We review the quantum stand-alone security model introduced by [36] in Sect. 2. In section Sect. 3, we construct a quantum parallel-OT protocol with one-sided, unbounded simulation. In more detail, we review in Sect. 3.1 the quantum OT protocol from [9] based on ideal commitments with selective opening security. Then in Sect. 3.2, we show how to boost it to construct a *parallel* OT protocol from the same assumptions. And finally, we provide a classical implementation of the commitment scheme with selective opening security in Sect. 3.3 which gives us ideal/real security except with unbounded receiver simulation. This result will be fed into our main technical contribution in Sect. 4 where we show how to construct extractable commitments from unbounded-simulation parallel-OT. In Sect. 4.2, we show how to construct (the intermediate primitive of) CDS from parallel-OT and one-way functions, and then in Sect. 4.3 we construct extractable commitments from CDS. Finally, in Sect. 5 we lift our results to achieve quantum protocols for multi-party (quantum) computation from one-way functions.

## 2 Quantum Stand-Alone Security Model

We adopt the quantum stand-alone security model from the work of Hallgren, Smith and Song [36], tailored to the two-party setting.



Let  $\mathcal{F}$  denote a *functionality*, which is a classical interactive machine specifying the instructions to realize a cryptographic task. A two-party protocol  $\Pi$  consists of a pair of quantum interactive machines  $(A, B)$ . We call a protocol *efficient* if  $A$  and  $B$  are both quantum poly-time machines. If we want to emphasize that a protocol is classical, i.e., all computation and all messages exchanged are classical, we then use lower-case letters (e.g.,  $\pi$ ). Finally, an adversary  $\mathcal{A}$  is another quantum interactive machine that intends to attack a protocol.

When a protocol  $\Pi = (A, B)$  is executed under the presence of an adversary  $\mathcal{A}$ , the state registers are initialized by a security parameter  $1^\lambda$  and a joint quantum state  $\sigma_\lambda$ . Adversary  $\mathcal{A}$  gets activated first, and may either **deliver** a message, i.e., instructing some party to read the proper segment of the network register, or **corrupt** a party. We assume all registers are authenticated so that  $\mathcal{A}$  cannot modify them, but otherwise  $\mathcal{A}$  can schedule the messages to be delivered in any arbitrary way. If  $\mathcal{A}$  corrupts a party, the party passes all of its internal state to  $\mathcal{A}$  and follows the instructions of  $\mathcal{A}$ . Any other party, once receiving a message from  $\mathcal{A}$ , gets activated and runs its machine. At the end of one round, some message is generated on the network register. Adversary  $\mathcal{A}$  is activated again and controls message delivery. At some round, the party generates some output and terminates.

We view  $\Pi$  and  $\mathcal{A}$  as a whole and model the composed system as another QIM, call it  $M_{\Pi, \mathcal{A}}$ . Then executing  $\Pi$  in the presence of  $\mathcal{A}$  is just running  $M_{\Pi, \mathcal{A}}$  on some input state, which may be entangled with a reference system available to a distinguisher.

**Protocol emulation and secure realization of a functionality.** A secure protocol is supposed to “emulate” an idealized protocol. Consider two protocols  $\Pi$  and  $\Gamma$ , and let  $M_{\Pi, \mathcal{A}}$  be the composed machine of  $\Pi$  and an adversary  $\mathcal{A}$ , and  $M_{\Gamma, \mathcal{S}}$  be that of  $\Gamma$  and another adversary  $\mathcal{S}$ . Informally,  $\Pi$  emulates  $\Gamma$  if the two machines  $M_{\Pi, \mathcal{A}}$  and  $M_{\Gamma, \mathcal{S}}$  are indistinguishable.

It is of particular interest to emulate an *ideal-world* protocol  $\tilde{\Pi}_{\mathcal{F}}$  for a functionality  $\mathcal{F}$  which captures the security properties we desire. In this protocol, two (dummy) parties  $\tilde{A}$  and  $\tilde{B}$  have access to an additional “trusted” party that implements  $\mathcal{F}$ . We abuse notation and call the trusted party  $\mathcal{F}$  too. Basically  $\tilde{A}$  and  $\tilde{B}$  invoke  $\mathcal{F}$  with their inputs, and then  $\mathcal{F}$  runs on the inputs and sends the respective outputs back to  $\tilde{A}$  and  $\tilde{B}$ . An execution of  $\tilde{\Pi}$  with an adversary  $\mathcal{S}$  is as before, except that  $\mathcal{F}$  cannot be corrupted. We denote the composed machine of  $\mathcal{F}$  and  $\tilde{\Pi}_{\mathcal{F}}$  as  $M_{\mathcal{F}, \mathcal{S}}$ .

**Definition 2.1 (Computationally Quantum-Stand-Alone Emulation).** *Let  $\Pi$  and  $\Gamma$  be two poly-time protocols. We say  $\Pi$  computationally quantum-stand-alone (C-QSA) emulates  $\Gamma$ , if for any poly-time QIM  $\mathcal{A}$  there exists a poly-time QIM  $\mathcal{S}$  such that  $M_{\Pi, \mathcal{A}} \approx_{qc} M_{\Gamma, \mathcal{S}}$ .*

**Definition 2.2 (C-QSA Realization of a Functionality).** *Let  $\mathcal{F}$  be a poly-time two-party functionality and  $\Pi$  be a poly-time two-party protocol. We say  $\Pi$  computationally quantum-stand-alone realizes  $\mathcal{F}$ , if  $\Pi$  C-QSA emulates  $\tilde{\Pi}_{\mathcal{F}}$ .*

Namely, for any poly-time  $\mathcal{A}$ , there is a poly-time  $\mathcal{S}$  such that  $M_{\Pi, \mathcal{A}} \approx_{qc} M_{\mathcal{F}, \mathcal{S}}$ .

**Definition 2.3 (Statistically Quantum-Stand-Alone Emulation).** Let  $\Pi$  and  $\Gamma$  be two poly-time protocols. We say  $\Pi$  statistically quantum-stand-alone (S-QSA) emulates  $\Gamma$ , if for any QIM  $\mathcal{A}$  there exists an QIM  $\mathcal{S}$  that runs in poly-time of that of  $\mathcal{A}$ , such that  $M_{\Pi, \mathcal{A}} \approx_{\diamond} M_{\Gamma, \mathcal{S}}$ .

We assume *static* corruption only in this work, where the identities of corrupted parties are determined before protocol starts. The definitions above consider computationally bounded (poly-time) adversaries, including simulators. Occasionally, we will work with *inefficient* simulators, which we formulate as unbounded simulation of corrupted party  $P$ .

**Definition 2.4 (Unbounded Simulation of Corrupted  $P$ ).** Let  $\Pi$  and  $\Gamma$  be two poly-time protocols. For any poly-time QIM  $\mathcal{A}$  corrupting party  $P$ , we say that  $\Pi$  C-QSA-emulates  $\Gamma$  against corrupted  $P$  with unbounded simulation, if there exists a QIM  $\mathcal{S}$  possibly unbounded such that  $M_{\Pi, \mathcal{A}} \approx_{qc} M_{\Gamma, \mathcal{S}}$ .

## 2.1 Modular Composition Theorem

It's shown that protocols satisfying the definitions of stand-alone emulation admit a modular composition [36]. Specifically, let  $\Pi$  be a protocol that uses another protocol  $\Gamma$  as a subroutine, and let  $\Gamma'$  be a protocol that QSA emulates  $\Gamma$ . We define the *composed* protocol, denoted  $\Pi^{\Gamma/\Gamma'}$ , to be the protocol in which each invocation of  $\Gamma$  is replaced by an invocation of  $\Gamma'$ . We allow multiple calls to a subroutine and also using multiple subroutines in a protocol  $\Pi$ . **However, quite importantly, we require that at any point, only one subroutine call be in progress.** This is more restrictive than the “network” setting, where many instances and subroutines may be executed *concurrently*.

In a *hybrid* model, parties can make calls to an ideal-world protocol  $\tilde{\Pi}_{\mathcal{G}}$  of some functionality  $\mathcal{G}$ <sup>4</sup>. We call such a protocol a  $\mathcal{G}$ -*hybrid* protocol, and denote it  $\Pi^{\mathcal{G}}$ . The execution of a hybrid-protocol in the presence of an adversary  $\mathcal{A}$  proceeds in the usual way. Assume that we have a protocol  $\Gamma$  that realizes  $\mathcal{G}$  and we have designed a  $\mathcal{G}$ -hybrid protocol  $\Pi^{\mathcal{G}}$  realizing another functionality  $\mathcal{F}$ . Then the composition theorem allows us to treat sub-protocols as equivalent to their ideal versions.

If the secure emulation involves unbounded simulation against a party, the proof in [36] can be extended to show that the composed protocol also emulates with unbounded simulation against the corresponding corrupted party.

**Theorem 2.1 (Modular Composition).** *All of the following holds.*

- Let  $\Pi$ ,  $\Gamma$  and  $\Gamma'$  be two-party protocols such that  $\Gamma'$  C-QSA-emulates  $\Gamma$ , then  $\Pi^{\Gamma/\Gamma'}$  C-QSA emulates  $\Pi$ . If  $\Gamma'$  C-QSA emulates  $\Gamma$  against corrupted  $P$  with unbounded simulation, then  $\Pi^{\Gamma/\Gamma'}$  C-QSA emulates against corrupted  $P$  with unbounded simulation.

<sup>4</sup> In contrast, we call it the *plain model* if no such trusted set-ups are available.

- Let  $\mathcal{F}$  and  $\mathcal{G}$  be poly-time functionalities. Let  $\Pi^{\mathcal{G}}$  be a  $\mathcal{G}$ -hybrid protocol that  $\mathbf{C}$ -QSA realizes  $\mathcal{F}$ , and  $\Gamma$  be a protocol that  $\mathbf{C}$ -QSA realizes  $\mathcal{G}$ , then  $\Pi^{\mathcal{G}/\Gamma}$   $\mathbf{C}$ -QSA realizes  $\mathcal{F}$ . If  $\Gamma$   $\mathbf{C}$ -QSA realizes  $\mathcal{G}$  against corrupted  $P$  with unbounded simulation then  $\Pi^{\mathcal{G}/\Gamma}$   $\mathbf{C}$ -QSA realizes  $\mathcal{F}$  against corrupted  $P$  with unbounded simulation.

### 3 Parallel OT with Unbounded Simulation from OWF

The goal of this section is to prove the following theorem.

**Theorem 3.1.** *Assuming the existence of  $pqOWF$ , there exists a protocol  $\Pi_{p-ot}$  that  $\mathbf{C}$ -QSA-emulates  $\mathcal{F}_{p-ot}$  with unbounded simulation against a malicious receiver.*

We prove this theorem as follows. In Sect. 3.1, we review the protocol of [9] that implies stand-alone-secure OT in  $\mathcal{F}_{so-com}$ -hybrid model. Then, in Sect. 3.2, we show how to build  $\mathcal{F}_{p-ot}$  from  $\mathcal{F}_{so-com}$ . Finally in Sect. 3.3, we construct  $\mathcal{F}_{so-com}$  with unbounded simulation against malicious sender.

#### 3.1 Stand-Alone-Secure OT in $\mathcal{F}_{so-com}$ -hybrid Model

In this section we present the quantum OT protocol assuming a selective opening-secure commitment scheme, that is, in the  $\mathcal{F}_{so-com}$  hybrid model. We would like to stress that the results in this section are not novel; they consist of a straightforward adaptation of previous results [9, 20, 61] to our setting/language, and our goal in this presentation is to provide a self-contained proof of its security. We describe the protocol  $\Pi_{qot}$  in Sect. 1.1 and we have the following.

**Theorem 3.2.**  *$\Pi_{qot}$   $\mathbf{C}$ -QSA-realizes  $\mathcal{F}_{ot}$  in the  $\mathcal{F}_{so-com}$  hybrid model.*

#### 3.2 Parallel Repetition for Protocols with Straight-Line Simulation

We show now that if  $\pi$  implements  $\mathcal{F}$  in the  $\mathcal{G}$ -hybrid model with an (efficient/unbounded) *straight-line* simulator, then a parallel repetition of  $\pi$ , denoted  $\pi^{\parallel}$  implements  $\mathcal{F}^{\parallel}$  in the  $\mathcal{G}^{\parallel}$ -hybrid model with an (efficient/unbounded) simulator. As a corollary, we get that a parallel repetition of the  $\mathcal{F}_{ot}$  protocol from the previous section is a secure implementation of parallel OT in the  $\mathcal{F}_{so-com}$  hybrid model.

**Theorem 3.3 (Parallel Repetition).** *Let  $\mathcal{F}$  and  $\mathcal{G}$  be two-party functionalities and let  $\pi$  be a secure implementation of  $\mathcal{F}$  in the  $\mathcal{G}$ -hybrid model with a straight-line simulator. Then,  $\pi^{\parallel}$  is a secure implementation of  $\mathcal{F}^{\parallel}$  in the  $\mathcal{G}^{\parallel}$ -hybrid model with straight-line simulation as well.*

**Corollary 3.1.** *The parallel repetition of any protocol that  $\mathbf{C}$ -QSA-realizes  $\mathcal{F}_{ot}$  in the  $\mathcal{F}_{so-com}$ -hybrid model with a straight-line simulator achieves  $\mathcal{F}_{p-ot}$  in the  $\mathcal{F}_{so-com}$ -hybrid model.*

### 3.3 Implementing $\mathcal{F}_{\text{so-com}}$ with Unbounded Simulation

In this section we provide an implementation of  $\mathcal{F}_{\text{so-com}}$  from Naor's commitment scheme and ZK protocols. Our protocol  $\Pi_{\text{so-com}}$  is described in Fig. 1 and we prove the following result.

**Theorem 3.4.** *Assuming the existence of pqOWF,  $\Pi_{\text{so-com}}$  C-QSA-realizes  $\mathcal{F}_{\text{so-com}}$  with unbounded simulation against malicious committer.*

We prove Theorem 3.4 by showing security against malicious committer with unbounded simulator in Lemma 3.1 and security against malicious receiver in Lemma 3.2.

**Parties:** The committer  $C$  and the receiver  $R$ .

**Inputs:**  $C$  gets  $k$   $\ell$ -bit strings  $m_1, \dots, m_k$  and  $R$  gets a subset  $I \subseteq [k]$  of messages to be decommitted

#### Commitment Phase

1.  $R$  sends  $\rho$  for Naor's commitment scheme
2. For  $i \in [k]$ ,  $C$  generates the commitments  $c_i = \text{com}_\rho(m_i, r_i)$ , where  $r_i$  is some private randomness.
3.  $C$  sends  $c_1, \dots, c_k$  to  $R$

#### Decommitment Phase

1.  $R$  sends  $I$  to  $C$
2.  $C$  sends  $(m_i)_{i \in I}$  to  $R$  and they run a ZK protocol to prove that there exists  $((\tilde{m}_i)_{i \notin I}, (r_i)_{i \in [k]})$  such that  $c_i = \text{com}_\rho(\tilde{m}_i, r_i)$

**Fig. 1.** Protocol for selective-opening commitment scheme  $\Pi_{\text{so-com}}$ .

**Lemma 3.1.** *Assuming the existence of pqOWF,  $\Pi_{\text{so-com}}$  C-QSA-emulates  $\mathcal{F}_{\text{so-com}}$  against corrupted committer  $\mathcal{A}$  with unbounded simulation.*

*Proof.* The unbounded simulator  $\mathcal{S}$  works as follows:

1. In the commitment phase,  $\mathcal{S}$  runs the honest protocol with  $\mathcal{A}$  and when receives the commitments  $\hat{c}_1, \dots, \hat{c}_k$  from  $\mathcal{A}$  and  $\mathcal{S}$  finds the messages  $\hat{m}_1, \dots, \hat{m}_k$  by brute force. If there is a  $\hat{c}_i$  that does not decommit to any message or decommits to more than one message  $\mathcal{S}$  aborts. Finally,  $\mathcal{S}$  inputs  $\hat{m}_1, \dots, \hat{m}_k$  to  $\mathcal{F}_{\text{so-com}}$
2. In the Decommitment phase,  $\mathcal{S}$  receives  $I$  from  $\mathcal{F}_{\text{so-com}}$ , forwards it to  $\mathcal{A}$ .  $\mathcal{S}$  receives  $(\tilde{m}_i)_{i \in I}$  from  $\mathcal{A}$  runs the honest verifier in the ZK protocol with  $\mathcal{A}$ , and rejects iff the ZK rejects or if for any  $i \in I$ ,  $\hat{m}_i \neq \tilde{m}_i$ .

The proof follows the statistically-binding property of Naor’s commitment scheme, so we can ignore commitments that open to more than one message, and by the ZK soundness property, which ensures that, up to negligible probability, if the commitments are not well-formed or if the sender tries to open then to a different value, both the simulator and the original receiver abort.

Due to space restrictions, we leave the details to the full version of our paper.

We now show security against malicious receiver.

**Lemma 3.2.** *Assuming the existence of  $pqOWF$ ,  $\Pi_{so-com}$  C-QSA-realizes  $\mathcal{F}_{so-com}$  against corrupted receiver  $\mathcal{A}$ .*

*Proof.* The simulator  $\mathcal{S}$  works as follows:

1. In the commitment phase,  $\mathcal{S}$  sends  $c_i = \text{com}_\rho(0, r_i)$  to  $\mathcal{A}$
2. In the decommitment phase,  $\mathcal{S}$  receives  $I$  from  $\mathcal{A}$ , uses it as input of  $\mathcal{F}_{so-com}$ .  $\mathcal{S}$  receives back the messages  $(m_i)_{i \in I}$ , sends them to  $\mathcal{A}$  and runs the ZK simulator of the proof that  $(c_i)_{i \in I}$  open to  $(m_i)_{i \in I}$  and that  $(c_i)_{i \notin I}$  are valid commitments.

The fact that  $M_{\Pi_{so-com}, \mathcal{A}} \approx_{qc} M_{\mathcal{F}_{so-com}, \mathcal{S}}$  follows from the computational zero-knowledge of the protocol and the computationally-hiding property of Naor’s commitment scheme.

## 4 Extractable Commitment from Unbounded Simulation OT

In this section, we construct an extractable commitment scheme using the unbounded simulation OT from Sect. 3. We do this in two steps. First, we define a new primitive, namely *verifiable* conditional disclosure of secrets (vCDS) in Sect. 4.1, and we construct a (unbounded simulation) vCDS protocol in Sect. 4.2 from the unbounded simulation OT. We then show how to use vCDS to construct an extractable commitment protocol that implements  $\mathcal{F}_{so-com}$  with efficient simulators in Sect. 4.3.

### 4.1 Verifiable Conditional Disclosure of Secrets (vCDS)

We define the primitive of (verifiable) conditional disclosure of secrets. Conditional disclosure of secrets [31] (CDS) for an NP-language  $\mathcal{L}$  is a two-party protocol where a sender (denoted  $\text{cds.S}$ ) and a receiver (denoted  $\text{cds.R}$ ) have a common input  $x$ , the sender has a message  $\mu$ , and the receiver (purportedly) has a witness  $w$  for the NP-relation  $R_{\mathcal{L}}$ . At the end of the protocol,  $\text{cds.R}$  gets  $\mu$  if  $R_{\mathcal{L}}(x, w) = 1$  and  $\perp$  otherwise, and the sender gets nothing. In a sense, this can be viewed as a *conditional* version of oblivious transfer, or as an interactive version of witness encryption.

The CDS functionality is defined in Fig. 2. We will construct a protocol  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  that securely realizes the CDS functionality in the quantum

**The Conditional Disclosure of Secret (CDS) Functionality  $\mathcal{F}_{CDS}$  for an NP language  $\mathcal{L}$ .**

Security Parameter:  $\lambda$ .

Parties: Sender  $S$  and Receiver  $R$ , adversary  $\mathcal{A}$ .

**Sender Query:**  $\mathcal{F}_{CDS}$  receives  $(\text{Send}, \text{sid}, (x, \mu))$  from  $S$ , where  $x \in \mathcal{L} \cap \{0, 1\}^{n_1(\lambda)}$  and  $m \in \{0, 1\}^{n_2(\lambda)}$  for polynomials  $n_1$  and  $n_2$ , records  $(\text{sid}, (x, \mu))$  and sends  $(\text{Input}, \text{sid}, x)$  to  $R$  and  $\mathcal{A}$ .

$\mathcal{F}_{CDS}$  ignores further send messages from  $S$  with  $\text{sid}$ .

**Receiver Query:**  $\mathcal{F}_{CDS}$  receives  $(\text{Witness}, \text{sid}, w)$  from party  $R$ , where  $w \in \{0, 1\}^{m(\lambda)}$  for a polynomial  $m$ .  $\mathcal{F}_{CDS}$  ignores the message if no  $(\text{sid}, \star)$  was recorded. Otherwise  $\mathcal{F}_{CDS}$  sends  $(\text{Open}, \text{sid}, x, \mu')$  to  $R$  where

$$\mu' = \begin{cases} \mu & \text{if } \mathcal{R}_{\mathcal{L}}(x, w) = 1 \\ \perp & \text{if } \mathcal{R}_{\mathcal{L}}(x, w) = 0 \end{cases}$$

$\mathcal{F}_{CDS}$  sends  $(\text{Open}, \text{sid}, x)$  to  $\mathcal{A}$  and ignores further messages from  $R$  with  $\text{sid}$ .

**Fig. 2.** The Conditional Disclosure of Secrets (CDS) functionality

stand-alone model. We will consider protocols with either efficient or unbounded simulators.

**Verifiability.** We will, in addition, also require the CDS protocol to be *verifiable*. Downstream, when constructing our extractable commitment protocol in Sect. 4.3, we want to be able to prove consistency of the transcript of a CDS sub-protocol. It is not a-priori clear how to do this since the CDS protocol we construct will either live in the OT-hybrid model, in which case the OT input is *not* contained in the protocol transcript and is unconstrained by it; or it uses quantum communication, in which case, again consistency cannot be expressed as an NP-statement.

**Definition 4.1 (Verifiability).** *Let  $\mathcal{L}$  be an NP language, and  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  be a CDS protocol between a sender  $\text{cds.S}$  and a receiver  $\text{cds.R}$ .  $\Pi$  is verifiable (w.r.t.  $\text{cds.S}$ ) if there is a polynomial time classical algorithm  $\text{Ver}$ , such that, the following properties are true:*

**Correctness:** *For every  $(x, \mu)$  and every  $w$ ,  $\text{cds.S}(x, \mu)$  after interacting with  $\text{cds.R}(w)$ , outputs on a special output tape a proof  $\pi$ , such that,  $\text{Ver}(\tau, x, \mu, \pi) = 1$  where  $\tau$  is the transcript of classical messages exchanged in the interaction.*

**Binding:** *For every  $\lambda \in \mathbb{N}$ , every (potentially unbounded) adversary  $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ , every sequence of witnesses  $\{w_\lambda\}_\lambda$ , the probability that  $\mathcal{A}_\lambda$  wins in the following experiment is negligible.*

- $\mathcal{A}_\lambda$  after interacting with  $\text{cds.R}(1^\lambda, w)$ , outputs  $(x, \mu, \pi)$ . Let  $\tau$  be the transcript of classical messages exchanged in the interaction.



- $\mathcal{A}_\lambda$  wins if (a)  $\text{Ver}(\tau, x, \mu, \pi) = 1$ , (b)  $\text{cds.R}$  did not abort, and (c)  $\text{cds.R}$  outputs  $\mu'$  inconsistent with inputs  $(x, \mu)$  and  $w$ , that is,

$$\mu' \neq \begin{cases} \mu & \text{if } \mathcal{R}_\mathcal{L}(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$$

**Definition 4.2 (Verifiable CDS).** Let  $\mathcal{L}$  be an NP language, and  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  be a protocol between a sender  $\text{cds.S}$  and a receiver  $\text{cds.R}$ .  $\Pi$  is a verifiable CDS protocol if (a) it C-QSA-emulates  $\mathcal{F}_{\text{cds}}$  with an efficient simulator; and (b) it is verifiable according to Definition 4.1.

## 4.2 CDS Protocol from Unbounded Simulation OT

**Theorem 4.1.** Assume the existence of pqOWF. For every NP language  $\mathcal{L}$ , there is a verifiable CDS protocol  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  that C-QSA-emulates  $\mathcal{F}_{\text{cds}}$  for  $\mathcal{L}$  in the  $\mathcal{F}_{p\text{-ot}}$  hybrid model.

**Corollary 4.1.** Assume the existence of pqOWF, and a protocol that C-QSA-emulates  $\mathcal{F}_{p\text{-ot}}$  with unbounded simulation. Then, for every NP language  $\mathcal{L}$ , there is a verifiable CDS protocol  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  that C-QSA-emulates  $\mathcal{F}_{\text{cds}}$  for  $\mathcal{L}$  with unbounded simulation.

**Proof of Theorem 4.1.** The verifiable CDS protocol is described in Fig. 3. The protocol uses Naor’s classical statistically binding commitment protocol, Yao’s garbled circuits, and post-quantum zero knowledge proofs, all of which can be implemented from pqOWF. For a more detailed description of these ingredients, see the full version of our paper.

In Lemma 4.1, we show that the protocol has an efficient simulator for a corrupted receiver, and in Lemma 4.2, an efficient simulator for a corrupted sender (both in the OT hybrid model). Lemma 4.3 shows that the protocol is verifiable.  $\square$

**Lemma 4.1.** There is an efficient simulator against a malicious receiver.

*Proof.* The simulator  $\mathcal{S}$  interacts with  $\text{cds.R}^*$ , receives a string  $\rho$  from  $\text{cds.R}^*$  in Step 1, and intercepts the OT queries  $(\sigma^1, \dots, \sigma^{2^\lambda})$  in Step 4.

- **Case 1.**  $R_\mathcal{L}(x, \sigma^i) = 1$  for some  $i$ . Send (Witness,  $\text{sid}, \sigma^i$ ) to the CDS functionality and receive  $\mu$ . Simulate the rest of the protocol honestly using the CDS sender input  $(x, \mu)$ .
- **Case 2.**  $R_\mathcal{L}(x, \sigma^i) = 0$  for all  $i$ . Simulate the rest of the protocol honestly using the CDS sender input  $(x, 0)$ .

We now show, through a sequence of hybrids, that this simulator produces a view that is computationally indistinguishable from that in the real execution of  $\text{cds.S}(x, \mu)$  with  $\text{cds.R}^*$ .

Parties: The sender  $\text{cds.S}$  and the receiver  $\text{cds.R}$ . Inputs:  $\text{cds.S}$  has input  $(x, \mu)$  and  $\text{cds.R}$  has input  $w \in \{0, 1\}^m$ .

1. **Preamble:**  $\text{cds.R}$  sends a random string  $\rho$  as the first message of Naor's commitment scheme to  $\text{cds.S}$  and  $\text{cds.S}$  sends  $x$  to  $\text{cds.R}$ .
2. **Compute Garbled Circuits:**  $\text{cds.S}$  generates  $2\lambda$  garbled circuits, for the circuit computing  $G_{x,\mu}(w) = \mu' = \begin{cases} \mu & \text{if } \mathcal{R}_{\mathcal{L}}(x, w) = 1 \\ \perp & \text{otherwise} \end{cases}$ .

That is, for every  $i \in [2\lambda]$ ,  $(\hat{G}^i, \{\ell_b^{i,j}\}_{j \in [m], b \in \{0,1\}}) = \text{Garb}(G_{x,\mu}; \gamma_i)$ , where  $\hat{G}^i$  are the garbled circuits, and  $\ell$ 's are its associated labels.

3. **Cut-and-Choose:**  $\text{cds.R}$  samples a random subset  $\Lambda \subseteq [2\lambda]$ , by including each  $i \in [2\lambda]$  with probability  $1/2$ . For every  $i \in [2\lambda]$ , set

$$\sigma^i = \begin{cases} s^i \leftarrow \{0, 1\}^m & i \in \Lambda \\ w & i \notin \Lambda \end{cases}$$

4. **OT:** For every  $i \in [2\lambda], j \in [m], b \in \{0, 1\}$ ,  $\text{cds.S}$  samples  $r_b^{i,j}$ , the random coins for committing to the labels  $\ell_b^{i,j}$  via Naor's commitment scheme.  $\text{cds.S}$  and  $\text{cds.R}$  invokes  $\mathcal{F}_{\text{p-ot}}$  for  $2\lambda \times m$  parallel OT, where the  $(i, j)$ 'th OT for  $i \in [2\lambda], j \in [m]$  has sender's input strings  $(\ell_0^{i,j}, r_0^{i,j})$  and  $(\ell_1^{i,j}, r_1^{i,j})$ , and receiver's choice bit  $\sigma^{i,j}$  (which is the  $j$ -th bit of  $\sigma^i$ ) and  $\text{cds.R}$  receives  $(\tilde{\ell}^{i,j}, \tilde{r}^{i,j})$ .

We refer to the OTs with index  $(i, \star)$  as the  $i$ 'th batch. as they transfer labels of the  $i$ 'th garbled circuit  $\hat{G}^i$ .

5. **Send Garbled Circuits and Commitments to the Labels and  $\mu$ :**  $\text{cds.S}$  samples  $r^*$  and computes  $c^* = \text{com}_\rho(\mu; r^*)$  and  $c_b^{i,j} = \text{com}_\rho(\ell_b^{i,j}; r_b^{i,j})$ . Send  $\{\hat{G}^i\}_{i \in [2\lambda]}$  and  $(c^*, \{c_b^{i,j}\}_{i \in [2\lambda], j \in [m], b \in \{0,1\}})$  to the receiver  $\text{cds.R}$ .
6. **Proof of Consistency:**  $\text{cds.S}$  proves via ZK protocol that (a)  $c^*$  is a valid commitment to  $\mu$ , (b) every  $\hat{G}^i$  is a valid garbling of  $G_{x,\mu}$  with labels  $\{\ell_b^{i,j}\}_{j \in [m], b \in \{0,1\}}$ , and (c)  $c_b^{i,j}$  is a valid commitment to  $\ell_b^{i,j}$ .
7. **Checks:**  $\text{cds.R}$  performs the following checks:
  - If the ZK proof in the previous step is not accepting,  $\text{cds.R}$  aborts.
  - **$\Lambda$ -checks.** If there is  $i \in \Lambda$  and  $j \in [m]$ , such that,  $c_{\sigma^{i,j}}^{i,j} \neq \text{com}_\rho(\tilde{\ell}^{i,j}, \tilde{r}^{i,j})$ ,  $\text{cds.R}$  aborts and outputs  $\perp$ .
  - **$\bar{\Lambda}$ -check.** If for every  $i \notin \Lambda$ , there exists  $j \in [m]$ , such that,  $c_{\sigma^{i,j}}^{i,j} \neq \text{com}_\rho(\tilde{\ell}^{i,j}, \tilde{r}^{i,j})$ ,  $\text{cds.R}$  aborts and outputs  $\perp$ .
8. **Output:** If  $\text{cds.R}$  does not abort, there must exist  $i \notin \Lambda$  such that, for all  $j \in [m]$ ,  $c_{\sigma^{i,j}}^{i,j} = \text{com}_\rho(\tilde{\ell}^{i,j}, \tilde{r}^{i,j})$ . Evaluate the  $i$ 'th garbled circuit  $\hat{G}^i$  to get  $\mu' = \text{GEval}(\hat{G}^i, \{\tilde{\ell}^{i,j}\}_{j \in [m]})$ , and output  $x', \mu'$ .

**Fig. 3.** The verifiable CDS Scheme in  $\mathcal{F}_{\text{p-ot}}$ -hybrid model. The steps in color involve communication while the others only involve local computation.

*Hybrid 0.* This corresponds to the real execution of the protocol where the sender has input  $(x, m)$ . The view of  $\text{cds.R}^*$  consists of

$$\left[ \rho, \{ \widehat{G}^i, \widetilde{\ell}^{i,j}, \widetilde{r}^{i,j}, c_b^{i,j} \}_{i \in [2\lambda], j \in [m], b \in \{0,1\}}, c^*, \tau_{\text{ZK}} \right]$$

where  $\rho$  is the message sent by  $\text{cds.R}^*$  in Step 1, the strings  $\widetilde{\ell}^{i,j}$  and  $\widetilde{r}^{i,j}$  are received by  $\text{cds.R}^*$  from the OT functionality in Step 4, the garbled circuits  $\widehat{G}^i$  and the commitments  $c_b^{i,j}$  and  $c^*$  in Step 5, and  $\tau_{\text{ZK}}$  is the transcript of the ZK protocol between  $\text{cds.S}$  and  $\text{cds.R}^*$  in Step 6. (See the protocol in Fig. 3).

*Hybrid 1.* This is identical to hybrid 0 except that we run the simulator to intercept the OT queries  $(\sigma^1, \dots, \sigma^{2\lambda})$  of  $\text{cds.R}^*$ . The rest of the execution remains the same. Of course, the transcript produced is identical to that in hybrid 0.

*Hybrid 2.* In this hybrid, we replace the transcript  $\tau_{\text{ZK}}$  of the zero-knowledge protocol with a simulated transcript. This is indistinguishable from hybrid 1 by (post-quantum) computational zero-knowledge. Note that generating this hybrid does not require us to use the randomness underlying the commitments  $c_{1-\sigma^{i,j}}^{i,j}$  and  $c^*$ . (The randomness underlying  $c_{\sigma^{i,j}}^{i,j}$  are revealed as part of the OT responses to  $\text{cds.R}^*$ .)

*Hybrid 3.* In this hybrid, we replace half the commitments, namely  $c_{1-\sigma^{i,j}}^{i,j}$ , as well as  $c^*$  with commitments of 0. This is indistinguishable from hybrid 2 by (post-quantum) computational hiding of Naor commitments.

*Hybrid 4.* In this hybrid, we proceed as follows. If the simulator is in case 1, that is  $R_{\mathcal{L}}(x, \sigma^i) = 1$  for some  $i$ , proceed as in hybrid 3 with no change. On the other hand, if the simulator is in case 2, that is  $R_{\mathcal{L}}(x, \sigma^i) = 0$  for all  $i$ , replace the garbled circuits with simulated garbled circuits that always output  $\perp$  and let the commitments  $c_{\sigma^{i,j}}^{i,j}$  be commitments of the simulated labels. This is indistinguishable from hybrid 3 where the garbled circuits are an honest garbling of  $G_{x,\mu}$  because of the fact that all the garbled evaluations output  $\perp$  in hybrid 3, and because of the post-quantum security of the garbling scheme.

Hybrids 5–7 undo the effects of hybrids 2–4 in reverse.

*Hybrid 5.* In this hybrid, we replace the simulated garbled circuit with the real garbled circuit for the circuit  $G_{x,0}$ . This is indistinguishable from hybrid 4 because of the fact that all the garbled evaluations output  $\perp$  in this hybrid, and because of the post-quantum security of the garbling scheme.

*Hybrid 6.* In this hybrid, we let all commitments be to the correct labels and messages. This is indistinguishable from hybrid 5 by (post-quantum) computational hiding of Naor commitments.

*Hybrid 7.* In this hybrid, we replace the simulated ZK transcript with the real ZK protocol transcript. This is indistinguishable from hybrid 6 by (post-quantum) computational zero-knowledge.

This final hybrid matches exactly the simulator. This finishes the proof.

**Lemma 4.2.** *There is an inefficient statistical simulator against a malicious sender.*

*Proof.* The simulator  $\mathcal{S}$  interacts with  $\text{cds.S}^*$  as follows:

1. Send a string  $\rho$  to  $\text{cds.S}^*$  in Step 1, as in the protocol;
2. *Intercept* the OT messages  $(\ell_0^{i,j}, r_0^{i,j})$  and  $(\ell_1^{i,j}, r_1^{i,j})$  from  $\text{cds.S}^*$  in Step 4.
3. Run the rest of the protocol as an honest receiver  $\text{cds.R}$  would.
4. If the ZK proof rejects or if any  $\Lambda$ -check fails,  $\mathcal{S}$  aborts and outputs  $\perp$ . (Note the simulator does not perform the  $\bar{\Lambda}$ -check).
5. Otherwise, extract  $\mu$  from  $c^*$  using unbounded time, and send  $(x, \mu)$  to the ideal functionality and halt.

The transcript generated by  $\mathcal{S}$  is identical to the one generated in the real world where  $\text{cds.R}$  on input  $w$  interacts with  $\text{cds.S}^*$ . It remains to analyze the output distribution of  $\text{cds.R}$  in the simulation vis-a-vis the real world.

1. Since the  $\Lambda$ -checks performed on the commitments of garbled instances in  $\Lambda$  by the simulator and the ones performed by the honest receiver in the real protocol are exactly the same, we have that the probability that the probability of abort is the same (for this step) in both scenarios.
2. The probability that the honest receiver in the real protocol aborts on the  $\bar{\Lambda}$ -check, conditioned on the fact that the  $\Lambda$ -checks passed, is negligible.

Thus, we have that the output distributions of the receiver are negligibly close between the simulation and the real world, finishing up the proof.

**Lemma 4.3.** *The protocol is verifiable.*

*Proof.* We first construct a verification algorithm  $\text{Ver}$ .

- The classical transcript  $\tau$  consists of  $\rho, x, \{\hat{G}^i\}_{i \in [2\lambda]}, c^*, \{c_b^{i,j}\}_{i \in [2\lambda], j \in [m], b \in \{0,1\}}$ .
- At the end of the protocol,  $\text{cds.S}$  outputs  $(x, \mu, r^*)$  on its special output tape.
- The verification algorithm  $\text{Ver}(\tau, x, \mu', r') = 1$  iff  $c^* = \text{com}_\rho(\mu'; r')$ .

We first claim that for honest  $\text{cds.S}$  and  $\text{cds.R}$  with  $(x, w) \in \mathcal{R}_\mathcal{L}$ , we have that  $\text{Ver}(\tau, x, \mu, r) = 1$ . Since all parties in the protocol are honest the input  $x$  in  $\tau$  is the same as the one output by  $\text{cds.S}$  and we have that  $c^*$  is the commitment to the honest message using the correct randomness, so  $\text{Ver}$  outputs 1.

To show binding, assume that the verification passes and the receiver does not abort. Then, we know that there is at least one  $i \notin \Lambda$  such that the  $i$ -th garbled circuit+input pair is correct and the circuit is the garbling of  $G_{x,\mu}$ . The verifier will evaluate the circuit on input  $w$  and obtain either  $\perp$  when  $R_\mathcal{L}(x, w) = 0$  or  $\mu$  when  $R_\mathcal{L}(x, w) = 1$ , exactly as required.

### 4.3 Extractable Commitment from CDS

**Theorem 4.2.** *Assume the existence of pqOWF. There is a commitment protocol  $\langle C, R \rangle$  that C-QSA-emulates  $\mathcal{F}_{\text{so-com}}$  with efficient simulators.*

**Parties:** The committer  $C$  and the receiver  $R$ .

**Inputs:**  $C$  gets a message vector  $\vec{\mu} = (\mu_1, \dots, \mu_{\ell(n)})$  and  $R$  gets  $1^n$ .

### Commitment Phase

1. **Preamble.**  $C$  sends a random string  $\rho$  to  $R$ , and  $R$  sends a random string  $\rho^*$  to  $C$ , as the first message of the Naor commitment scheme.
2. **Set up a Trapdoor Statement.**
  - $R$  sends a Naor commitment  $c = \text{com}_\rho(0; r)$ .
  - $R$  proves to  $C$  using a ZK protocol that  $c$  is a commitment to 0, that is,  $((c, \rho, 0), r) \in \mathcal{R}_{\mathcal{L}_{\text{com}}}$ . If the ZK verifier rejects,  $C$  aborts.
3. **CDS.**  $C$  and  $R$  run the CDS protocol  $\langle \text{cds.S}, \text{cds.R} \rangle$  for the language  $\mathcal{L}_{\text{com}}$  where  $C$  acts as  $\text{cds.S}$  with input  $x = (c, \rho, 1)$  and message  $\vec{\mu}$ , and  $R$  acts as  $\text{cds.R}$  with input 0.  
 $C$  aborts if  $\text{cds.S}$  aborts, else  $C$  obtains the protocol transcript  $\tau$  and  $\text{cds.S}$ 's proof  $\pi$ .  $R$  aborts if  $\text{cds.R}$  aborts, or if  $\text{cds.R}$  outputs  $(x', \vec{\mu}')$  but  $x' \neq (\rho, c, 1)$ .
4. **Commit and Prove Consistency.**
  - $C$  sends a Naor commitment  $c^* = \text{com}_{\rho^*}(\vec{\mu}; r^*)$ .
  - $C$  proves to  $R$  using a ZK protocol there exists a  $\vec{\mu}$  such that  $(x = (\rho, c, 1), \vec{\mu})$  is the input that  $C$  used in the CDS protocol and  $\vec{\mu}$  is committed in  $c^*$ , that is:

$$\text{Ver}(\tau, x, \vec{\mu}, \pi) = 1 \text{ and } c^* = \text{com}_{\rho^*}(\vec{\mu}, r^*)$$

5.  $R$  accepts this commitment if the ZK proof is accepting.

### Decommitment Phase

1.  $R$  sends  $I \subseteq [\ell]$ .
2.  $C$  sends  $\vec{\mu}|_I$  and proves via a ZK protocol that  $c^*|_I$  commits to  $\vec{\mu}|_I$ .
3.  $R$  accepts this decommitment if the ZK proof is accepting.

**Fig. 4.** Extractable Selective-Opening-Secure commitment scheme

*Proof.* The construction of our extractable commitment scheme is given in Fig. 4. The protocol uses Naor's classical statistically binding commitment protocol and a verifiable CDS protocol  $\Pi = \langle \text{cds.S}, \text{cds.R} \rangle$  that  $\mathbf{C}$ -QSA-emulates  $\mathcal{F}_{\text{cds}}$  (with unbounded simulation) for  $\mathcal{L}_{\text{com}}$ , the language consisting of all Naor's commitments  $(\rho, c)$  to a bit  $b$ :  $\mathcal{R}_{\mathcal{L}_{\text{com}}}((\rho, c, b), r) = 1$  iff  $c = \text{com}_\rho(b; r)$ .

We defer a detailed description of these tools to the full version of our paper.

In Lemma 4.4 (resp. Lemma 4.5), we show that the protocol has an efficient simulator for a corrupted sender (resp. receiver).

**Lemma 4.4.** *There is an efficient simulator against a malicious sender.*

*Proof.* The simulator  $\mathcal{S}$  against a malicious committer  $C^*$  works as follows.

1. In step 1, proceed as an honest receiver would.

2. In step 2, send a Naor commitment  $c = \text{com}_\rho(1; r)$  (instead of 0) and simulate the ZK proof.
3. In step 3, run the honest CDS protocol with  $r$  as witness, gets  $\mu$  and sends it to the ideal functionality  $\mathcal{F}_{\text{so-com}}$ .
4. Run the rest of the protocol as an honest receiver would.

We now show, through a sequence of hybrids, that this simulator produces a joint distribution of a view of  $C^*$  together with an output of  $R$  that is computationally indistinguishable from that in the real execution of  $C^*$  with  $R$ . In order to show this we consider the following sequence of hybrids.

*Hybrid 0.* This corresponds to the protocol  $\Pi_{H_0}^{\text{ECom}}$ , where  $\mathcal{S}_0$  sits between  $C^*$  and the honest receiver in the real protocol and just forwards their messages. It follows trivially that  $M_{\Pi_{\text{ECom}}, C^*} \approx_{qc} M_{\Pi_{H_0}^{\text{ECom}}, \mathcal{S}_0}$ .

*Hybrid 1.*  $\mathcal{S}_1$  interacts with  $C^*$  following the protocol  $\Pi_{H_1}^{\text{ECom}}$ , which is the same as  $\Pi_{H_0}^{\text{ECom}}$  except that  $\mathcal{S}_1$  uses the ZK simulator instead of the proof that  $((c, \rho, 0), r) \in \mathcal{R}_{\mathcal{L}_{\text{com}}}$ . From the computational zero-knowledge property of the protocol, we have that  $M_{\Pi_{H_0}^{\text{ECom}}, \mathcal{S}_0} \approx_{qc} M_{\Pi_{H_1}^{\text{ECom}}, \mathcal{S}_1}$ .

*Hybrid 2.*  $\mathcal{S}_2$  interacts with  $C^*$  following the protocol  $\Pi_{H_2}^{\text{ECom}}$ , which is the same as  $\Pi_{H_1}^{\text{ECom}}$  except that  $\mathcal{S}_2$  sends  $c' = \text{com}_\rho(1; r)$  instead of the (honest) commitment of 0. When  $\mathcal{S}_2$  simulates  $\mathcal{F}_{\text{zk}}$ , she still sends a message that  $c'$  is a valid input. It follows from computationally hiding property of Naor's commitment scheme that  $M_{\Pi_{H_1}^{\text{ECom}}, \mathcal{S}_1} \approx_{qc} M_{\Pi_{H_2}^{\text{ECom}}, \mathcal{S}_2}$ .

*Hybrid 3.*  $\mathcal{S}_3$  interacts with  $C^*$  following the protocol  $\Pi_{H_3}^{\text{ECom}}$ , which is the same as  $\Pi_{H_2}^{\text{ECom}}$  except that  $\mathcal{S}_3$  now uses the private randomness  $r$  as a witness that  $c'$  is a commitment of 1.

Since our protocol realizes  $\mathcal{F}_{\text{CDS}}$ ,  $\text{cds.S}^*$  (controlled by  $C^*$ ) does not behave differently depending on the input of  $\text{cds.R}$ , so the probability of abort in step 3 does not change. Notice also that  $\text{Ver}(\tau, x, \mu, \pi)$  is independent of  $\text{cds.R}$ 's message, so the acceptance probability of the ZK proof does not change either.

Then, if the ZK proof leads to acceptance, by the soundness of the protocol, we know that  $\text{Ver}(\tau, x, \mu, \pi) = 1$  and by the binding of the commitment  $c^*$ , such a  $\mu$  is uniquely determined.

Finally, by the verifiability of the CDS protocol, we know that the receiver either aborts or outputs the specified  $\mu$ . Thus, the outputs of the receiver  $R$  in the simulated execution and the real execution must be the same in this case.

**Lemma 4.5.** *There is an efficient simulator against a malicious receiver.*

*Proof.* The simulator  $\mathcal{S}$  against a malicious receiver  $R^*$  proceeds as follows.

- In steps 1 and 2, proceed as an honest sender would.
- In step 3, run the CDS protocol using a message vector  $\mu = \mathbf{0}$  of all zeroes.
- In step 4, commit to the all-0 vector and produce a simulated ZK proof.
- During decommitment, send  $I \subseteq [\ell]$  to the ideal functionality and receive  $\mu|_I$ . Send  $\mu|_I$  to  $R^*$ , and simulate the ZK proof.



We now show, through a sequence of hybrids, that this simulator is computationally indistinguishable from the real execution of  $C(\mu)$  with  $R^*$ .

*Hybrid 0.* This corresponds to the protocol  $\Pi_{H_0}^{\text{ECom}}$ , where  $\mathcal{S}_0$  sits between the honest committer  $C$  and  $R^*$ , and it just forwards their messages. It follows trivially that  $M_{\Pi_{\text{ECom}}, C^*} \approx_{qc} M_{\Pi_{H_0}^{\text{ECom}}, \mathcal{S}_0}$ .

*Hybrid 1.*  $\mathcal{S}_1$  interacts with  $R^*$  following the protocol  $\Pi_{H_1}^{\text{ECom}}$ , which is the same as  $\Pi_{H_0}^{\text{ECom}}$  except that  $\mathcal{S}_1$  uses the ZK simulator in Step 4 and the decommitment phase. From the computational zero-knowledge property, we have that  $M_{\Pi_{H_0}^{\text{ECom}}, \mathcal{S}_0} \approx_{qc} M_{\Pi_{H_1}^{\text{ECom}}, \mathcal{S}_1}$ .

*Hybrid 2.*  $\mathcal{S}_2$  interacts with  $R^*$  following the protocol  $\Pi_{H_2}^{\text{ECom}}$ , which is the same as  $\Pi_{H_1}^{\text{ECom}}$  except that  $\mathcal{S}_2$  sets  $c^*$  to be a commitment to 0. It follows from the computationally-hiding property of the commitment scheme that  $M_{\Pi_{H_1}^{\text{ECom}}, \mathcal{S}_1} \approx_{qc} M_{\Pi_{H_2}^{\text{ECom}}, \mathcal{S}_2}$ .

*Hybrid 3.*  $\mathcal{S}_3$  interacts with  $R^*$  following the protocol  $\Pi_{H_3}^{\text{ECom}}$ , which is the same as  $\Pi_{H_2}^{\text{ECom}}$  except that  $\mathcal{S}_3$  uses  $\mu = 0^\ell$  as the  $\text{cds.S}$  message.

From the soundness of the ZK proof in Step 2, we have that  $c$  is not a commitment of 1. In this case, by the security of CDS,  $R^*$  does not receive  $\mu$ , so the change of the message cannot be distinguished.

Notice that Hybrid 3 matches the description of the simulator  $\mathcal{S}$ , and therefore  $M_{\Pi_{H_2}^{\text{ECom}}, \mathcal{S}_2} \approx_{qc} M_{\mathcal{F}_{\text{so-com}}, \mathcal{S}}$ .

## 5 Multiparty (Quantum) Computation in MiniQCrypt

Our quantum protocol realizing  $\mathcal{F}_{\text{so-com}}$  from quantum-secure OWF allows us to combine existing results and realize secure computation of any two-party or multi-party classical functionality as well as quantum circuit in MiniQCrypt.

**Theorem 5.1.** *Assuming that post-quantum secure one-way functions exist, for every classical two-party and multi-party functionality  $\mathcal{F}$ , there is a quantum protocol C-QSA-emulates  $\mathcal{F}$ .*

*Proof.* By Theorem 3.2, we readily realize  $\mathcal{F}_{\text{ot}}$  in MiniQCrypt. In the  $\mathcal{F}_{\text{ot}}$ -hybrid model, any classical functionality  $\mathcal{F}$  can be realized statistically by a classical protocol in the universal-composable model [42]. The security can be lifted to the quantum universal-composable model as shown by Unruh [61]. As a result, we also get a classical protocol in the  $\mathcal{F}_{\text{ot}}$ -hybrid model that S-QSA emulates  $\mathcal{F}$ . Plugging in the quantum protocol for  $\mathcal{F}_{\text{ot}}$ , we obtain a quantum protocol that C-QSA-emulates  $\mathcal{F}$  assuming existence of quantum-secure one-way functions.

Now that we have a protocol that realizes any classical functionality in MiniQCrypt, we can instantiate  $\mathcal{F}_{\text{mpc}}$  used in the work of [24] to achieve a protocol for secure multi-party quantum computation where parties can jointly evaluate an arbitrary quantum circuit on their private quantum input states. Specifically

consider a quantum circuit  $Q$  with  $k$  input registers. Let  $\mathcal{F}_Q$  be the ideal protocol where a trusted party receives private inputs from  $k$  parties, evaluate  $Q$ , and then send the outputs to respective parties. We obtain the following.

**Theorem 5.2.** *Assuming that post-quantum secure one-way functions exist, for any quantum circuit  $Q$ , there is a quantum protocol that C-QSA-emulates the  $\mathcal{F}_Q$ .*

**Acknowledgements.** We thank the Simons Institute for the Theory of Computing for providing a meeting place where the seeds of this work were planted. VV thanks Ran Canetti for patiently answering his questions regarding universally composable commitments.

Most of this work was done when AG was affiliated to CWI and QuSoft. HL was supported by NSF grants CNS-1528178, CNS-1929901, CNS-1936825 (CAREER), CNS-2026774, a Hellman Fellowship, a JP Morgan AI Research Award, the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236, and a subcontract No. 2017-002 through Galois. FS was supported by NSF grants CCF-2041841, CCF-2042414, and CCF-2054758 (CAREER). VV was supported by DARPA under Agreement No. HR00112020023, a grant from the MIT-IBM Watson AI, a grant from Analog Devices, a Microsoft Trustworthy AI grant, and a DARPA Young Faculty Award. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, DARPA, the National Science Foundation, or the U.S. Government.

## References

1. Agarwal, A., Bartusek, J., Goyal, V., Khurana, D., Malavolta, G.: Post-quantum multi-party computation in constant rounds (2020). [arXiv:2005.12904](https://arxiv.org/abs/2005.12904). <https://arxiv.org/abs/2005.12904>
2. Ananth, P., La Placa, R.L.: Secure quantum extraction protocols. CoRR, abs/1911.07672 (2019)
3. Barak, B.: The complexity of public-key cryptography. Cryptology ePrint Archive, Report 2017/365, 2017. <https://eprint.iacr.org/2017/365>
4. Bartusek, J., Coladangelo, A., Khurana, D., Ma, F.: One-way functions imply secure computation in a quantum world (2020)
5. Beaver, D.: Correlated pseudorandomness and the complexity of private computations. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, pp. 479–488. ACM (1996)
6. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-01001-9\\_1](https://doi.org/10.1007/978-3-642-01001-9_1)
7. Ben-Or, M., Crépeau, C., Gottesman, D., Hassidim, A., Smith, A.: Secure multi-party quantum computation with (only) a strict honest majority. In: 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 249–260. IEEE (2006)
8. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: IEEE International Conference on Computers, Systems and Signal Processing, vol. 175, p. 8 (1984)

9. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, M.-H.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 351–366. Springer, Heidelberg (1992). [https://doi.org/10.1007/3-540-46766-1\\_29](https://doi.org/10.1007/3-540-46766-1_29)  
As references [10, 11] and [51, 52] are same, we have deleted the duplicate reference and renumbered accordingly. Please check and confirm
10. Bitansky, N., Shmueli, O.: Post-quantum zero knowledge in constant rounds. In: Makarychev, K., Makarychev, Y., Tulsiani, M., Kamath, G., Chuzhoy, J. (eds.) STOC 2020, pp. 269–279. ACM (2020)
11. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians (1986)
12. Bouman, N.J., Fehr, S.: Sampling in a quantum population, and applications. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 724–741. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14623-7\\_39](https://doi.org/10.1007/978-3-642-14623-7_39)
13. Brakerski, Z., Christiano, P., Mahadev, U., Vazirani, U.V., Vidick, T.: A cryptographic test of quantumness and certifiable randomness from a single quantum device. In: FOCS 2018, pp. 320–331 (2018)
14. Buhrman, H., Christandl, M., Schaffner, C.: Complete insecurity of quantum protocols for classical two-party computation. *Phys. Rev. Lett.* **109**(16), 160501 (2012)
15. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: FOCS, pp. 136–145. IEEE (2001)
16. Chailloux, A., Gutoski, G., Sikora, J.: Optimal bounds for semi-honest quantum oblivious transfer. *Chic. J. Theor. Comput. Sci.* **2016**, 1–17 (2016)
17. Colbeck, R.: Quantum and relativistic protocols for secure multi-party computation. Ph.D. Thesis, Trinity College, University of Cambridge (2009)
18. Crépeau, C., Gottesman, D., Smith, A.: Secure multi-party quantum computation. In: Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, pp. 643–652 (2002)
19. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions. In: 29th Annual Symposium on Foundations of Computer Science, pp. 42–52 (1988)
20. Damgård, I., Fehr, S., Lunemann, C., Salvail, L., Schaffner, C.: Improving the security of quantum protocols via commit-and-open. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 408–427. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-03356-8\\_24](https://doi.org/10.1007/978-3-642-03356-8_24)
21. Damgård, I.B., Fehr, S., Renner, R., Salvail, L., Schaffner, C.: A tight high-order entropic quantum uncertainty relation with applications. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 360–378. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74143-5\\_20](https://doi.org/10.1007/978-3-540-74143-5_20)
22. Damgård, I.B., Fehr, S., Salvail, L., Schaffner, C.: Cryptography in the bounded-quantum-storage model. *SIAM J. Comput.* **37**(6), 1865–1890 (2008)
23. Dixon, A.R., Yuan, Z.L., Dynes, J.F., Sharpe, A.W., Shields, A.J.: Gigahertz decoy quantum key distribution with 1 mbit/s secure key rate. *Opt. Express* **16**(23), 18790 (2008)
24. Dulek, Y., Grilo, A.B., Jeffery, S., Majenz, C., Schaffner, C.: Secure multi-party quantum computation with a dishonest majority. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 729–758. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-45727-3\\_25](https://doi.org/10.1007/978-3-030-45727-3_25)
25. Dupuis, F., Fehr, S., Lamontagne, P., Salvail, L.: Adaptive versus non-adaptive strategies in the quantum setting with applications. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 33–59. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-53015-3\\_2](https://doi.org/10.1007/978-3-662-53015-3_2)

26. Dupuis, F., Nielsen, J.B., Salvail, L.: Secure two-party quantum evaluation of unitaries against specious adversaries. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 685–706. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14623-7\\_37](https://doi.org/10.1007/978-3-642-14623-7_37)
27. Dupuis, F., Nielsen, J.B., Salvail, L.: Actively secure two-party evaluation of any quantum operation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 794–811. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-32009-5\\_46](https://doi.org/10.1007/978-3-642-32009-5_46)
28. Fang, J., Unruh, D., Weng, J., Yan, J., Zhou, D.: How to base security on the perfect/statistical binding property of quantum bit commitment? IACR Cryptol. ePrint Arch. **2020**, 621 (2020)
29. Fehr, S., Katz, J., Song, F., Zhou, H.-S., Zikas, V.: Feasibility and completeness of cryptographic tasks in the quantum world. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 281–296. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36594-2\\_16](https://doi.org/10.1007/978-3-642-36594-2_16)
30. Fehr, S., Schaffner, C.: Composing quantum protocols in a classical environment. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 350–367. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-00457-5\\_21](https://doi.org/10.1007/978-3-642-00457-5_21)
31. Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. In: Vitter, J.S. (ed.) STOC 1998, pp. 151–160. ACM (1998)
32. Goldreich, O.: Foundations of Cryptography: Volume 2 Basic Applications, 1st edn. Cambridge University Press, Cambridge (2009)
33. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC, pp. 218–229. ACM Press (May 1987)
34. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design (extended abstract). In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 171–185. Springer, Heidelberg (1987). [https://doi.org/10.1007/3-540-47721-7\\_11](https://doi.org/10.1007/3-540-47721-7_11)
35. Grilo, A.B., Lin, H., Song, F., Vaikuntanathan, V.: Oblivious transfer is in miniqcrypt. Cryptology ePrint Archive, Report 2020/1500 (2020). <https://eprint.iacr.org/2020/1500>
36. Hallgren, S., Smith, A., Song, F.: Classical cryptographic protocols in a quantum world. Int. J. Quant. Inf. **13**(04), 1550028 (2015). Preliminary version in Crypto 2011
37. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
38. Hiskett, P.A., et al.: Long-distance quantum key distribution in optical fibre. New J. Phys. **8**(9), 193 (2006)
39. Impagliazzo, R.: A personal view of average-case complexity. In: Structure in Complexity Theory Conference, Annual, p. 134, Los Alamitos, CA, USA. IEEE Computer Society (Jun 1995)
40. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: Johnson, D.S. (ed.) STOC 1989, pp. 44–61. ACM (1989)
41. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). [https://doi.org/10.1007/978-3-540-45146-4\\_9](https://doi.org/10.1007/978-3-540-45146-4_9)
42. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-85174-5\\_32](https://doi.org/10.1007/978-3-540-85174-5_32)

43. Kilian, J.: Founding cryptography on oblivious transfer. In: 20th ACM STOC, pp. 20–31. ACM Press (May 1988)
44. König, R., Wehner, S., Wullschlegel, J.: Unconditional security from noisy quantum storage. *IEEE Trans. Inf. Theor.* **58**(3), 1962–1984 (2012)
45. Liao, S.-K., et al.: Satellite-relayed intercontinental quantum network. *Phys. Rev. Lett.* **120**(3), 030501 (2018)
46. Liu, Y.K.: Building one-time memories from isolated qubits. In: 5th Conference on Innovations in Theoretical Computer Science, pp. 269–286 (2014)
47. Liu, Y.-K.: Single-shot security for one-time memories in the isolated qubits model. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 19–36. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44381-1\\_2](https://doi.org/10.1007/978-3-662-44381-1_2)
48. Lo, H.-K.: Insecurity of quantum secure computations. *Phys. Rev. A* **56**(2), 1154–1162 (1997)
49. Lo, H.K., Chau, H.F.: Is quantum bit commitment really possible? *Phys. Rev. Lett.* **78**(17), 3410–3413 (1997)
50. Lunemann, C., Nielsen, J.B.: Fully simulatable quantum-secure coin-flipping and applications. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 21–40. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21969-6\\_2](https://doi.org/10.1007/978-3-642-21969-6_2)
51. Maji, H.K., Prabhakaran, M., Rosulek, M.: A zero-one law for cryptographic complexity with respect to computational UC security. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 595–612. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14623-7\\_32](https://doi.org/10.1007/978-3-642-14623-7_32)
52. Mayers, D., Salvail L.: Quantum oblivious transfer is secure against all individual measurements. In: Proceedings Workshop on Physics and Computation. PhysComp 1994, pp. 69–77 (1994)
53. Mayers, D.: Unconditionally secure quantum bit commitment is impossible. *Phys. Rev. Lett.* **78**(17), 3414 (1997)
54. Naor, M.: Bit commitment using pseudo-randomness. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 128–136. Springer, New York (1990). [https://doi.org/10.1007/0-387-34805-0\\_13](https://doi.org/10.1007/0-387-34805-0_13)
55. Pugh, C.J., et al.: Airborne demonstration of a quantum key distribution receiver payload. *Quant. Sci. Technol.* **2**(2), 024009 (2017)
56. Rabin, M.: How to exchange secrets by oblivious transfer. Technical Memo TR-81, Aiken Computation Laboratory, Harvard University (1981)
57. Rudich, S.: The use of interaction in public cryptosystems. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 242–251. Springer, Heidelberg (1992). [https://doi.org/10.1007/3-540-46766-1\\_19](https://doi.org/10.1007/3-540-46766-1_19)
58. Salvail, L.: Quantum bit commitment from a physical assumption. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 338–353. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0055740>
59. Salvail, L., Schaffner, C., Sotáková, M.: Quantifying the leakage of quantum protocols for classical two-party cryptography. *Int. J. Quant. Inf.* **13**(04), 1450041 (2015)
60. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: FOCS 1994, pp. 124–134. IEEE Computer Society (1994)
61. Unruh, D.: Universally composable quantum multi-party computation. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 486–505. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_25](https://doi.org/10.1007/978-3-642-13190-5_25)

- 62. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-29011-4\\_10](https://doi.org/10.1007/978-3-642-29011-4_10)
- 63. Vazirani, U., Vidick, T.: Certifiable quantum dice: or, true random number generation secure against quantum adversaries. In: STOC 2012, pp. 61–76. Association for Computing Machinery (2012)
- 64. Watrous, J.: Zero-knowledge against quantum attacks. SIAM J. Comput. **39**(1), 25–58 (2009). Preliminary version in STOC 2006
- 65. Wiesner, S.: Conjugate coding. SIGACT News **15**(1), 78–88 (1983)
- 66. Yao, A.C.C.: Security of quantum protocols against coherent measurements. In: 27th ACM STOC, pp. 67–75. ACM Press (May/June 1995)



## ZERO-KNOWLEDGE PROOF SYSTEMS FOR QMA\*

ANNE BROADBENT<sup>†</sup>, ZHENGFENG JI<sup>‡</sup>, FANG SONG<sup>§</sup>, AND JOHN WATROUS<sup>¶</sup>

**Abstract.** Prior work has established that all problems in NP admit classical zero-knowledge proof systems, and under reasonable hardness assumptions for quantum computations, these proof systems can be made secure against quantum attacks. We prove a result representing a further quantum generalization of this fact, which is that every problem in the complexity class QMA has a quantum zero-knowledge proof system. More specifically, assuming the existence of an unconditionally binding and quantum computationally concealing commitment scheme, we prove that every problem in the complexity class QMA has a quantum interactive proof system that is zero-knowledge with respect to efficient quantum computations. Our QMA proof system is sound against arbitrary quantum provers, but only requires an honest prover to perform polynomial-time quantum computations, provided that it holds a quantum witness for a given instance of the QMA problem under consideration. The proof system relies on a new variant of the QMA-complete local Hamiltonian problem in which the local terms are described by Clifford operations and standard basis measurements. We believe that the QMA-completeness of this problem may have other uses in quantum complexity.

**Key words.** QMA, local-Hamiltonian problem, zero-knowledge, quantum computation

**AMS subject classifications.** 81P45, 81P68, 81P94

**DOI.** 10.1137/18M1193530

**1. Introduction.** Zero-knowledge proof systems, which were first introduced by Goldwasser, Micali, and Rackoff [28], are interactive protocols that allow a prover to convince a verifier of the validity of a statement while revealing no additional information beyond the statement's validity. As paradoxical as it may seem upon a first consideration, several problems that are not known to be efficiently computable, such as the quadratic non-residuosity, graph isomorphism, and graph non-isomorphism problems, admit zero-knowledge proof systems [26, 28]. Under reasonable intractability assumptions, Goldreich, Micali, and Wigderson [26] gave a zero-knowledge protocol for the graph 3-coloring problem and, because of its NP-completeness, for all NP problems. This line of work was further extended in [7], which showed that all problems in IP have zero-knowledge proof systems.

Since the invention of this concept, zero-knowledge proof systems have become a cornerstone of modern theoretical cryptography. In addition to the conceptual inno-

---

\*Received by the editors June 15, 2018; accepted for publication (in revised form) November 5, 2019; published electronically March 10, 2020. A preliminary version of this paper appeared in Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS 2016), IEEE, Piscataway, NJ, 2016, pp. 31–40.

<https://doi.org/10.1137/18M1193530>

**Funding:** The first, third, and fourth authors were supported in part by Canada's NSERC. The third author was supported in part by Cryptoworks21, CIFAR, and the U.S. National Science Foundation.

<sup>†</sup>Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada (abroadbe@uottawa.ca).

<sup>‡</sup>Centre for Quantum Software and Information, School of Computer Science, University of Technology Sydney, Sydney, Australia, and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China (jizhengfeng@gmail.com).

<sup>§</sup>Computer Science and Engineering Department, Texas A&M University, College Station, TX 77843 (fang.song@tamu.edu).

<sup>¶</sup>Institute for Quantum Computing and School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and Canadian Institute for Advanced Research, Toronto, ON MSG 1M1, Canada (watrous@uwaterloo.ca).

vation of a complexity-theoretic notion of knowledge, zero-knowledge proof systems are essential building blocks in a host of cryptographic constructions. One notable example is the design of secure two-party and multiparty computation protocols [25].

The extensive works on zero-knowledge largely reside in a classical world. The development of quantum information science and technology has urged another look at the landscape of zero-knowledge proof systems in a *quantum* world. Namely, both honest users and adversaries may potentially possess the capability to exchange and process quantum information. There are, of course, zero-knowledge protocols that immediately become insecure in the presence of quantum attacks due to efficient quantum algorithms that break the intractability assumptions upon which these protocols rely. For instance, Shor's quantum algorithms for factoring and computing discrete logarithms [49] invalidate the use of these problems, generally conjectured to be classically hard, as a basis for the security of zero-knowledge protocols against quantum attacks. Even with computational assumptions against quantum adversaries, however, it is still highly nontrivial to establish the security of classical zero-knowledge proof systems in the presence of malicious quantum verifiers because of a technical reason that we now briefly explain.

The zero-knowledge property of a proof system for a fixed input string is concerned with the computations that may be realized through an interaction between a (possibly malicious) verifier and the prover. That is, the malicious verifier may take an arbitrary additional input (usually called the *auxiliary input* to distinguish it from the input string to the proof system under consideration), interact with the prover in any way it sees fit, and produce an output that is representative of what it has learned through the interaction. Roughly speaking, the prover is said to be *zero-knowledge* on the fixed input string if any computation of the sort just described can be efficiently approximated<sup>1</sup> by a *simulator* operating entirely on its own—meaning that it does not interact with the prover, and in the case of an NP problem it does not possess a witness for the fixed problem instance being considered. The proof system is then said to be zero-knowledge when this zero-knowledge property holds for all yes-instances of the problem under consideration.

Classically, the zero-knowledge property is typically established through a technique known as *rewinding*. In essence, the simulator can store a copy of the auxiliary input, and it can make guesses and store intermediate states representing a hypothetical prover/verifier interaction—and if it makes a bad guess or otherwise experiences bad luck when simulating this hypothetical interaction, it simply reverts to an earlier stage (or back to the beginning) of the simulation and tries again. Indeed, it is generally the simulator's freedom to disregard the temporal restrictions of the actual prover/verifier interaction in a way such as this that makes it possible to succeed.

However, rewinding a quantum simulation is more problematic; the *no-cloning theorem* [60] forbids one from copying quantum information, making it impossible to store a copy of the input or of an intermediate state, and measurements generally have an irreversible effect [21] that may partially destroy quantum information. Such difficulties were first observed by van de Graaf [54] and further studied in [14, 55]. Later, a *quantum rewinding* technique was found [58] to establish that several interactive proof systems, including the Goldreich–Micali–Wigderson graph 3-coloring proof system [26], remain zero-knowledge against malicious quantum verifiers (un-

<sup>1</sup>Different notions of approximations are considered, including *statistical* approximations and *computational* approximations, which require that the simulator's computation is either statistically (or information-theoretically) indistinguishable or computationally indistinguishable from the malicious verifier's computation. This paper is primarily concerned with the computational variant.

der appropriate quantum intractability assumptions in some cases). It follows that all NP problems have zero-knowledge proof systems even against quantum malicious verifiers, provided that a quantum analogue of the intractability assumption required by the Goldreich–Micali–Wigderson graph 3-coloring proof system are in place.

This work studies the quantum analogue of NP, known as QMA, in the context of zero-knowledge. This is the class of problems having succinct quantum witnesses satisfying similar completeness and soundness conditions to NP (or its randomized variant MA). Quantum witnesses and verification are conjectured to be more powerful than their classical counterparts: there are problems that admit short quantum witnesses, whereas there is no known method for verification using a polynomial-sized classical witness. In other words,  $\text{NP} \subseteq \text{QMA}$  holds trivially, and the containment is typically conjectured to be proper. The question we address in this paper is, *does every problem in QMA have a zero-knowledge quantum interactive proof system?* In more philosophical terms, viewing quantum witnesses as precious sources of knowledge, *can one always devise a proof system that reveals nothing about a quantum witness beyond its validity?*

**1.1. Our contributions.** We answer the above question positively by constructing a quantum interactive proof system for any problem in QMA. It is zero-knowledge against any polynomial-time quantum adversary, under a reasonable quantum intractability assumption.

**THEOREM 1.1.** *Assuming the existence of an unconditionally binding and quantum computationally concealing bit commitment scheme, every problem in QMA has a quantum computational zero-knowledge proof system.*

A few of the desirable features of our proof system are as follows:

1. Our proof system has a simple structure, similar to the graph 3-coloring proof system of Goldreich–Micali–Wigderson (and to so-called  $\Sigma$ -protocols more generally). It can be viewed as a three-phase process: the prover commits to a quantum witness, the verifier makes a random challenge, and finally the prover responds to the challenge by partial opening of the committed information that suffices to certify the validity.
2. All communications in our proof system are classical except for the first commitment message, and the verifier can measure the quantum message immediately upon its arrival (which has a strong technological appeal).
3. Our protocol is based on plausible computational assumptions. The sort of bit commitment scheme it requires can be implemented, for instance, under the existence of injective one-way functions that are hard to invert in quantum polynomial time.
4. Our protocol is prover-efficient: given a valid quantum witness, an honest prover only needs to perform efficient quantum computations in order to convince the verifier to accept with high probability. Moreover, as has already been suggested, aside from the preparation of the first quantum message, all of the remaining computations performed by the honest prover are classical polynomial-time computations. No computational assumptions on the prover are required in the soundness case; the protocol is sound against arbitrary quantum provers.

As a key ingredient of our zero-knowledge proof system, we introduce a new variant of the  $k$ -local Hamiltonian problem and prove that it remains QMA-complete (with respect to Karp reductions). The  $k$ -local Hamiltonian problem asks if the minimum

eigenvalue (or ground state energy in physics parlance) of an  $n$ -qubit Hamiltonian  $H = \sum_j H_j$ , where each  $H_j$  is  $k$ -local (i.e., acts trivially on all but  $k$  of the  $n$  qubits), is below a particular threshold value. This problem was introduced and proved to be QMA-complete by Kitaev, Shen, and Vazirani [40] for the case  $k = 5$ , and subsequently improved to  $k = 2$  [37]. We show (for the case  $k = 5$ ) that each  $H_j$  can be restricted to be realized by a Clifford operation, followed by a standard basis measurement, and the QMA-completeness is preserved. Beyond its use in this paper, this fact has the potential to provide other insights into the study of quantum Hamiltonian complexity. For an arbitrary problem  $A \in \text{QMA}$ , we can reduce an instance of  $A$  efficiently to an instance of the  $k$ -local Clifford Hamiltonian problem, and a valid witness for  $A$  can also be transformed into a witness for the corresponding  $k$ -local Clifford Hamiltonian problem instance by an efficient quantum procedure. As a result,  $A$  has a zero-knowledge proof system by composing this reduction with our zero-knowledge proof system for the  $k$ -local Clifford Hamiltonian.

Our proof system also employs a new encoding scheme for quantum states, which we construct by extending the *trap scheme* proposed in [11]. While our new scheme can be seen as a *quantum authentication scheme* (cf. [2, 5, 6]), it in addition allows performing arbitrary constant-qubit Clifford circuits and measuring in the computational basis directly on authenticated data without the need for auxiliary states. The only previously known scheme supporting this feature requires high-dimensional quantum systems (i.e., qudits rather than qubits) [6], which make it inconvenient in our setting where all quantum operations are on qubits.

**1.2. Overview of protocol and techniques.** A natural approach to constructing zero-knowledge proofs for QMA is to consider a quantum analogue of the Goldreich–Micali–Wigderson proof system for graph 3-coloring (which we will hereafter refer to as the GMW 3-coloring proof system), in which the prover commits to a 3-coloring of the input graph and reveals only the colors of the vertices corresponding to an edge randomly selected by the verifier. Let us focus in particular on the local Hamiltonian problem, and consider a proof system in which the prover holds a quantum witness state for an instance of this problem, commits to this witness, and receives the challenge from the verifier (which, let us say, is a randomly chosen term of the local Hamiltonian). The prover might then open the commitments of the set of qubits on which the term acts nontrivially so that the verifier can measure the local energy for this term and determine acceptance accordingly.

There is a major difficulty when one attempts to carry out such an approach for QMA. The zero-knowledge property of the GMW 3-coloring proof system depends crucially on a structural property of the problem: the honest prover is free to randomize the three colors used in its coloring, and when the commitments to the colors of two neighboring vertices are revealed, the verifier will see just a uniform mixture over all pairs of different colors. This uniformity of the coloring marginals is important in achieving the zero-knowledge property of the proof system. Unlike the case of 3-coloring, however, none of the known QMA-complete problems under Karp reductions has such desirable properties. For example, if we use local Hamiltonian problems directly in a GMW-type proof system, of the sort suggested above, information about the reduced state of the quantum witness will be leaked to the verifier, possibly violating the zero-knowledge requirement.

To overcome the difficulty suggested above, we employ several ideas that enable the prover to “partially” open the commitments, revealing only the fact that the committed state is supported on certain subspaces. Our first technique simplifies

the verification circuit for QMA-complete problems through the introduction of the local Clifford–Hamiltonian problem that was already described. More specifically, our formulation of this problem requires every Hamiltonian term to take the form  $C^*|0^k\rangle\langle 0^k|C$  for some Clifford operation  $C$ . Because the local Clifford–Hamiltonian problem remains QMA-complete, it implies a random Clifford verification procedure for problems in QMA: intuitively, the verification of a quantum witness has been simplified to a Clifford measurement followed by a classical verification.

The Clifford verification procedure works in harmony with the encryption of quantum data via the quantum one-time pad and other derived hybrid schemes that are used by our proof system. This has the important effect of transforming statements about quantum states into statements about the classical keys of the quantum one-time pad, which naturally leads to our second main idea: the use of zero-knowledge proofs for NP against quantum attacks to simplify the construction of zero-knowledge proof systems for QMA. In our protocol, the verifier measures the encrypted quantum data and asks the prover to prove, using a zero-knowledge protocol for NP, that the decryption of this result is consistent with the verifier accepting.

In fact, if the verifier measures the quantum data according to the specifications of the protocol, the combination of the Clifford verification and the use of zero-knowledge proofs for NP suffices. A problem arises, however, if the verifier does not perform the honest measurement. Our third technique, inspired by work on quantum authentication [2, 6, 11, 17], employs a new scheme for encoding quantum states. Roughly speaking, if the prover encodes a witness state under our encoding scheme, then the verifier is essentially forced to perform the measurement honestly—any attempt to fake a “logically different” measurement result will succeed with negligible probability. In our proof system, we adapt the trap scheme proposed in [11] so that we can perform any constant-sized Clifford operations on authenticated quantum data followed by computational basis measurements, benefiting along the way from ideas concerning quantum computation on authenticated quantum data.

The resulting zero-knowledge proof system for QMA has a similar overall structure to the GMW 3-coloring protocol: the prover encodes the quantum witness state using a quantum authentication scheme, and sends the encoded quantum data together with a commitment to the secret keys of the authentication to the verifier. The verifier randomly samples a term  $C^*|0^k\rangle\langle 0^k|C$  in the local Clifford–Hamiltonian problem, applies the operation  $C$  transversally on the encoded quantum data, and measures all qubits corresponding to the  $k$  qubits of the selected term in the computational basis, and sends the measurement outcomes to the prover. The prover and verifier then invoke a quantum-secure zero-knowledge proof for the NP statement that the commitment correctly encodes an authentication key and, under this key, the verifier’s measurement outcomes do not decode to  $0^k$ .

**1.3. Comparisons to related work.** There has been other work on quantum complexity and theoretical cryptography, some of which is discussed below, that allows one to conclude statements having some similarity to our results. We will argue, however, that with respect to the problem of devising zero-knowledge quantum interactive proof systems for QMA, our main result is stronger in almost all respects. In addition, we believe that our proof system is appealing both because it is conceptually simple and represents a natural extension of well-known classical methods.

1. *Zero-knowledge proofs for all of IP.* Hallgren et al. [32] proved that, under certain technical conditions, any classical zero-knowledge proof system can be made secure against malicious quantum verifiers. A well-known result

of Ben-Or et al. [7] establishes that any problem in IP has a classical zero-knowledge protocol under a suitable cryptographic assumption. Although we have not verified that this zero-knowledge protocol for IP satisfies the technical conditions required by Hallgren et al. [32], we suspect that this is the case, assuming the existence of a quantum computationally hiding commitment scheme. If indeed this is so, it implies the existence of a classical protocol that is zero-knowledge against malicious quantum verifiers for all IP and, hence, for QMA because QMA is contained in IP. However, this generic protocol requires a computationally *unbounded* prover to carry out the honest protocol, and it is unlikely to allow for a reduced round complexity (e.g., constant round with constant soundness error) without causing unexpected consequences in complexity theory [27, 29, 56].

2. *Secure two-party computations.* One alternative approach to constructing zero-knowledge proof systems for QMA is to apply the general tool of secure two-party quantum computation [6, 16, 17]. In particular, we may imagine two parties, a prover and a verifier, jointly evaluating the verification circuit of a QMA problem, with the prover holding a quantum witness as its private input. In principle, one can design a two-party computation protocol so that the verifier learns the validity of the statement but nothing more about the prover's private input. While we believe that a careful analysis could make this approach work, it comes at a steep cost. First, we need to make significantly stronger computational assumptions, as secure quantum two-party computation relies on (at least) secure computations of classical functions against quantum adversaries. The best-known quantum-secure protocols for classical two-party computation assume quantum-secure dense public-key encryption [33] or similar primitives [42], in contrast to the existence of a quantum computationally hiding commitment scheme.<sup>2</sup> Second, the protocol obtained this way is an *argument* system. That is, the protocol is sound only against computationally bounded dishonest provers. Moreover, the generic quantum two-party computation protocol evaluates the verification circuit gate by gate and, in particular, interactions are unavoidable for some (non-Clifford) gates. This causes the round complexity to grow in proportion to the size of the verification circuit. In addition, the communications are inherently quantum, which makes the protocol much more demanding from a technological viewpoint.

On the positive side, through this approach, it is possible to achieve a negligible soundness error using just one copy of the witness state. In contrast, our proof system directly inherits the soundness error of the most natural and direct verification for the local Clifford–Hamiltonian problem (i.e., randomly select a Hamiltonian term and measure). If one reduces an arbitrary QMA-verification procedure to an instance of this problem, the resulting soundness guarantee will generally be weakened by this reduction.

3. *Zero-knowledge proofs for density matrix consistency.* It was pointed out by Liu [41] that the density matrix consistency problem, which asks if there exists a global state of  $n$  qubits that is consistent with a collection of  $k$ -qubit density matrix marginals, should admit a simple zero-knowledge proof system following the GMW 3-coloring approach. (See also [13] for further details

<sup>2</sup>Roughly speaking, this distinction is analogous to “cryptomania” vs “minicrypt” according to Impagliazzo’s five-world paradigm [35].

regarding this claim.) While it approaches our main result, it does not necessarily admit a zero-knowledge proof system for all problems in QMA, as the density matrix consistency problem is only known to be hard for QMA with respect to Cook reductions.

4. *Verification for QMA, nonlocal games, and follow-up work.* We note that Clifford verification with classical postprocessing of QMA was considered in [44] using magic states as ancillary resources. Our construction is arguably simpler, uses only constant-size Clifford operations and, most importantly, does not require any resource states. This helps one to avoid checking the correctness of resource states in the final zero-knowledge protocol. Our Clifford–Hamiltonian verification also finds applications in offering an alternative proof of the single-qubit measurement verification for QMA in [45], as well as in the study of nonlocal games [20, 36]. Moreover, following a previous version of this work [12], Vidick and Zhang [52] showed that our techniques can be applied to the conceptually simple QMA-complete “XY Hamiltonian problem” [19]. They obtain a *classical* zero-knowledge *argument* system for QMA, at the cost of sacrificing perfect completeness, and assuming the prover is given polynomially many copies of the witness state.

**Organization.** Section 2 summarizes notation, definitions, and primitives that are used for the construction of our zero-knowledge proof system. Section 3 describes the variant of the local Hamiltonian problem mentioned above. We present our zero-knowledge proof system for QMA in section 4 and prove its completeness and soundness in section 5 and zero-knowledge property in section 6. We conclude with some remarks and future directions in section 7.

**2. Preliminaries.** This section summarizes some of the notation, definitions, and known facts concerning quantum information and computation, cryptography, and other topics that are used throughout the paper. We refer to [40, 48, 57] for further details on the theory of quantum information and computation. Further information on classical zero-knowledge and cryptography can be found in [22, 23].

**2.1. Basic terminology.** Throughout the paper we let  $\Sigma = \{0, 1\}$  denote the binary alphabet, and only consider strings, promise problems, and complexity classes over this alphabet. For a string  $x \in \Sigma^*$ ,  $|x|$  denotes its length. A function  $g : \mathbb{N} \rightarrow \mathbb{N}$  is a *polynomially bounded function* if there exists a deterministic polynomial-time Turing machine  $M_g$  that outputs  $1^{g(n)}$  on input  $1^n$  for every nonnegative integer  $n$ . A function  $f : \mathbb{N} \rightarrow [0, \infty)$  is said to be *negligible* if, for every polynomially bounded function  $g$ , it holds that  $f(n) < 1/g(n)$  for all but finitely many values of  $n$ .

**2.2. Quantum information basics.** When we refer to a *quantum register* in this paper, we simply mean a collection of qubits that we wish to view as a single unit and to which we give some name. Names of registers will always be uppercase letters in a *sans serif* font, such as X, Y, and Z. The finite-dimensional complex Hilbert spaces associated with registers will be denoted by capital script letters such as  $\mathcal{X}$ ,  $\mathcal{Y}$ , and  $\mathcal{Z}$ , using the same letter in the two different fonts to denote a quantum register and its corresponding space for convenience. Dirac notation is used to express vectors in Hilbert spaces and linear mappings between them in a standard way.

For a given space  $\mathcal{X}$ , we let  $L(\mathcal{X})$  denote the set of all linear mappings (or *operators*) from  $\mathcal{X}$  to itself. The identity element of  $L(\mathcal{X})$  is denoted  $\mathbb{1}_{\mathcal{X}}$ , or just as  $\mathbb{1}$  when  $\mathcal{X}$  can be taken as implicit. The inner product between operators  $A$  and  $B$  is defined as  $\langle A, B \rangle = \text{Tr}(A^* B)$ .



*Quantum states* are represented by density operators, which are positive semidefinite operators having unit trace. Under the assumption that  $\mathcal{X}$  corresponds to  $n$  qubits, a linear map  $\Phi : L(\mathcal{X}) \rightarrow L(\mathcal{Y})$  is *completely positive* if and only if its Choi operator

$$(2.1) \quad J(\Phi) = \sum_{x,y \in \Sigma^n} \Phi(|x\rangle\langle y|) \otimes |x\rangle\langle y|$$

is positive semidefinite, and  $\Phi$  is said to be a *channel* if it is both completely positive and preserves trace. Channels are mappings from density operators to density operators that, in principle, represent physically realizable operations. A *measurement* is described by a collection of positive semidefinite operators  $\{M_j\}$  such that  $\sum_j M_j = \mathbb{1}$ , with the probability that the measurement on state  $\rho$  results in outcome  $j$  being given by  $\langle M_j, \rho \rangle$ .

We review a few definitions of norms on operators, which are used to discuss the distinguishability of quantum states and channels. The *trace norm* of an operator  $X \in L(\mathcal{X})$  is defined as  $\|X\|_1 = \text{Tr} \sqrt{X^* X}$ . For any linear map  $\Phi : L(\mathcal{X}) \rightarrow L(\mathcal{Y})$ , the *diamond norm* (or completely bounded trace norm) [3, 39, 40] is defined as

$$\|\Phi\|_\diamond = \max \{ \|(\Phi \otimes \mathbb{1}_{L(\mathcal{W})})(X)\|_1 : X \in L(\mathcal{X} \otimes \mathcal{W}), \|X\|_1 \leq 1 \},$$

where  $\mathcal{W}$  is any space with dimension equal to that of  $\mathcal{X}$ . (The value remains the same for any choice of  $\mathcal{W}$ , provided its dimension is at least that of  $\mathcal{X}$ .)

**Quantum gates and circuits.** A *quantum circuit* is an acyclic network of quantum gates connected by wires. The quantum gates represent quantum channels while the wires represent qubits on which the channels act.

We will refer to two types of quantum circuits in this paper: *unitary* quantum circuits and *general* quantum circuits. By unitary quantum circuits we mean circuits composed of unitary gates (such as the ones described below) chosen from some finite gate set. General quantum circuits are composed of gates that may correspond to channels that are not necessarily unitary. It is sufficient for the purposes of this paper that we consider just two simple nonunitary gates: *ancillary gates*, which input nothing and output a qubit in the  $|0\rangle$  state; and *erasure gates*, which input one qubit and output nothing (and correspond to the channel described by the trace mapping). As is described elsewhere [3, 59], arbitrary channels mapping one register to another can always be approximated arbitrarily closely by quantum circuits whose gates include a universal collection of unitary gates together with ancillary and erasure gates. The *size* of a quantum circuit is the number of gates in the circuit plus the number of qubits on which it acts. We will refer specifically to the following well-known single-qubit unitary gates:

1. *Pauli gates*:

$$(2.2) \quad X : |a\rangle \mapsto |1-a\rangle \quad \text{and} \quad Z : |a\rangle \mapsto (-1)^a |a\rangle$$

for each  $a \in \Sigma$ , as well as  $Y = iXZ$ .

2. *Hadamard gate*:

$$(2.3) \quad H : |a\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^a}{\sqrt{2}}|1\rangle$$

for each  $a \in \Sigma$ .

## 3. Phase gate:

$$(2.4) \quad P : |a\rangle \mapsto i^a |a\rangle$$

for each  $a \in \Sigma$ .

In addition, for any  $k$ -qubit unitary quantum gate  $U$  we define the *controlled- $U$*  gate as

$$(2.5) \quad \Lambda(U) : |a\rangle|x\rangle \mapsto |a\rangle U^a |x\rangle$$

for each  $a \in \Sigma$  and  $x \in \Sigma^k$ .

The  $k$ -qubit *Pauli group* is the group containing all unitary operators of the form

$$(2.6) \quad \alpha U_1 \otimes \cdots \otimes U_k,$$

where  $\alpha \in \{1, i, -1, -i\}$  and  $U_1, \dots, U_k \in \{\mathbb{1}, X, Y, Z\}$ , where  $\mathbb{1}$  denotes the single-qubit identity operation. Elements of this group are also referred to as *Pauli operations*. If  $a, b \in \Sigma^k$  are binary strings of length  $k$ , then we write

$$(2.7) \quad X^a = X^{a_1} \otimes \cdots \otimes X^{a_k} \quad \text{and} \quad Z^b = Z^{b_1} \otimes \cdots \otimes Z^{b_k}$$

to denote the Pauli operations obtained from these strings as indicated.

Channels that can be expressed as convex combinations of unitary channels that correspond to Pauli operations are called *Pauli channels*. An example of a Pauli channel that is relevant to this paper is the *completely depolarizing* channel

$$(2.8) \quad \Omega(\rho) = \frac{1}{4} \sum_{a,b \in \Sigma} (X^a Z^b) \rho (X^a Z^b)^* = \frac{\mathbb{1}}{2}$$

for any single-qubit density operator  $\rho$ . We thus see that the effect of  $\Omega$  is to completely randomize the state of a single-qubit system. By treating a random choice of a pair  $(a, b)$  as a secret key, we obtain a quantum generalization of the one-time pad, known as the *quantum one-time pad* [4]. When the channel is performed independently on  $k$  qubits, the effect is given by

$$(2.9) \quad \Omega^{\otimes k}(\rho) = 2^{-k} \mathbb{1} \otimes \cdots \otimes \mathbb{1}$$

for every  $k$ -qubit density operator  $\rho$ . The quantum one-time pad generalizes naturally to any choice of the number  $k$ .

Sometimes it will be convenient to consider quantum circuits that implement measurements. When we refer to a *measurement circuit*, we mean any general quantum circuit, followed by a measurement of all of its output qubits with respect to the standard basis. If  $Q$  is a measurement circuit that is applied to a collection of qubits in the state  $\rho$ , then  $Q(\rho)$  is interpreted as a string-valued random variable describing the resulting measurement. We will only need to refer to measurement circuits outputting a single bit in this paper.

A  $k$ -qubit *Clifford circuit* is any unitary quantum circuit on  $k$  qubits whose gates are drawn from the set  $\{H, P, \Lambda(X)\}$  containing Hadamard, phase, and controlled-not gates. (It is common that one also allows Pauli gates to be included in this set for convenience. Given that  $X = HPPH$  and  $Z = PP$ , there is no generality lost in using the smaller gate set in the definition.) The set of all unitary operators that can be described by  $k$ -qubit Clifford circuits forms a finite group known as the *Clifford group*.

Up to scalar multiples, the  $k$ -qubit Clifford group is the normalizer of the  $k$ -qubit Pauli group: if  $U$  is a  $k$ -qubit unitary operator for which it holds that  $UVU^*$  is an element of the  $k$ -qubit Pauli group for every  $k$ -qubit Pauli group element  $V$ , then  $U = \alpha C$  for  $\alpha \in \mathbb{C}$  satisfying  $|\alpha| = 1$  and  $C$  being a  $k$ -qubit Clifford group element. Given the description of a  $k$ -qubit Pauli group element  $V$  and a  $k$ -qubit Clifford circuit  $C$ , one can efficiently compute a description of the  $k$ -qubit Pauli group element  $CVC^*$  [31].

Clifford circuits are not universal for quantum computation. Two examples (among other known examples) of universal gate sets are the following:

1. Hadamard, phase, and Toffoli gates:  $\{H, P, \Lambda(\Lambda(X))\}$ .
2. Hadamard and controlled-phase gates:  $\{H, \Lambda(P)\}$ .

The first of these choices is sometimes easier to work with, but we will make use of the fact that the second gate set is universal in this work.

### 2.3. Polynomial-time generated families of quantum circuits and QMA.

Any quantum circuit with gates drawn from a fixed, finite gate set can be encoded as a binary string, with respect to a variety of possible encoding schemes. The specific details of such encoding schemes are not important within the context of this paper, so we will leave it to the reader to imagine that a sensible and efficient encoding scheme for quantum circuits has been selected, relative to whatever gate set is under consideration. It should be assumed, of course, that a circuit's size and its encoding length are polynomially related.

For any infinite set of binary strings  $S \subseteq \Sigma^*$ , a collection  $\{V_x : x \in S\}$  of quantum circuits is said to be *polynomial-time generated* if there exists a deterministic polynomial-time Turing machine that, on input  $x \in S$ , outputs an encoding of  $V_x$ . The assumptions on encoding schemes suggested above imply that, if  $\{V_x : x \in S\}$  is a polynomial-time generated collection, then  $V_x$  must have size polynomial in  $|x|$ .

Next we will define the complexity class QMA, which is commonly viewed as the most natural quantum generalization of NP.

**DEFINITION 2.1.** *A promise problem  $A = (A_{yes}, A_{no})$  is contained in the complexity class  $\text{QMA}_{\alpha, \beta}$  if there exists a polynomial-time generated collection*

$$(2.10) \quad \{V_x : x \in A_{yes} \cup A_{no}\}$$

*of quantum circuits and a polynomially bounded function  $p$  possessing the following properties:*

1. *For every string  $x \in A_{yes} \cup A_{no}$ , one has that  $V_x$  is a measurement circuit taking  $p(|x|)$  input qubits and outputting a single bit.*
2. *Completeness. For all  $x \in A_{yes}$ , there exists a  $p(|x|)$ -qubit state  $\rho$  such that  $\Pr(V_x(\rho) = 1) \geq \alpha$ .*
3. *Soundness. For all  $x \in A_{no}$  and all  $p(|x|)$ -qubit states  $\rho$ ,  $\Pr(V_x(\rho) = 1) \leq \beta$ .*

In this definition,  $\alpha, \beta \in [0, 1]$  may be constant values or functions of the length of the input string  $x$ . When they are omitted, it is to be assumed that they are  $\alpha = 2/3$  and  $\beta = 1/3$ . Known error reduction methods [40, 43, 46] imply that a wide range of selections of  $\alpha$  and  $\beta$  give rise to the same complexity class. In particular,  $\text{QMA}_{\alpha, \beta}$  coincides with  $\text{QMA}_{\alpha, \beta}$  for  $\alpha = 1 - 2^{-q(|x|)}$  and  $\beta = 2^{-q(|x|)}$  for any polynomially bounded function  $q$ .

### 2.4. Quantum computational indistinguishability and zero-knowledge.

Next we review notions of quantum state and channel discrimination, as well as zero-knowledge in a quantum setting (as defined in [58]).

We first specify what it means for two collections of quantum states to be quantum computationally indistinguishable. The definition that follows may be viewed as being a nonuniform notion of quantum computational indistinguishability, as it places no uniformity conditions on quantum circuits and allows for an *auxiliary* quantum state  $\sigma$  to assist in the task of state discrimination.

**DEFINITION 2.2** (quantum computationally indistinguishable states). *Let  $S$  be an infinite set of binary strings, let  $r$  be a polynomially bounded function, and let  $\rho_x$  and  $\xi_x$  be states on  $r(|x|)$  qubits for each  $x \in S$ . The collections  $\{\rho_x : x \in S\}$  and  $\{\xi_x : x \in S\}$  are quantum computationally indistinguishable if, for every choice of polynomially bounded functions  $s$  and  $k$ , there exists a negligible function  $\varepsilon$  such that the following property holds for every string  $x \in S$ : for every  $k(|x|)$ -qubit state  $\sigma$  and every measurement circuit  $Q$  on  $r(|x|) + k(|x|)$  qubits having size  $s(|x|)$ , it is the case that*

$$(2.11) \quad |\Pr[Q(\rho_x \otimes \sigma) = 1] - \Pr[Q(\xi_x \otimes \sigma) = 1]| \leq \varepsilon(|x|).$$

This notion extends naturally to distinguishing collections of channels, as the following definition makes precise.

**DEFINITION 2.3** (quantum computationally indistinguishable channels). *Let  $S$  be an infinite set of binary strings, let  $q$  and  $r$  be polynomially bounded functions, and let  $\Phi_x$  and  $\Psi_x$  be channels from  $q(|x|)$  qubits to  $r(|x|)$  qubits for each  $x \in S$ . The collections  $\{\Phi_x : x \in S\}$  and  $\{\Psi_x : x \in S\}$  are quantum computationally indistinguishable if, for every choice of polynomially bounded functions  $s$  and  $k$ , there exists a negligible function  $\varepsilon$  such that the following property holds for every string  $x \in S$ : for every state  $\sigma$  on  $q(|x|) + k(|x|)$  qubits and every measurement circuit  $Q$  on  $r(|x|) + k(|x|)$  qubits having size  $s(|x|)$ , it is the case that*

$$(2.12) \quad |\Pr[Q((\Phi_x \otimes \mathbb{1})(\sigma)) = 1] - \Pr[Q((\Psi_x \otimes \mathbb{1})(\sigma)) = 1]| \leq \varepsilon(|x|).$$

We will also make use of statistical notions of indistinguishability for states and channels, which are defined as follows.

**DEFINITION 2.4** (statistically indistinguishable states). *Let  $S$  be an infinite set of binary strings, let  $r$  be a polynomially bounded function, and let  $\rho_x$  and  $\xi_x$  be states on  $r(|x|)$  qubits for each  $x \in S$ . The collections  $\{\rho_x : x \in S\}$  and  $\{\xi_x : x \in S\}$  are statistically indistinguishable if there exists a negligible function  $\varepsilon$  such that, for all  $x \in S$ ,*

$$(2.13) \quad \frac{1}{2} \|\rho_x - \xi_x\|_1 \leq \varepsilon(|x|).$$

**DEFINITION 2.5** (statistically indistinguishable channels). *Let  $S$  be an infinite set of binary strings, let  $q$  and  $r$  be polynomially bounded functions, and let  $\Phi_x$  and  $\Psi_x$  be channels from  $q(|x|)$  qubits to  $r(|x|)$  qubits for each  $x \in S$ . The collections  $\{\Phi_x : x \in S\}$  and  $\{\Psi_x : x \in S\}$  are statistically indistinguishable if there exists a negligible function  $\varepsilon$  such that, for all  $x \in S$ ,*

$$(2.14) \quad \frac{1}{2} \|\Phi_x - \Psi_x\|_{\diamond} \leq \varepsilon(|x|).$$

Next we review the definition of quantum computational zero-knowledge proof systems as defined in [58]. Let  $(P, V)$  be a quantum or classical interactive proof

system for a promise problem  $A$ . An arbitrary (possibly malicious) verifier  $V'$  is any quantum computational process that interacts with  $P$  according to the structural specification of  $(P, V)$ . Similarly to the classical notion of auxiliary input zero-knowledge, a verifier  $V'$  will take, in addition to the input string  $x$ , an auxiliary input, and produce some output. This is crucial for the composition of zero-knowledge proof systems. The most general situation allowed by quantum information theory is that both the auxiliary input and the output are quantum, meaning that the verifier operates on quantum registers whose initial state is arbitrary and may be entangled with some external system. Also similarly to the classical case, we will assume that for any given polynomial-time verifier  $V'$  there exist polynomially bounded functions  $q$  and  $r$  that determine the number of auxiliary input qubits and output qubits of  $V'$ . To say that  $V'$  is a polynomial-time verifier means that the entire action of  $V'$  must be described by some polynomial-time generated family of quantum circuits.

The interaction of a verifier  $V'$  with  $P$  on input  $x$  induces some channel from the verifier's  $q(|x|)$  auxiliary input qubits to  $r(|x|)$  output qubits. Let  $\mathcal{W}$  denote the vector space corresponding to the auxiliary input qubits, let  $\mathcal{Z}$  denote the space corresponding to the output qubits, and let  $\Phi_x : L(\mathcal{W}) \rightarrow L(\mathcal{Z})$  denote the resulting channel induced by the interaction of  $V'$  with  $P$  on input  $x$ . A simulator  $S$  for a given verifier  $V'$  is described by a polynomial-time generated family of general quantum circuits that agrees with  $V'$  on the functions  $q$  and  $r$  representing the number of auxiliary input qubits and output qubits, respectively. Such a simulator does not interact with  $P$ , but simply induces a channel that we will denote by  $\Psi_x : L(\mathcal{W}) \rightarrow L(\mathcal{Z})$  on each input  $x$ .

**DEFINITION 2.6** (quantum computational zero-knowledge). *An interactive proof system  $(P, V)$  for a promise problem  $A$  is quantum computational zero-knowledge if, for every polynomial-time generated quantum verifier  $V'$ , there exists a polynomial-time generated quantum simulator  $S$  that satisfies the following requirements:*

1. *The verifier  $V'$  and simulator  $S$  agree on the polynomially bounded functions  $q$  and  $r$  that specify the number of auxiliary input qubits and output qubits, respectively.*
2. *Let  $\Phi_x$  be the channel that results from the interaction between  $V'$  and  $P$  on input  $x$ , and let  $\Psi_x$  be the channel induced by the simulator  $S$  on input  $x$ , both as described above. Then the collections  $\{\Phi_x : x \in A_{\text{yes}}\}$  and  $\{\Psi_x : x \in A_{\text{yes}}\}$  are quantum computationally indistinguishable.*

**2.5. Cryptographic tools.** In this section we introduce cryptographic building blocks that are useful in our proof system. We emphasize that, as is typical in the classical setting, we formulate all computational security properties (e.g., concealing in a commitment scheme) with respect to nonuniform quantum adversaries, which provides more stringent security requirements and is crucial in many security proofs.

**Commitment schemes.** Our definition for quantum computationally secure commitment schemes is as follows. We note explicitly that this is a noninteractive definition: all messages are from a sender to a receiver.

**DEFINITION 2.7** (quantum computationally secure commitments). *A quantum computationally secure commitment scheme for an alphabet  $\Gamma$  is a collection of polynomial-time computable functions  $\{f_n : n \in \mathbb{N}\}$  taking the form*

$$(2.15) \quad f_n : \Gamma \times \Sigma^{p(n)} \rightarrow \Sigma^{q(n)}$$

*for polynomially bounded functions  $p$  and  $q$ , such that the following conditions hold:*

1. Unconditionally binding property. For every choice of  $n \in \mathbb{N}$ ,  $a, b \in \Gamma$ , and  $r, s \in \Sigma^{p(n)}$ , one has that  $f_n(a, r) = f_n(b, s)$  implies  $a = b$ .
2. Quantum computationally concealing property. For every  $a \in \Gamma$  and  $n \in \mathbb{N}$ , define

$$(2.16) \quad \rho_{a,n} = \frac{1}{2^{p(n)}} \sum_{r \in \Sigma^{p(n)}} |f_n(a, r)\rangle \langle f_n(a, r)|.$$

For every choice of  $a, b \in \Gamma$  the ensembles  $\{\rho_{a,n} : n \in \mathbb{N}\}$  and  $\{\rho_{b,n} : n \in \mathbb{N}\}$  are quantum computationally indistinguishable.

Such a bit commitment scheme (i.e.,  $\Gamma = \{0, 1\}$ ) can be constructed based on certain quantum intractability assumptions. As shown in [1], it suffices to have quantum-resistant one-way *permutations*, which are permutations that can be computed efficiently on a classical computer but are hard to invert for both classical and quantum polynomial-time algorithms. The same commitment scheme remains quantum-secure based on a slightly weaker assumption of quantum-resistant *injective* one-way functions. To commit to a string, one can independently use the commitment described above bit by bit.

Based on a quantum-secure commitment scheme, we can obtain the other two essential cryptographic building blocks in our protocol: a zero-knowledge proof system for NP and a coin-flipping protocol, both secure against quantum adversaries.

**Zero-knowledge proof for NP.** In [58] it was proved that the GMW 3-coloring protocol [26] remains zero-knowledge in the presence of quantum verifiers, assuming the existence of a quantum computationally secure commitment scheme. This means that we have a classical zero-knowledge proof system for any NP language that is secure against arbitrary polynomial-time quantum verifiers.

**Coin flipping.** A coin-flipping protocol is an interactive process that allows two parties to jointly toss random coins. It is not necessary for us to consider this notion generally, as we only make use of one specific coin-flipping protocol, namely, Blum's coin-flipping protocol [8] in which an honest prover commits to a random  $y \in \Sigma$ , the honest verifier selects  $z \in \Sigma$  at random, the prover reveals  $y$ , and the two participants agree that the random bit generated is  $r = y \oplus z$ .

Damgård and Lunemann [15] proved, assuming the existence of a quantum-secure commitment scheme, that Blum's coin-flipping protocol is quantum-secure. This protocol generates one random coin, and we will need to flip logarithmically many random bits. A simple way of achieving this is by sequential repetition, but more effectively it is possible to extend the analysis of Damgård and Lunemann and show that parallel repetition of Blum's protocol logarithmically many times remains quantum-secure.

**2.6. Concatenated Steane codes.** The last topic to be discussed in this section concerns the existence of quantum error-correcting codes having certain properties that are important to the functioning of our zero-knowledge proof system for QMA. There are multiple choices of codes that satisfy our requirements, but in the interest of simplicity we will describe just one specific family of codes in this category.

These codes are based on the 7-qubit *Steane code* [51], in which one qubit is encoded into 7 qubits by the following action on standard basis states:

$$(2.17) \quad |0\rangle \mapsto \frac{1}{\sqrt{8}} \sum_{x \in \mathcal{D}_7^0} |x\rangle \quad \text{and} \quad |1\rangle \mapsto \frac{1}{\sqrt{8}} \sum_{x \in \mathcal{D}_7^1} |x\rangle,$$

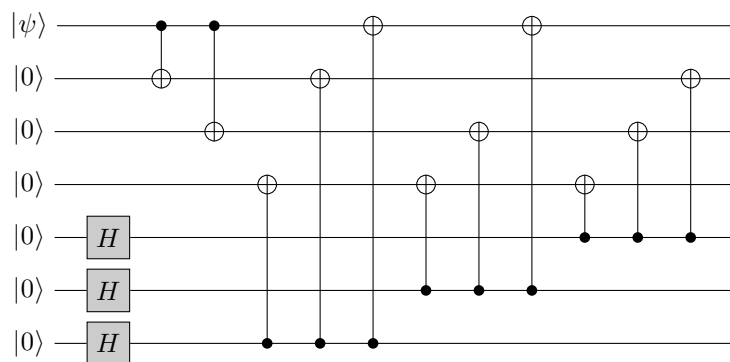


FIG. 2.1. A Clifford circuit encoder for the 7-qubit Steane code. Hereafter we will write  $U_7$  to refer to the unitary operator on 7 qubits described by this circuit.

where

$$\begin{aligned}\mathcal{D}_7^0 &= \{0000000, 0001111, 0110011, 0111100, 1010101, 1011010, 1100110, 1101001\}, \\ \mathcal{D}_7^1 &= \{0010110, 0011001, 0100101, 0101010, 1000011, 1001100, 1110000, 1111111\}.\end{aligned}$$

It is the case that  $\mathcal{D}_7^0$  is a  $[7, 4]$ -Hamming code, while

$$(2.18) \quad \mathcal{D}_7 = \mathcal{D}_7^0 \cup \mathcal{D}_7^1$$

is the dual code to  $\mathcal{D}_7^0$  (i.e., it is the code consisting of all binary strings of length 7 whose inner product with any codeword in  $\mathcal{D}_7^0$  is even). This is an example of a *CSS code* [48], and it is capable of correcting single-qubit errors. The standard error-correcting procedure, which we do not actually need in this paper, is to first reversibly correct errors in the standard basis, with respect to the code  $\mathcal{D}_7$ , and then to do the same with respect to the diagonal basis. The 7-qubit Clifford circuit depicted in Figure 2.1 encodes one qubit into 7 with respect to this code, assuming 6 qubits in the  $|0\rangle$  state are made available.

One of the properties of the 7-qubit Steane code that is important from the viewpoint of this paper is that it admits a *transversal* application of Clifford operations, in the sense that is explained in Figure 2.2.

Note that by concatenating the 7-qubit Steane code with itself, one obtains a code having similar properties to the 7-qubit code and, in addition, having a large minimum distance for the underlying code. More specifically, suppose that  $N = 7^t$  for  $t$  being an even positive integer. (We take  $t$  to be even for convenience, as this eliminates the entrywise complex conjugation on Clifford operations induced by their transversal application.) By concatenating the 7-qubit Steane code to itself  $t$  times, one obtains a quantum error-correcting code in which one qubit is encoded into  $N$  qubits in the following way:

$$(2.19) \quad |0\rangle \mapsto \frac{1}{\sqrt{8^t}} \sum_{x \in \mathcal{D}_N^0} |x\rangle \quad \text{and} \quad |1\rangle \mapsto \frac{1}{\sqrt{8^t}} \sum_{x \in \mathcal{D}_N^1} |x\rangle,$$

where  $\mathcal{D}_N^0, \mathcal{D}_N^1 \subseteq \Sigma^N$  are related in a way that generalizes the case  $N = 7$ . In particular,  $\mathcal{D}_N^0$  is a binary linear code having  $8^t$  elements, and whose dual code takes the form  $\mathcal{D}_N = \mathcal{D}_N^0 \cup \mathcal{D}_N^1$  for  $\mathcal{D}_N^1 \subseteq \Sigma^N$  being a coset of  $\mathcal{D}_N^0$ .



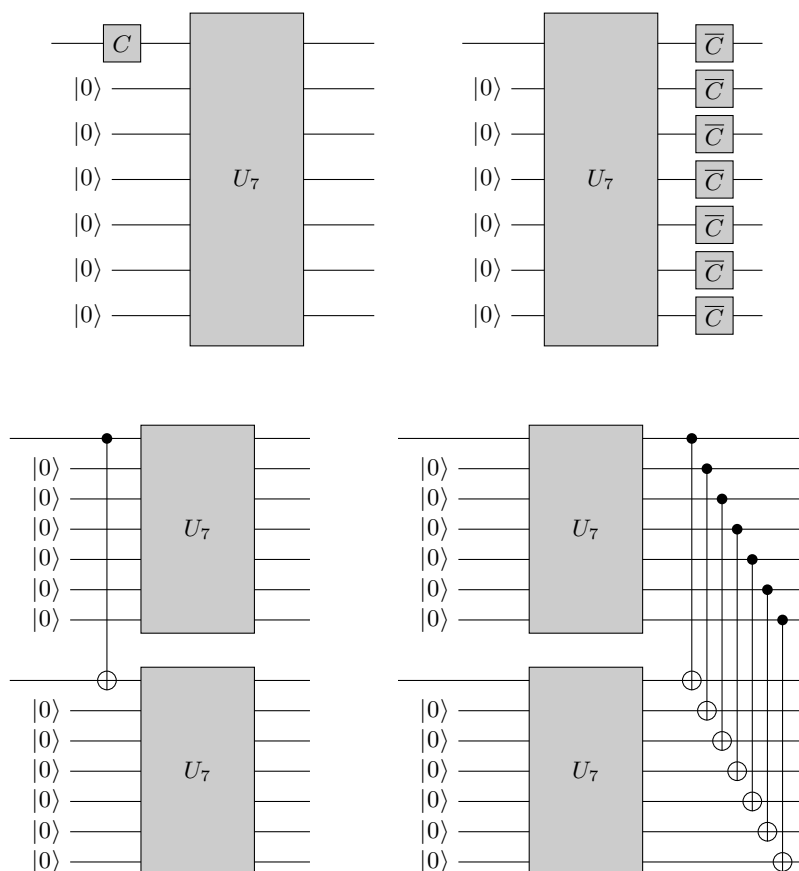


FIG. 2.2. The 7-qubit Steane code allows for the transversal application of Clifford operations. That is, the circuits on the left are equivalent to the corresponding circuits on the right. In general, the application of any Clifford operation on  $k$  qubits prior to being encoded is equivalent to the entrywise complex conjugate of that Clifford operation being applied 7 times to the  $7k$  qubits that encode the original  $k$  qubits.

The  $t$ -fold concatenation of the 7-qubit Steane code inherits the properties of the 7-qubit Steane code mentioned above. A Clifford circuit  $U_N$  acting on  $N$  qubits,  $N-1$  of which are to be initialized in the  $|0\rangle$  state, performs the encoding. This circuit is obtained by creating a tree from multiple copies of the circuit  $U_7$  in the natural way. The code allows for Clifford operations to be applied transversally.

An added feature of the concatenated versions of the 7-qubit Steane code is that it corrects more errors than the ordinary 7-qubit code. In particular, we will make use of the fact that the code  $\mathcal{D}_N$ , for  $N = 7^t$ , has minimum Hamming weight  $3^t$  for a nonzero code word. This allows one to obtain a polynomial-length code for any polynomial lower bound on the minimum nonzero Hamming weight of a code word.

**3. The local Clifford–Hamiltonian problem.** The local Hamiltonian problem is a well-known example of a complete problem for QMA, provided that certain assumptions are in place regarding the gap between the ground state energy (i.e., the smallest eigenvalue) of input Hamiltonians for yes- and no-inputs. A general and somewhat imprecise formulation of the local Hamiltonian problem is as follows.

*The  $k$ -local Hamiltonian ( $k$ -LH) problem.*

*Input:* A collection  $H_1, \dots, H_m$  of  $k$ -local Hamiltonian operators, each acting on  $n$  qubits and satisfying  $0 \leq H_j \leq \mathbb{1}$  for  $j = 1, \dots, m$ , along with real numbers  $\alpha$  and  $\beta$  satisfying  $\alpha < \beta$ .

*Yes:* There exists an  $n$ -qubit state  $\rho$  such that  $\langle \rho, H_1 + \dots + H_m \rangle \leq \alpha$ .

*No:* For every  $n$ -qubit state  $\rho$ , it holds that  $\langle \rho, H_1 + \dots + H_m \rangle \geq \beta$ .

This problem statement is imprecise in the sense that it does not specify how  $\alpha$  and  $\beta$  are to be represented or what requirements are placed on the gap  $\beta - \alpha$  mentioned above. We will be more precise about these issues when formulating a restricted version of this problem below, but it is appropriate that we first summarize what is already known.

It is known that  $k$ -LH is complete for QMA (with respect to Karp reductions) provided  $\alpha$  and  $\beta$  are input in a reasonable way and separated by an inverse polynomial gap; this was first proved by Kitaev, Shen, and Vyalı [40] for the case  $k = 5$ , then by Kempe and Regev [38] for  $k = 3$  and Kempe, Kitaev, and Regev [37] for  $k = 2$ . If one adds the additional requirement that  $\alpha$  is exponentially small, which will be important in the context of this paper, then QMA-completeness for  $k = 5$  still follows from Kitaev's proof, but the proofs of Kempe and Regev and Kempe, Kitaev, and Regev do not imply the same for  $k = 3$  and  $k = 2$ . On the other hand, the works of Bravyi [9] and Gosset and Nagaj [30] do establish QMA-completeness for exponentially small  $\alpha$  for  $k = 4$  and  $k = 3$ , respectively.

The restricted version of the local Hamiltonian we introduce is one in which each Hamiltonian term  $H_j$  is not only  $k$ -local and satisfies  $0 \leq H_j \leq \mathbb{1}$  but, furthermore, on the  $k$  qubits on which it acts nontrivially, its action must be given by a rank 1 projection operator of the form

$$(3.1) \quad C_j^* |0^k\rangle \langle 0^k| C_j$$

for some choice of a  $k$ -qubit Clifford operation  $C_j$ . For brevity, we will refer to any such operator as a  *$k$ -local Clifford–Hamiltonian projection*. The precise statement of our problem variant is as follows.

*The  $k$ -local Clifford–Hamiltonian ( $k$ -LCH) problem.*

*Input:* A collection  $H_1, \dots, H_m$  of  $k$ -local Clifford–Hamiltonian projections, along with positive integers  $p$  and  $q$  expressed in unary notation (i.e., as strings  $1^p$  and  $1^q$ ) and satisfying  $2^p > q$ .

*Yes:* There exists an  $n$ -qubit state  $\rho$  such that  $\langle \rho, H_1 + \dots + H_m \rangle \leq 2^{-p}$ .

*No:* For every  $n$ -qubit state  $\rho$ , it holds that  $\langle \rho, H_1 + \dots + H_m \rangle \geq 1/q$ .

It may be noted that, by the particular way we have stated this problem, we are focusing on a variant of the local Hamiltonian problem in which the parameter  $\alpha$  may be exponentially small and the gap  $\beta - \alpha$  is at least inverse polynomial.

**THEOREM 3.1.** *The 5-local Clifford–Hamiltonian problem is QMA-complete with respect to Karp reductions. Moreover, for any choice of a promise problem  $A \in \text{QMA}$  and a polynomially bounded function  $p$ , there exists a Karp reduction  $f$  from  $A$  to*

5-LCH having the form

$$(3.2) \quad f(x) = \langle H_1, \dots, H_m, 1^{p(|x|)}, 1^q \rangle$$

for every  $x \in A_{\text{yes}} \cup A_{\text{no}}$ .

*Proof.* The containment of the 5-local Clifford–Hamiltonian problem in QMA follows from the fact that the 5-LH problem is in QMA for the same choice of the ground state energy bounds. It therefore remains to prove the statement concerning the QMA-hardness of the 5-LCH problem.

Let  $A = (A_{\text{yes}}, A_{\text{no}})$  be any promise problem in QMA and let  $p$  be a polynomially bounded function. Using a standard error reduction procedure for QMA, one may conclude that there exists a polynomial-time generated collection

$$(3.3) \quad \{V_x : x \in A_{\text{yes}} \cup A_{\text{no}}\}$$

of measurement circuits having these properties:

1. If  $x \in A_{\text{yes}}$ , then there exists a state  $\rho$  such that  $V_x(\rho) = 1$  with probability  $1 - 2^{-p(|x|)}$ .
2. If  $x \in A_{\text{no}}$ , then for all quantum states  $\rho$  representing valid inputs to  $V_x$  it holds that  $V_x(\rho) = 1$  with probability at most  $1/2$ .

It is known that  $\{\Lambda(P), H\}$  is a universal gate set for quantum computation, so there is no loss of generality in assuming each  $V_x$  is a quantum circuit using gates from this set, together with a supply of ancillary qubits initialized to the state  $|0\rangle$ . For technical reasons (which are discussed later) we will assume something marginally stronger, which is that each  $V_x$  uses gates from the set  $\{\Lambda(P), H \otimes H\}$ . That is, every Hadamard gate appearing in  $V_x$  is paired with another Hadamard gate to be applied at the same time but on a different qubit. Note that for any circuit composed of gates from the set  $\{\Lambda(P), H\}$ , this stronger condition is easily met by adding to this circuit a number of additional Hadamard gates on an otherwise unused ancilla qubit.

Now consider the 5-local circuit-to-Hamiltonian construction of Kitaev, Shen, and Vyalı [40], for a given choice of  $V_x$ . In this construction, the resulting Hamiltonians have the form

$$(3.4) \quad H_{\text{total}} = H_{\text{in}} + H_{\text{out}} + H_{\text{clock}} + H_{\text{prop}},$$

where the terms check the initialization, readout, validity of unary clock, and propagation of computation, respectively. It follows from Kitaev's proof that, for  $x \in A_{\text{yes}}$ , the resulting Hamiltonian  $H_{\text{total}}$  has ground state energy at most  $2^{-p(|x|)}$ , and for  $x \in A_{\text{no}}$  the ground state energy of  $H_{\text{total}}$  is at least  $1/q(|x|)$ , for some polynomially bounded function  $q$ . To complete the proof, it suffices to demonstrate that each of these terms can be expressed as a sum of Clifford–Hamiltonian projections.

The first three terms,  $H_{\text{in}}$ ,  $H_{\text{out}}$ , and  $H_{\text{clock}}$ , can easily be expressed as sums of Clifford–Hamiltonian projections, as they are all projection operators that are diagonal in the standard basis. The propagation term has the form  $H_{\text{prop}} = \sum_{t=1}^T H_{\text{prop},t}$ , where each operator  $H_{\text{prop},t}$  takes the form

$$(3.5) \quad \begin{aligned} H_{\text{prop},t} &= \frac{1}{2} [ (|100\rangle\langle 100|_{t-1,t,t+1} + |110\rangle\langle 110|_{t-1,t,t+1}) \otimes \mathbb{1} \\ &\quad - |110\rangle\langle 100|_{t-1,t,t+1} \otimes U_t - |100\rangle\langle 110|_{t-1,t,t+1} \otimes U_t^* ] \\ &= |10\rangle\langle 10|_{t-1,t+1} \otimes \frac{1}{2} [\mathbb{1}_t \otimes \mathbb{1} - |1\rangle\langle 0|_t \otimes U_t - |0\rangle\langle 1|_t \otimes U_t^*]. \end{aligned}$$

Here, the first three qubits (indexed by  $t-1$ ,  $t$ , and  $t+1$ ) refer to qubits in a clock register and  $U_t$  represents the  $t$ th unitary gate in  $V_x$ . To prove that each propagation operator  $H_{\text{prop},t}$  can be expressed as a sum of Clifford–Hamiltonian projections, it suffices to prove the same for every projection of the form

$$(3.6) \quad \frac{1}{2} [\mathbb{1} \otimes \mathbb{1} - |1\rangle\langle 0| \otimes U - |0\rangle\langle 1| \otimes U^*]$$

for  $U$  being either  $\Lambda(P)$  or  $H \otimes H$ .

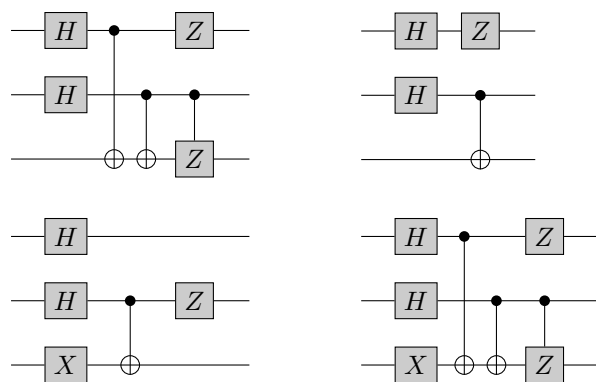
In the case that  $U = \Lambda(P)$ , one has that the projection (3.6) is the sum of the four Clifford–Hamiltonian projections corresponding to these vectors:

$$(3.7) \quad \begin{aligned} |-\rangle|00\rangle &= (ZH \otimes \mathbb{1} \otimes \mathbb{1})|000\rangle, \\ |-\rangle|01\rangle &= (ZH \otimes \mathbb{1} \otimes X)|000\rangle, \\ |-\rangle|10\rangle &= (ZH \otimes X \otimes \mathbb{1})|000\rangle, \\ |\odot\rangle|11\rangle &= (P^*H \otimes X \otimes X)|000\rangle, \end{aligned}$$

where  $|\odot\rangle = (|0\rangle - i|1\rangle)/\sqrt{2}$ . In the case that  $U = H \otimes H$ , one has that the projection (3.6) is the sum of the four Clifford–Hamiltonian projections corresponding to these vectors:

$$(3.8) \quad \begin{aligned} |\psi_1\rangle &= (|000\rangle - |011\rangle - |101\rangle - |110\rangle)/2, \\ |\psi_2\rangle &= (|000\rangle + |011\rangle - |100\rangle - |111\rangle)/2, \\ |\psi_3\rangle &= (|001\rangle - |010\rangle + |101\rangle - |110\rangle)/2, \\ |\psi_4\rangle &= (|001\rangle + |010\rangle - |100\rangle + |111\rangle)/2. \end{aligned}$$

All four of these vectors are obtained by a Clifford operation applied to the all-zero state. In particular, when the following Clifford circuits are applied to the state  $|000\rangle$ , the states  $|\psi_1\rangle$ ,  $|\psi_2\rangle$ ,  $|\psi_3\rangle$ , and  $|\psi_4\rangle$  are obtained:



This completes the proof.  $\square$

*Remark 3.2.* If one is given a witness to a given QMA problem  $A$ , it is possible to efficiently compute a witness to the corresponding  $k$ -local Hamiltonian problem instance through Kitaev's reduction by preparing a superposition of clock states and then running a verification circuit for the corresponding number of steps. Our reduction also inherits this property.

*Remark 3.3.* There is no loss of generality in setting  $q = 1$  in the statement of the  $k$ -LCH problem, meaning that Theorem 3.1 holds for this somewhat simplified problem statement. This may be proved by repeating each Hamiltonian term  $q$  times in a given problem instance and adjusting  $p$  as necessary.

*Remark 3.4.* States of the form  $C|0^k\rangle$  for a Clifford operation  $C$ , are stabilizer states of  $k$  qubits. Theorem 3.1 therefore implies that there exists a QMA verification procedure in which the verifier randomly chooses a  $k$ -qubit stabilizer state and checks whether the quantum witness state is orthogonal to it.

*Remark 3.5.* If one takes  $U = H$  in (3.6), the resulting projection operator projects onto the two-dimensional subspace spanned by  $|-\rangle|\gamma_0\rangle$  and  $|+\rangle|\gamma_1\rangle$ , where

$$(3.9) \quad |\gamma_0\rangle = \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle \quad \text{and} \quad |\gamma_1\rangle = \sin(\pi/8)|0\rangle - \cos(\pi/8)|1\rangle$$

are eigenvectors of  $H$ . This projection cannot be expressed as a sum of Clifford–Hamiltonian projections, which explains why we needed to replace  $H$  with  $H \otimes H$  in the proof above.

While considering this projection is not useful for proving Theorem 3.1, we do obtain from it a different result. In particular, we obtain an alternative proof of a result due to Morimae, Nagaj, and Schuch [45] establishing that single-qubit measurements and classical postprocessing are sufficient for QMA verification. Reference [45] actually provides two proofs of this fact, one based on measurement-based quantum computation and the other based on a local-Hamiltonian problem type of approach similar to what we propose. While their local-Hamiltonian approach does not work for one-sided error (or QMA<sub>1</sub>) verifications, ours does (as does their measurement-based quantum computation proof).

**4. Description of the proof system.** We now describe our zero-knowledge protocol for the local Clifford–Hamiltonian problem. The main steps of the proof system are described in the subsections that follow, and the entire proof system is summarized in Figure 4.1. Properties of the proof system, including completeness, soundness, and the zero-knowledge property, are discussed in later sections of the paper.

As was previously suggested, our proof system assumes the existence of a quantum computationally secure commitment scheme. Throughout this section it is to be assumed that an instance of the  $k$ -LCH problem has been selected. The instance describes Clifford–Hamiltonian projections  $H_1, \dots, H_m$ , each given by  $H_j = C_j^*|0^k\rangle\langle 0^k|C_j$  for  $k$ -qubit Clifford operations  $C_1, \dots, C_m$ , along with a specification of which of the  $n$  qubits these projections act upon. The proof system does not refer to the parameters  $p$  and  $q$  in the description of the  $k$ -LCH problem, as these parameters are only relevant to the performance of the proof system and not its implementation. It must be assumed, however, that the completeness parameter  $2^{-p}$  is a negligible function of the entire problem instance size in order for the proof system to be zero-knowledge, and we will make this assumption hereafter.

**4.1. Prover’s witness encoding.** Let  $\mathbf{X} = (X_1, \dots, X_n)$  be an  $n$ -tuple of single-qubit registers. These qubits are assumed to initially be in the prover’s possession, and they store an  $n$ -qubit quantum state  $\rho$  representing a possible witness for the instance of the  $k$ -LCH problem under consideration.

The first step of the proof system requires the prover to encode the state of  $\mathbf{X}$ , using a scheme that consists of four steps. Throughout the description of these steps it is to be assumed that  $N$  is a polynomially bounded function of the input size and is

*Prover's encoding step:*

The prover selects a tuple  $(t, \pi, a, b)$  uniformly at random, where  $t = t_1 \cdots t_n$  for  $t_1, \dots, t_n \in \{0, +, \odot\}^N$ ,  $\pi \in S_{2N}$ , and  $a = a_1 \cdots a_n$  and  $b = b_1 \cdots b_n$  for  $a_1, \dots, a_n, b_1, \dots, b_n \in \Sigma^{2N}$ . The witness state contained in qubits  $(X_1, \dots, X_n)$  is encoded into qubit tuples

$$(4.1) \quad (Y_1^1, \dots, Y_{2N}^1), \dots, (Y_1^n, \dots, Y_{2N}^n)$$

as described in the main text. These qubits are sent to the verifier, along with a commitment to the tuple  $(\pi, a, b)$ .

*Coin-flipping protocol:*

The prover and verifier engage in a coin-flipping protocol, choosing a string  $r$  of a fixed length uniformly at random. This random string  $r$  determines a Hamiltonian term  $H_r = C_r^* |0^k\rangle \langle 0^k| C_r$  that is to be tested.

*Verifier's measurement:*

The verifier applies the Clifford operation  $C_r$  transversally to the qubits

$$(4.2) \quad (Y_1^{i_1}, \dots, Y_{2N}^{i_1}), \dots, (Y_1^{i_k}, \dots, Y_{2N}^{i_k}),$$

and measures all of these qubits in the standard basis for  $(i_1, \dots, i_k)$  being the indices of the qubits upon which the Hamiltonian term  $H_r$  acts nontrivially. The result of this measurement is sent to the prover.

*Prover's verification and response:*

The prover checks that the verifier's measurement results are consistent with the states of the trap qubits and the concatenated Steane code, aborting the proof system if not (causing the verifier to reject). In case the measurement results are consistent, the prover demonstrates that these measurement results are consistent with its prior commitment to  $(\pi, a, b)$  and with the Hamiltonian term  $H_r$ , through a classical zero-knowledge proof system for the corresponding NP statement described in the main text. The verifier accepts or rejects accordingly.

FIG. 4.1. Summary of the zero-knowledge proof system for the LCH problem.

an even positive integer power of 7. In effect,  $N$  acts as a security parameter (for the zero-knowledge property of the proof system), and we take it to be an even power of 7 so that it may be viewed as a number of qubits that could arise from a concatenated Steane code allowing for a transversal application of Clifford operations, as described in section 2.6. In particular, through an appropriate choice of  $N$ , one may guarantee that this code has any desired polynomial lower bound for the minimum nonzero Hamming weight of its underlying classical code.

1. For each  $i = 1, \dots, n$ , the qubit  $X_i$  is encoded into qubits  $(Y_1^i, \dots, Y_N^i)$  by means of the concatenated Steane code. This results in the  $N$ -tuples

$$(4.3) \quad (Y_1^1, \dots, Y_N^1), \dots, (Y_1^n, \dots, Y_N^n).$$

2. To each of the  $N$ -tuples in (4.3), the prover concatenates an additional  $N$  trap qubits with each trap qubit being initialized to one of the single qubit pure states  $|0\rangle$ ,  $|+\rangle$ , or  $|\odot\rangle$ , selected independently and uniformly at random.

This results in qubits

$$(4.4) \quad (\mathbf{Y}_1^1, \dots, \mathbf{Y}_{2N}^1), \dots, (\mathbf{Y}_1^n, \dots, \mathbf{Y}_{2N}^n).$$

The prover stores the string  $t = t_1 \cdots t_n$ , for  $t_1, \dots, t_n \in \{0, +, \odot\}^N$  representing the randomly chosen states of the trap qubits.

3. A random permutation  $\pi \in S_{2N}$  is selected, and the qubits in each of the  $2N$ -tuples (4.4) are permuted according to  $\pi$ . (Note that it is a single permutation  $\pi$  that is selected and applied to all of the  $2N$ -tuples simultaneously.)
4. The quantum one-time pad is applied independently to each qubit in (4.4) (after they are permuted in step 3). That is, for  $a_i, b_i \in \Sigma^{2N}$  chosen independently and uniformly at random, the unitary transformation  $X^{a_i}Z^{b_i}$  is applied to  $(\mathbf{Y}_1^i, \dots, \mathbf{Y}_{2N}^i)$ , and the strings  $a_i$  and  $b_i$  are stored by the prover for each  $i = 1, \dots, n$ .

The randomness required by these encoding steps is described by a tuple  $(t, \pi, a, b)$ , where  $t$  is the string representing the states of the trap qubits described in step 2,  $\pi \in S_{2N}$  is the permutation applied in step 3, and  $a = a_1 \cdots a_n$  and  $b = b_1 \cdots b_n$  are binary strings representing the Pauli operators applied in the one-time pad in step 4. After performing the above encoding steps, the prover sends the resulting qubits,

$$(4.5) \quad \mathbf{Y} = ((\mathbf{Y}_1^1, \dots, \mathbf{Y}_{2N}^1), \dots, (\mathbf{Y}_1^n, \dots, \mathbf{Y}_{2N}^n)),$$

along with a commitment

$$(4.6) \quad z = \text{commit}((\pi, a, b), s)$$

to the tuple  $(\pi, a, b)$ , to the verifier. Here we assume that  $s$  is a random string chosen by the prover that allows for this commitment. (It is not necessary for the prover to commit to the selection of the trap qubit states indicated by  $t$ , although it would not affect the properties of the proof system if it were modified so that the prover also committed to the trap qubit state selections.)

**4.2. Verifier's random challenge.** Upon receiving the prover's encoded witness and commitment, the verifier issues a challenge: for a randomly selected index  $j \in \{1, \dots, m\}$ , the verifier will check that the  $j$ th Hamiltonian term

$$(4.7) \quad H_j = C_j^* |0^k\rangle \langle 0^k| C_j$$

is not violated. Generally speaking, the verifier's actions in issuing this challenge are as follows: for a certain collection of qubits, the verifier applies the Clifford operation  $C_j$  transversally to those qubits, performs a measurement with respect to the standard basis, sends the outcomes to the prover, and then expects the prover to demonstrate that the obtained outcomes are valid (in the sense to be described later).

The randomly selected Hamiltonian term is to be determined by a binary string  $r$ , of a fixed length  $\lceil \log m \rceil$ , that should be viewed as being chosen uniformly at random. (In a moment we will discuss the random choice of  $r$ , which will be given by the output of a coin-flipping protocol that happens to be uniform for honest participants.) It is not important exactly how the binary strings of length  $\lceil \log m \rceil$  are mapped to the indices  $\{1, \dots, m\}$ , so long as every index is represented by at least one string—so that for a uniformly chosen string  $r$ , each Hamiltonian term  $j$  is selected with a nonnegligible probability. We will write  $H_r$  and  $C_r$  in place of  $H_j$  and  $C_j$ , and refer to the Hamiltonian term determined by  $r$ , when it is convenient to do this.



It would be natural to allow the verifier to randomly determine which Hamiltonian term is to be tested—but, as suggested above, we will assume that the challenge is determined through a *coin-flipping protocol* rather than leaving the choice to the verifier. More specifically, throughout the present subsection, it should be assumed that the random choice of the string  $r$  that determines which challenge is issued is the result of independent iterations of a commitment-based coin-flipping protocol (i.e., the honest prover commits to a random  $y_i \in \Sigma$ , the honest verifier selects  $z_i \in \Sigma$  at random, the prover reveals  $y_i$ , and the two participants agree that the  $i$ th random bit of  $r$  is  $r_i = y_i \oplus z_i$ ). This guarantees (assuming the security of the commitment protocol) that the choices are truly random, and greatly simplifies the analysis of the zero-knowledge property of the proof system. The use of such a protocol might not actually be necessary for the security of the proof system, but we leave the investigation of whether it is necessary to future work.

Now, let  $(i_1, \dots, i_k)$  denote the indices of the qubits upon which the Hamiltonian term determined by the random string  $r$  acts nontrivially. The verifier applies the Clifford operation  $C_r$  independently to each of the  $k$ -qubit tuples

$$(4.8) \quad (Y_1^{i_1}, \dots, Y_1^{i_k}), \dots, (Y_{2N}^{i_1}, \dots, Y_{2N}^{i_k}),$$

which is equivalent to saying that  $C_r$  is applied transversally to the tuples

$$(4.9) \quad (Y_1^{i_1}, \dots, Y_{2N}^{i_1}), \dots, (Y_1^{i_k}, \dots, Y_{2N}^{i_k})$$

that encode the qubits on which the Hamiltonian term  $H_r$  acts nontrivially. The qubits (4.9) are then measured with respect to the standard basis, and the results are sent to the prover. We will let  $u_{i_1}, \dots, u_{i_k} \in \Sigma^{2N}$  denote the binary strings representing the verifier's standard basis measurement outcomes (or claimed outcomes) corresponding to the measurements of the tuples (4.9).

**4.3. Prover's check and response.** Upon receiving the verifier's claimed measurement outcomes corresponding to the randomly selected Hamiltonian term, the prover first checks to see that these outcomes could indeed have come from the measurements specified above, and then tries to convince the verifier that these measurement outcomes are consistent with the selected term.

In more detail, suppose that the Hamiltonian term determined by  $r$  has been challenged. As above, we assume that this term acts nontrivially on the  $k$  qubits indexed by the  $k$ -tuple  $(i_1, \dots, i_k)$ , and we will write  $u = u_{i_1} \cdots u_{i_k} \in \Sigma^{2kN}$  to denote the verifier's claimed standard basis measurement outcomes.

To define the prover's check for this string, it will be helpful to first define a predicate  $R_r$ , which is a function of  $t$ ,  $\pi$ , and  $u$ , and essentially represents the prover's check *after* it has made an adjustment to the verifier's response to account for the one-time pad. For each  $i \in \{i_1, \dots, i_k\}$ , define strings  $y_i, z_i \in \Sigma^N$  so that

$$(4.10) \quad \pi(y_i z_i) = u_i.$$

The predicate  $R_r$  takes the value 1 if and only if these two conditions are met:

1.  $y_i \in \mathcal{D}_N$  for every  $i \in \{i_1, \dots, i_k\}$ , and  $y_i \in \mathcal{D}_N^1$  for at least one index  $i \in \{i_1, \dots, i_k\}$ .
2.  $\langle z_{i_1} \cdots z_{i_k} | C_r^{\otimes N} | t_{i_1} \cdots t_{i_k} \rangle \neq 0$ .

(Here we have written  $|t_{i_1} \cdots t_{i_k}\rangle$  to denote the pure state of  $kN$  qubits obtained by tensoring the states  $|0\rangle$ ,  $|+\rangle$ , and  $|\odot\rangle$  in the most natural way.) The first condition concerns measurement outcomes corresponding to nontrap qubits, and reflects the

condition that these measurement outcomes are proper encodings of binary values—but not all of which encode 0. The second condition concerns the consistency of the verifier's measurements with the trap qubits.

Next, we will define a predicate  $Q_r$ , which is a function of the variables  $t, \pi, a, b$ , and  $u$ , where  $t, \pi$ , and  $u$  are as above and  $a, b \in \Sigma^{2nN}$  refer to the strings used for the one-time pad. The predicate  $Q_r$  represents the prover's actual check, in the case that the Hamiltonian term determined by  $r$  has been selected, including an adjustment to account for the one-time pad. Let  $c_1, \dots, c_n, d_1, \dots, d_n \in \Sigma^{2N}$  be the unique strings for which the equation

$$(4.11) \quad C_r^{\otimes 2N} (X^{a_1} Z^{b_1} \otimes \dots \otimes X^{a_n} Z^{b_n}) = \alpha (X^{c_1} Z^{d_1} \otimes \dots \otimes X^{c_n} Z^{d_n}) C_r^{\otimes 2N}$$

holds for some choice of  $\alpha \in \{1, i, -1, -i\}$ . The Clifford operation  $C_r$  acts trivially on those qubits indexed by strings outside of the set  $\{i_1, \dots, i_k\}$ , so it must be the case that  $c_i = a_i$  and  $d_i = b_i$  for  $i \notin \{i_1, \dots, i_k\}$ , but for those indices  $i \in \{i_1, \dots, i_k\}$  it may be the case that  $c_i \neq a_i$  and  $d_i \neq b_i$ . We will also write  $c = c_1 \dots c_n$  and  $d = d_1 \dots d_n$  for the sake of convenience. Given a description of the Clifford operation  $C_r$  it is possible to efficiently compute  $c$  and  $d$  from  $a$  and  $b$ . Having defined  $c$  and  $d$ , we may now express the predicate  $Q_r$  as follows:

$$(4.12) \quad Q_r(t, \pi, u, a, b) = R_r(t, \pi, u \oplus c_{i_1} \dots c_{i_k}).$$

In essence, the predicate  $Q_r$  checks the validity of the verifier's claimed measurement results by first adjusting for the one-time pad, then referring to  $R_r$ .

The prover evaluates the predicate  $Q_r$ , and aborts the proof system if the predicate evaluates to 0 (as this is indicative of a dishonest verifier). Otherwise, the prover aims to convince the verifier that the measurement outcomes  $u$  are consistent with the prover's encoding, and also that they are not in violation of the Hamiltonian term  $H_r$ . It does this specifically by engaging in a classical zero-knowledge proof system for the following NP statement: there exists a random string  $s$  and an encoding key  $(t, \pi, a, b)$  such that (i)  $\text{commit}((\pi, a, b), s)$  matches the prover's initial commitment  $z$ , and (ii)  $Q_r(t, \pi, u, a, b) = 1$ .

It will be convenient later, in the analysis of the proof system, to sometimes view  $r$  as being an input to the predicates defined above. Specifically, we define predicates

$$(4.13) \quad Q(r, t, \pi, a, b, u) = Q_r(t, \pi, a, b, u) \quad \text{and} \quad R(r, t, \pi, u) = R_r(t, \pi, u)$$

for this purpose. We also note explicitly that these predicates are polynomial-time computable.

**5. Completeness and soundness of the proof system.** It is evident that the proof system described in the previous section is complete. For a given instance of the local Clifford–Hamiltonian problem, if the prover and verifier both behave honestly, as suggested in the description of the proof system, the verifier will accept with precisely the same probability that would be obtained by randomly selecting a Hamiltonian term, measuring the original  $n$ -qubit witness state against the corresponding projection, and accepting or rejecting accordingly. For a positive problem instance, this acceptance probability is at least  $1 - 2^{-P}$  (for every choice of a random string  $r$ ).

Next we will consider the soundness of the proof system. We will prove that on a negative instance of the problem, the honest verifier must reject with nonnegligible probability. The prover initially sends to the verifier the qubits

$$(5.1) \quad (Y_1^1, \dots, Y_{2N}^1), \dots, (Y_1^n, \dots, Y_{2N}^n),$$

along with a commitment  $z = \text{commit}((\pi, a, b), s)$  to a tuple  $(\pi, a, b)$ . We have assumed that the commitment is perfectly binding, so there is a well-defined tuple  $(\pi, a, b)$  that is determined by the prover's commitment  $z$ . We may assume without loss of generality that this tuple has the proper form (meaning that  $\pi \in S_{2N}$  is a permutation and  $a$  and  $b$  are binary strings of length  $2nN$ , as specified in the description of the proof system), as a commitment to a string not of this form must lead to rejection with high probability in all cases. Let  $\xi$  be the state of the qubits

$$(5.2) \quad (\Upsilon_1^1, \dots, \Upsilon_N^1), \dots, (\Upsilon_1^n, \dots, \Upsilon_N^n)$$

that is obtained by inverting the quantum one-time pad with respect to the strings  $a$  and  $b$ , inverting the permutation of each of the tuples (5.1) with respect to the permutation  $\pi$ , and discarding the last  $N$  qubits within each tuple (i.e., the trap qubits). For an honest prover, the state  $\xi$  would be the state obtained by encoding the original witness state using the concatenated Steane code—although in general it cannot be assumed that  $\xi$  arises in this way. Although the verifier is not capable of recovering the state  $\xi$  on its own, because it does not know  $(\pi, a, b)$ , it will nevertheless be helpful to refer to the state  $\xi$  for the purposes of establishing the soundness condition of the proof system.

We will define a collection of  $N$ -qubit projections operators and a channel from  $N$  qubits to one that will be useful for establishing soundness. First, let

$$(5.3) \quad \Pi_0 = \sum_{x \in \mathcal{D}_N^0} |x\rangle\langle x| \quad \text{and} \quad \Pi_1 = \sum_{x \in \mathcal{D}_N^1} |x\rangle\langle x|,$$

where  $\mathcal{D}_N^0$  and  $\mathcal{D}_N^1$  are subsets of  $\Sigma^N$  representing classical code words of the concatenated Steane code. A standard basis measurement of any qubit encoded using this code will necessarily yield an outcome in one of these two sets: an encoded  $|0\rangle$  state yields an outcome in  $\mathcal{D}_N^0$ , and an encoded  $|1\rangle$  state yields an outcome in  $\mathcal{D}_N^1$ . The projections  $\Pi_0$  and  $\Pi_1$  therefore correspond to these two possibilities, while the projection operator  $\mathbb{1} - (\Pi_0 + \Pi_1)$  corresponds to the situation in which a standard basis measurement has yielded a result outside of the classical code space  $\mathcal{D}_N = \mathcal{D}_N^0 \cup \mathcal{D}_N^1$ . Also define projections

$$(5.4) \quad \Delta_0 = \frac{\mathbb{1}^{\otimes N} + Z^{\otimes N}}{2} \quad \text{and} \quad \Delta_1 = \frac{\mathbb{1}^{\otimes N} - Z^{\otimes N}}{2},$$

which are the projections onto the spaces spanned by all even- and odd-parity standard basis states, respectively. It holds that  $\Pi_0 \leq \Delta_0$  and  $\Pi_1 \leq \Delta_1$ , as the codewords in  $\mathcal{D}_N^0$  all have even parity and the codewords in  $\mathcal{D}_N^1$  all have odd parity. Finally, define a channel  $\Xi_N$ , mapping  $N$  qubits to 1 qubit, as follows:

$$(5.5) \quad \Xi_N(\sigma) = \frac{\langle \mathbb{1}^{\otimes N}, \sigma \rangle \mathbb{1} + \langle X^{\otimes N}, \sigma \rangle X + \langle Y^{\otimes N}, \sigma \rangle Y + \langle Z^{\otimes N}, \sigma \rangle Z}{2}$$

for every  $N$ -qubit operator  $\sigma$ . It is evident that this mapping preserves trace, and is completely positive when  $N \equiv 1 \pmod{4}$ , which holds because  $N$  is an even power of 7. The complete positivity of  $\Xi_N$  when  $N \equiv 1 \pmod{4}$  may be verified by establishing that its Choi operator is positive semidefinite, which is a routine verification:

$$(5.6) \quad \begin{aligned} J(\Xi_N) &= \frac{1}{2} (\mathbb{1}^{\otimes(N+1)} + X^{\otimes(N+1)} - Y^{\otimes(N+1)} + Z^{\otimes(N+1)}) \\ &= \frac{1}{8} (\mathbb{1}^{\otimes(N+1)} + X^{\otimes(N+1)} - Y^{\otimes(N+1)} + Z^{\otimes(N+1)})^2. \end{aligned}$$

One may observe that the adjoint mapping to  $\Xi_N$  is given by

$$(5.7) \quad \Xi_N^*(\tau) = \frac{\langle \mathbb{1}, \tau \rangle \mathbb{1}^{\otimes N} + \langle X, \tau \rangle X^{\otimes N} + \langle Y, \tau \rangle Y^{\otimes N} + \langle Z, \tau \rangle Z^{\otimes N}}{2},$$

and satisfies

$$(5.8) \quad \Xi_N^*(|0\rangle\langle 0|) = \Delta_0 \quad \text{and} \quad \Xi_N^*(|1\rangle\langle 1|) = \Delta_1.$$

Now, consider the state  $\rho = \Xi_N^{\otimes n}(\xi)$  of the qubits  $(X_1, \dots, X_n)$  that is obtained from  $\xi$  when  $\Xi_N$  is applied independently to each of the  $N$ -tuples of qubits in (5.2). We will prove that the verifier must reject with nonnegligible probability for a given choice of  $r$  provided that  $\rho$  violates the corresponding Hamiltonian term  $H_r$ . Because every  $n$ -qubit state creates a nonnegligible violation in at least one Hamiltonian term for a negative problem instance, this will suffice to prove the soundness of the proof system.

For each random string  $r$  generated by the coin-flipping procedure, one may define a measurement on the state  $\xi$  that corresponds to the verifier's actions and final decision to accept or reject given this choice of  $r$ , assuming the prover behaves optimally after the coin flipping and the verifier's measurement take place. Specifically, corresponding to the Hamiltonian term  $H_r = C_r^*|0^k\rangle\langle 0^k|C_r$ , acceptance is represented by a projection operator  $\Lambda_r$  on the qubits

$$(5.9) \quad (Y_1^{i_1}, \dots, Y_N^{i_1}), \dots, (Y_1^{i_k}, \dots, Y_N^{i_k})$$

defined as follows:

$$(5.10) \quad \Lambda_r = \sum_{\substack{z \in \Sigma^k \\ z \neq 0^k}} (C_r^{\otimes N})^* (\Pi_{z_1} \otimes \dots \otimes \Pi_{z_k}) (C_r^{\otimes N}).$$

The probability that the verifier rejects, for a given choice of  $r$ , is therefore at least  $1 - \langle \Lambda_r, \xi \rangle$ . Because  $\Pi_0 \leq \Delta_0$  and  $\Pi_1 \leq \Delta_1$ , the probability of rejection is therefore at least

$$(5.11) \quad \begin{aligned} 1 - \sum_{\substack{z \in \Sigma^k \\ z \neq 0^k}} \langle (C_r^{\otimes N})^* (\Delta_{z_1} \otimes \dots \otimes \Delta_{z_k}) (C_r^{\otimes N}), \xi \rangle \\ = \langle (C_r^{\otimes N})^* (\Delta_0 \otimes \dots \otimes \Delta_0) (C_r^{\otimes N}), \xi \rangle. \end{aligned}$$

By considering properties of the channel  $\Xi_N$ , we conclude that the verifier rejects with probability at least

$$(5.12) \quad \begin{aligned} & \langle (C_r^{\otimes N})^* (\Xi_N^*(|0\rangle\langle 0|) \otimes \dots \otimes \Xi_N^*(|0\rangle\langle 0|)) C_r^{\otimes N}, \xi \rangle \\ & = \langle 0^k | \Xi_N^{\otimes k} (C_r^{\otimes N} \xi (C_r^{\otimes N})^*) | 0^k \rangle = \langle C_r^* | 0^k \rangle \langle 0^k | C_r, \Xi_N^{\otimes k}(\xi) \rangle = \langle H_r, \rho \rangle. \end{aligned}$$

Here we have used the observation that

$$(5.13) \quad \Xi_N^{\otimes k} (C^{\otimes N} \sigma (C^{\otimes N})^*) = C \Xi_N^{\otimes k}(\sigma) C^*$$

for every  $k$ -qubit Clifford operation  $C$  and every  $kN$ -qubit state  $\sigma$ , which may be verified by considering the action of  $\Xi_N$  on Hadamard, phase, and controlled-not gates.

Intuitively speaking, the argument above shows that whatever state a malicious prover sends in the first message, one can essentially decode that state with respect to a highly simplified variant of the encoding scheme (after peeling off the quantum one-time pad and discarding the trap qubits), recovering a state that would pass the Hamiltonian energy test with at least the same probability as the verifier's acceptance probability in our zero-knowledge proof system. Because this probability must be bounded away from 1 on average for any no-instance of the problem, we obtain a soundness guarantee for the proof system.

**6. Zero-knowledge property of the proof system.** We now prove that the proof system described in section 4 is zero-knowledge in the quantum computational sense, assuming that the commitment scheme used in the proof system is unconditionally binding and quantum computationally concealing. The proof has several steps, to be presented below, but first we will summarize the main technical goal of the proof.

Figure 6.1 shows a diagram of the interaction between the honest participants in the proof system. A cheating verifier aiming to extract knowledge from the prover might, of course, not follow the prescribed actions of the honest verifier. In particular, the cheating verifier may take a quantum register as input, store quantum information in-between its actions, and output a quantum register. Figure 6.2 illustrates such a cheating verifier interacting with the honest prover. The goal of the proof is to demonstrate that, for any cheating verifier of the form suggested by Figure 6.2, there exists an efficient simulator that implements a channel that is computationally indistinguishable from the channel implemented by the cheating verifier and prover interaction. In particular, the simulator does not have access to the witness state  $\rho$ . This will be done, through a hybrid-style argument, over the course of several steps.

**Step 1: Simulating the coin-flipping protocol.** By the results of [15], there must exist an efficient simulator  $S_1$  for the interaction of  $V'_1$  with  $P_1$ . To be more precise, for  $S_1$  being given an input of the same form as  $V'_1$ , along with a uniformly chosen random string  $r$  of the length required by our proof system, the resulting action is quantum computationally indistinguishable from  $V'_1$  interacting with  $P_1$ .

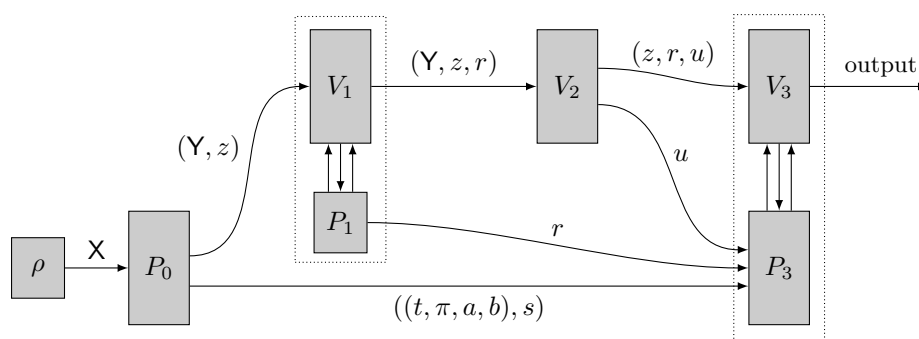


FIG. 6.1. The interaction between honest participants. The prover's quantum witness  $\rho$  is encoded into  $Y$  together with the encoding key  $(t, \pi, a, b)$  by the prover's action  $P_0$ . The string  $z$  represents the prover's commitment to  $(\pi, a, b)$  and the string  $s$  represents random bits used by the prover to implement this commitment. The string  $r$  represents the random bits generated by the coin-flipping protocol, which is depicted within the dotted rectangle on the left. The string  $u$  represents the verifier's standard basis measurements for a subset of the qubits of  $Y$  determined by the challenge corresponding to the random string  $r$ . The classical zero-knowledge protocol is depicted within the dotted rectangle on the right.

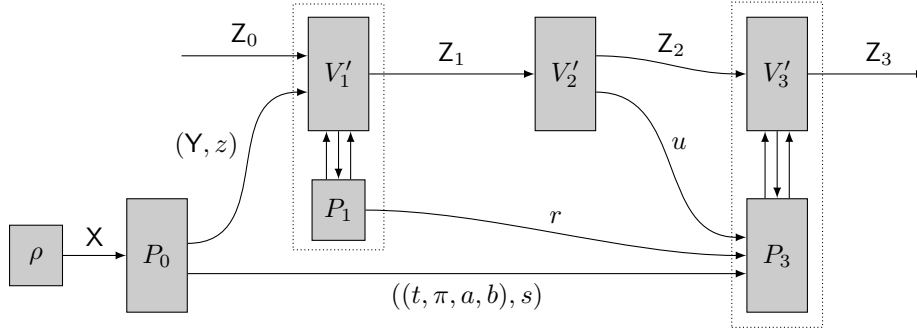


FIG. 6.2. A potentially dishonest verifier takes an auxiliary quantum register  $Z_0$  as input, may store quantum information (represented by registers  $Z_1$  and  $Z_2$ ), and outputs quantum information stored in register  $Z_3$ .

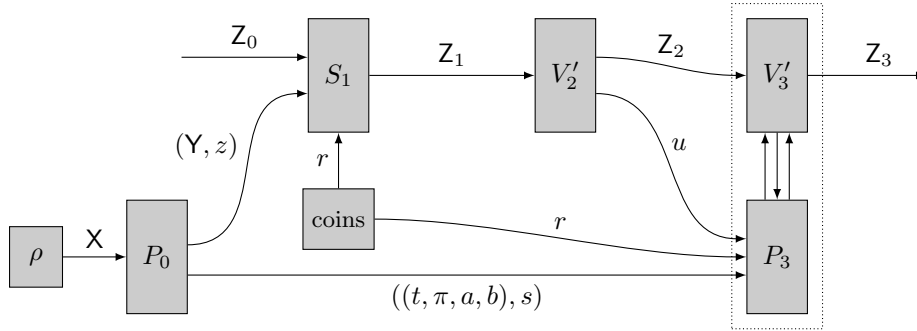


FIG. 6.3. The interaction corresponding to the execution of the coin-flipping protocol has been replaced by a simulator  $S_1$  along with a true random string generator (labeled coins).

Figure 6.3 illustrates the process that is obtained by performing this substitution. As the simulator  $S_1$  together with the true random string generator is computationally indistinguishable from the interaction between  $V'_1$  and  $P_1$ , the process illustrated in Figure 6.3 is computationally indistinguishable from the process illustrated in Figure 6.2. It therefore suffices for us to prove that the process illustrated in Figure 6.3 can be efficiently simulated (without access to the witness state  $\rho$ ).

**Step 2: Simulating the classical zero-knowledge protocol.** In the next step of the proof, we replace the interaction between a cheating verifier  $V'_3$  and the prover  $P_3$  in the classical zero-knowledge protocol by an efficient simulator  $S_3$  together with the predicate  $Q$ , as is illustrated in Figure 6.4.

To describe this step in greater detail, we first observe that the prover holds an encoding key  $(t, \pi, a, b)$  along with the random string  $s$  it has used to commit to the tuple  $(\pi, a, b)$ . The commitment  $z = \text{commit}((\pi, a, b), s)$  is sent to the verifier, together with the encoding register  $Y$ , in the first step of the proof system. The verifier then sends a string  $u$  that, in the honest case, represents the output of a measurement of some subset of the qubits of  $Y$  with respect to the standard basis, after the transversal application of a Clifford operation depending on the random choice of  $r$ . The statement that the honest prover aims to prove in the classical zero-knowledge protocol is that there exists an encoding key  $(t, \pi, a, b)$  along with a string  $s$  such that  $z = \text{commit}((\pi, a, b), s)$  and  $Q(r, t, \pi, a, b, u) = 1$ .

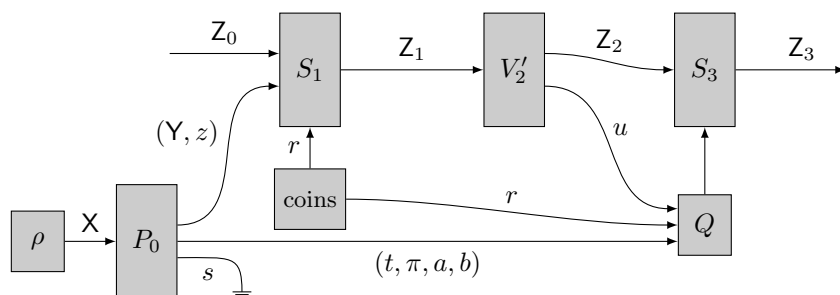


FIG. 6.4. The interaction corresponding to the execution of the classical zero-knowledge protocol has been replaced by a simulator  $S_3$  along with the predicate  $Q$ . It is assumed that when the output of  $Q$  is 0, the simulator  $S_3$  behaves as the cheating verifier  $V_3'$  would when the prover aborts the proof system. The string  $s$  produced by  $P_0$  in forming the commitment to  $(\pi, a, b)$  is discarded.

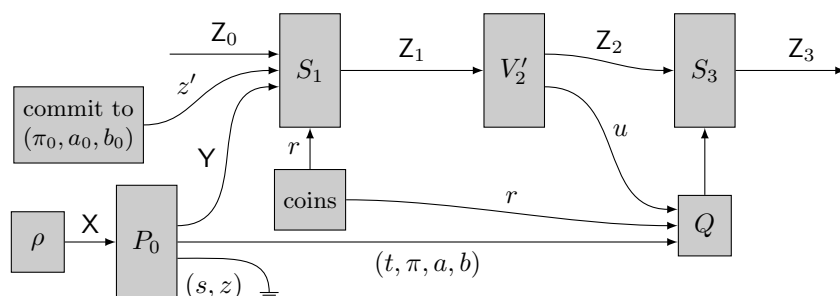


FIG. 6.5. The commitment  $z$  given as input to  $S_1$  has been replaced by a dummy commitment  $z'$  to a fixed tuple  $(\pi_0, a_0, b_0)$ . Having been replaced by  $z'$ , the original commitment  $z$  computed by  $P_0$  may be considered to be discarded along with the random string  $s$  used to form that commitment.

The honest prover always holds an encoding key  $(t, \pi, a, b)$  and a binary string  $s$  for which  $z = \text{commit}((\pi, a, b), s)$ , so we need not concern ourselves with the case that this is not so. The case that  $Q(r, t, \pi, a, b, u) = 1$  therefore corresponds to a yes-instance of the classical zero-knowledge protocol, and by the assumption that the classical zero-knowledge protocol is indeed computational zero-knowledge against quantum attacks (q.v. Definition 2.6), there must therefore exist an efficient simulator  $S_3$  that computes a transformation from  $Z_2$  to  $Z_3$  that is computationally indistinguishable from the one induced by the interaction between  $V_3'$  and  $P_3$  in this case (which is signaled to  $S_3$  when it receives a 1 input from the predicate  $Q$ ). We have assumed that the prover aborts in the case  $Q(r, t, \pi, a, b, u) = 0$ , and so we define  $S_3$  so that when it receives a 0 input from the predicate  $Q$ , it directly mimics whatever  $V_3'$  does in the situation that the prover aborts. It follows that the process described in Figure 6.4 is computationally indistinguishable from the one described by Figure 6.3. We observe that the string  $s$  used by  $P_0$  to form the commitment  $z = \text{commit}((\pi, a, b), s)$  can safely be discarded immediately after  $P_0$  is run, as it is never again used in Figure 6.4.

**Step 3: Eliminating the commitment.** The next step is to eliminate the commitment. To this end, we consider the process described in Figure 6.5, which is identical to Figure 6.4 except that the commitment  $z$  to the tuple  $(\pi, a, b)$  given as input to  $S_1$  has been replaced by a dummy commitment  $z'$  to a fixed tuple  $(\pi_0, a_0, b_0)$ . Specifically, we take  $\pi_0$  to be the identity permutation and  $a_0$  and  $b_0$  to be all-zero



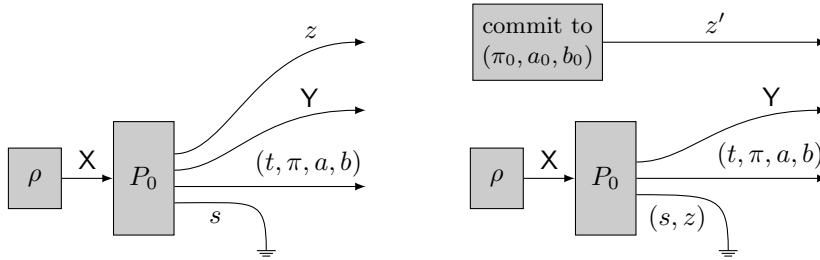


FIG. 6.6. The processes described in Figures 6.4 and 6.5 differ only in the initial portions depicted. As the subsequent steps are quantum polynomial-time computable and identical for two processes, we find that the processes described in Figures 6.4 and 6.5 are quantum computationally indistinguishable provided that the states generated by the processes depicted are quantum computationally indistinguishable.

strings of length  $2nN$ . For the sake of clarity, we note explicitly that we do not replace  $(\pi, a, b)$  with  $(\pi_0, a_0, b_0)$  as an input to the predicate  $Q$ , it is only the commitment to  $S_1$  that is changed from Figure 6.4 to Figure 6.5. We claim that the processes described in Figures 6.4 and 6.5 are quantum computationally indistinguishable.

To verify this claim, observe first that  $S_1$ ,  $V'_2$ ,  $S_3$ ,  $Q$ , and the generation of the random coin flips  $r$  are all polynomial-time computable quantum processes. Therefore, if the processes described in Figures 6.4 and 6.5 were computationally distinguishable, the simpler processes described in Figure 6.6, which simply generate states, would also necessarily be computationally distinguishable. This is because the processes described in Figures 6.4 and 6.5 are obtained by composing the processes depicted in Figure 6.6 with exactly the same polynomial-time computable quantum process obtained from  $S_1$ ,  $V'_2$ ,  $S_3$ ,  $Q$ , and the generation of the random coin flips  $r$ .

To justify the claim made above, it therefore suffices to prove that the processes shown in Figure 6.6 are quantum computationally indistinguishable. Observe that the states generated by these two processes can be expressed as

$$(6.1) \quad \sum_z p(z) |z\rangle\langle z| \otimes \tau_z \quad \text{and} \quad \sum_z p(z) |z'\rangle\langle z'| \otimes \tau_z$$

for some choice of a distribution  $p$  and a collection of states  $\{\tau_z\}$  representing both  $Y$  and  $(t, \pi, a, b)$ . If these two states were quantum computationally distinguishable, then by convexity the states

$$(6.2) \quad |z\rangle\langle z| \otimes \tau_z \quad \text{and} \quad |z'\rangle\langle z'| \otimes \tau_z$$

would also be quantum computationally distinguishable for at least one choice of  $z$ , which directly contradicts the concealing property of the commitment scheme. We have therefore proved that the processes described in Figures 6.4 and 6.5 are computationally indistinguishable. For clarification, notice that when we replace  $z$  by  $z'$ , the input to the classical zero-knowledge proof could become a negative instance. However,  $S_3$  cannot distinguish between a yes- or no-instance here, for otherwise the hiding property of the commitment would be broken.

Before proceeding to the next step of the proof, it will be convenient to simplify the description of the process illustrated in Figure 6.5 without making any changes to the process itself. First, recall that  $P_0$  is obtained by first performing the encoding steps described in section 4.1, followed by the formation of the commitment

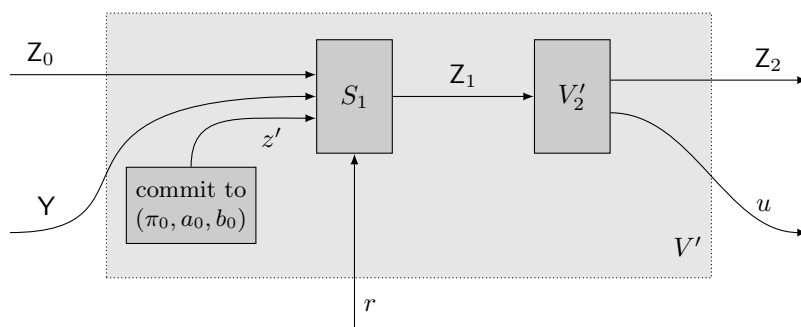


FIG. 6.7. The commitment  $z'$  to a fixed tuple  $(\pi_0, a_0, b_0)$ , the simulator  $S_1$ , and the dishonest verifier action  $V'_2$  may be merged into a single efficiently implementable action  $V'$  that represents an attack against the encoding scheme.

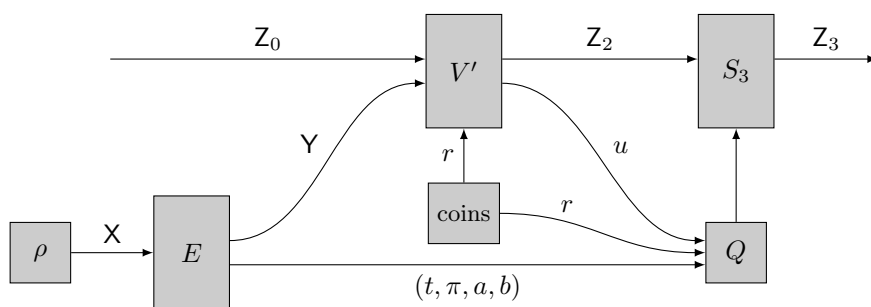


FIG. 6.8. After making the simplifications described in the text, a process identical to the one described in Figure 6.5 is obtained. The boxed labeled  $E$  represents the encoding step performed by the prover, as described in section 4.1, and the box labeled  $V'$  denotes the merger of  $S_1$ ,  $V'_2$ , and the formation of the dummy commitment.

$z = \text{commit}((\pi, a, b), s)$  (along with the random string  $s$  used to form this commitment). However, given that the commitment  $z$  and the random string  $s$  are discarded in the process described in Figure 6.5, we may as well replace  $P_0$  with the process that performs just the encoding steps alone, without the formation of the commitment. We will name this process  $E$ , and in the interest of clarity let us state explicitly that  $E$  is the process that takes  $X$  as input and outputs  $Y$  along with  $(t, \pi, a, b)$ , as described in section 4.1. Second, we may merge the commitment to the fixed tuple  $(\pi_0, a_0, b_0)$ , the simulator  $S_1$ , and the cheating verifier action  $V'_2$  to form the single, efficiently implementable action  $V'$  as suggested by Figure 6.7. The process resulting from these simplifications is illustrated in Figure 6.8.

**Step 4: Simulating an attack on the encoding scheme.** It remains to prove that, for any efficiently implementable actions  $V'$  and  $S_3$ , the channel implemented by the process described by Figure 6.8 can be efficiently simulated. In fact, it will be possible to efficiently simulate this channel with statistical accuracy, not just in a computationally indistinguishable sense. This is not surprising: we have claimed that the computational zero-knowledge property of our proof system is based on a computationally concealing commitment scheme, and the uses of the commitment scheme have all been eliminated from consideration by the steps above.

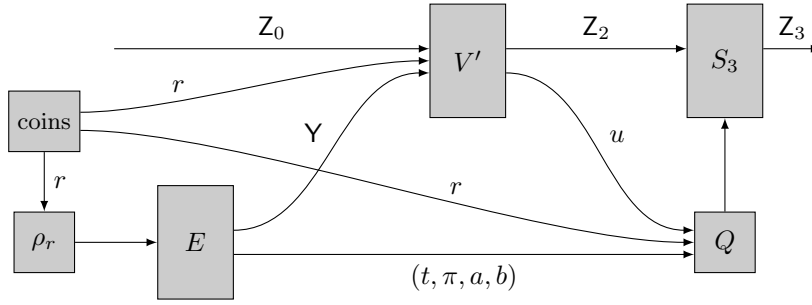


FIG. 6.9. The simulation of the process shown in Figure 6.8 is nearly identical to that process, except that it uses the random string  $r$  to encode a state  $\rho_r$  that is guaranteed to pass the challenge corresponding to  $r$ , rather than encoding the witness state  $\rho$ .

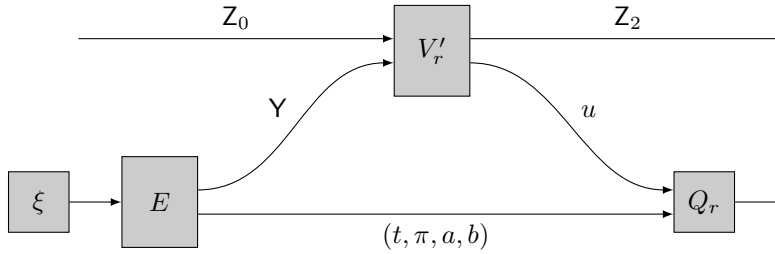


FIG. 6.10. An arbitrary  $n$ -qubit state  $\xi$  is encoded, and the cheating verifier  $V'$  and predicate  $Q$  for a fixed choice of a string  $r$  interact as depicted. It will be proved that the channels obtained by substituting  $\rho$  and  $\rho_r$  for  $\xi$  are approximately equal.

At this point we may describe the simulator directly: it is illustrated in Figure 6.9, and it represents the most straightforward approach to obtaining a simulator.

This simulator differs from the process described in Figure 6.8 in that it uses the output of the random string generator to choose a quantum state that, once encoded, passes the randomly selected Hamiltonian term challenge with certainty. It is trivial to efficiently prepare such a state given the string  $r$ . It remains to prove that the channel implemented by the simulator described in Figure 6.9 is indistinguishable from the channel implemented by the process described in Figure 6.8. By convexity it suffices to prove that this is so for every fixed choice of the string  $r$ . Moreover, it suffices to prove that the two processes obtained by removing  $S_3$  from Figures 6.8 and 6.9, so that the outputs of the processes are  $Z_2$  and the output bit of  $Q$ , are indistinguishable—for composing those two processes with the same action  $S_3$  cannot make them more distinguishable.

With this goal in mind, consider the process described in Figure 6.10, in which an arbitrary state  $\xi$  is encoded (corresponding either to  $\rho$  or  $\rho_r$  in Figures 6.8 and 6.9), and the string  $r$  is fixed (which has been indicated by the substitution of  $V'_r$  and  $Q_r$  for  $V'$  and  $Q$ , respectively). We will prove that the channel implemented by any such process can have only a limited dependence on the state  $\xi$ .

More specifically, let us assume that  $\xi_0$  and  $\xi_1$  are arbitrary  $n$ -qubit states, let  $p_0$  and  $p_1$  denote the probabilities with which these two states would pass the challenge determined by  $r$  for an honest prover and verifier pair (i.e.,  $p_i = 1 - \langle \xi_i | H_r | \xi_i \rangle$ ,  $i = 0, 1$ ). Let  $\Psi_0$  and  $\Psi_1$  denote the channels from  $Z_0$  to  $Z_2$  together with the output bit of the

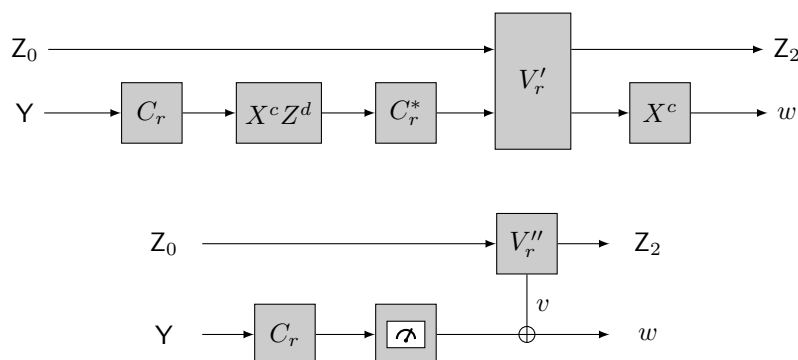


FIG. 6.11. The prover's one-time pad merged with the cheating verifier operation  $V'_r$ . Averaging over random choices of  $c$  and  $d$  results in a process that can alternatively be described as illustrated in the lower diagram. In this process,  $V''_r$  represents a so-called quantum instrument, which transforms  $Z_0$  into  $Z_2$  and produces a classical measurement outcome. In this case, this classical measurement outcome is XORed onto the string produced by a standard basis measurement of those qubits that correspond to the Hamiltonian term given by  $r$ . (Here, one should interpret  $C_r$  and  $C_r^*$  as referring to the transversal application of the corresponding Clifford operation, and interpret the rightmost  $X^c$  operation in the top circuit as a classical XOR from the relevant bits of  $c$  onto the verifier's output string  $u$ .)

predicate  $Q_r$  that are implemented by the process shown in Figure 6.10 when  $\xi_0$  or  $\xi_1$  is substituted for  $\xi$ , respectively.

We claim that if  $|p_0 - p_1|$  is negligible, then the distance  $\|\Psi_0 - \Psi_1\|_\diamond$  is also negligible. The two steps that follow establish that this claim is true. By the assumption that the prover initially holds a witness state  $\rho$  that satisfies every Hamiltonian term with probability exponentially close to 1, this will complete the proof.

**Step 5: Twirling the cheating verifier.** To prove the fact suggested above regarding the channel implemented by Figure 6.10, we will naturally need to make use of the specific properties of the encoding scheme, which have not played an important role in the analysis thus far. The first step is to recognize that the effect of the prover's one time pad is to *twirl*<sup>3</sup> the verifier as Figure 6.11 illustrates.

More specifically, the last step of the encoding process is the quantum one-time pad: the prover independently chooses one of the Pauli operations  $\mathbb{1}$ ,  $X$ ,  $Z$ , or  $XZ$  for each qubit of  $Y$  and applies that operation, storing the randomly selected strings  $a, b \in \Sigma^{2Nn}$ . With respect to the Clifford operation  $C_r$  associated with the randomly selected challenge (determined by the string  $r$ ), the prover computes the pair  $(c, d)$  for which it holds that

$$(6.3) \quad X^a Z^b = (C_r^{\otimes 2N})^* X^c Z^d (C_r^{\otimes 2N}).$$

The first step in computing the predicate  $Q_r$  is the application of  $X^c$  to the string  $u$ , which is supposed to represent the outcome of a standard basis measurement of a subset of the qubits after the transversal application of  $C_r$  to the corresponding qubits in the register  $Y$ . The resulting string  $w = u \oplus c_{i_1} \cdots c_{i_k}$  is then fed into the predicate  $R_r$  described previously. Merging the Clifford operation  $C_r^*$  with the cheating verifier

<sup>3</sup>The term twirl is commonly used in quantum information theory to describe a process whereby a symmetrization over a collection of randomly chosen unitary operations has a particular effect on a state or channel. Twirled states and channels often take on a significantly simpler form than the original state or channel prior to twirling. See examples in [2, 10].

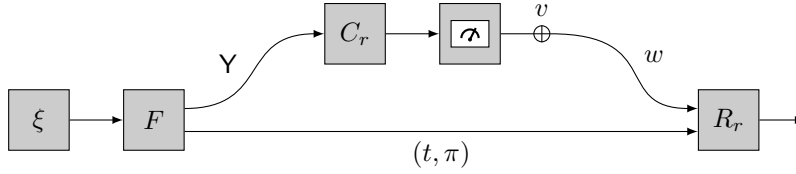


FIG. 6.12. An XOR attack against the prover's encoding scheme without the one-time pad. The transformation  $F$  denotes the first three steps of the prover's encoding scheme.

operation  $V'_r$ , then averaging over  $c$  and  $d$  chosen uniformly at random (which is equivalent to averaging over  $a$  and  $b$  chosen uniformly at random), one obtains a process of the form illustrated in the lower diagram in Figure 6.11. In greater detail, the channel obtained by first performing  $Z^d$  on  $Y$  for  $d$  chosen uniformly at random, followed by the operation  $C_r^*$  performed on  $Y$ , followed by  $V'_r$  on  $(Z_0, Y)$ , is a channel that effectively treats  $Y$  as if it were classical, so that it can be expressed as

$$(6.4) \quad \sum_{y,z} \Phi_{y,z} \otimes \Delta_{y,z},$$

where each  $\Phi_{y,z}$  is a completely positive map and  $\Delta_{y,z}$  is defined as

$$(6.5) \quad \Delta_{y,z}(Y) = |z\rangle\langle y|Y|y\rangle\langle z|$$

for every  $y \in \Sigma^{2nN}$  and  $z \in \Sigma^{2kN}$ . For a uniformly selected string  $c$ , composing this operation with the XOR operations represented by  $X^c$  yields the operation

$$(6.6) \quad \sum_{y,z} \Phi_{y,z} \otimes \sum_c \Delta_{y \oplus c, z \oplus c_{i_1} \cdots c_{i_k}} = \sum_{y,z} \Phi_{y,z} \otimes \sum_c \Delta_{c, c_{i_1} \cdots c_{i_k} \oplus y_{i_1} \cdots y_{i_k} \oplus z},$$

which has the form suggested in Figure 6.11 (for  $v = y_{i_1} \cdots y_{i_k} \oplus z$ ).

By the observation we have just made, it suffices to consider processes of the form described in Figure 6.12, in which an  $n$ -qubit state  $\xi$  is encoded as described by the first three steps in the prover's encoding procedure (but not including the one-time pad), the Clifford operation  $C_r$  (for a fixed choice of  $r$ ) is applied transversally to the resulting register, and the qubits on which those transversal Clifford operations act are measured with respect to the standard basis. For some arbitrary but fixed string  $v$ , the XOR of the outcome of this measurement with  $v$  is fed into the predicate  $R_r$ . The process outputs a single bit, obtained by evaluating the predicate  $R_r$ .

**Step 6: Encoding security under XOR attacks.** Now let us return to the claim made previously, in which  $\xi_0$  and  $\xi_1$  represent  $n$ -qubit states,  $p_0$  and  $p_1$  denote the probabilities with which these two states would pass the challenge determined by  $r$  (for an honest prover and verifier pair), and  $\Psi_0$  and  $\Psi_1$  denote the channels implemented by the process shown in Figure 6.10 when  $\xi_0$  or  $\xi_1$  is substituted for  $\xi$ , respectively. If it is the case that the distributions of output bits obtained by substituting  $\xi_0$  and  $\xi_1$  for  $\xi$  in Figure 6.12 have negligible statistical difference, then it follows that the difference  $\|\Psi_0 - \Psi_1\|_\diamond$  is also negligible. It therefore remains to argue that the distributions obtained by substituting  $\xi_0$  and  $\xi_1$  into Figure 6.12 have negligible statistical difference.

Before finishing off the last step of the analysis, it is helpful to consider the possible outcomes of the measurement, the definition of  $R_r$ , and the behavior of the

procedure described in Figure 6.12 when  $v = 0 \cdots 0$  is the all-zero string. For any choice of  $\xi$ , the measurement is guaranteed to yield a string of length  $2kN$  taking the form  $u_{i_1} \cdots u_{i_k}$ , where  $u_{i_1}, \dots, u_{i_k} \in \Sigma^{2N}$  and  $(i_1, \dots, i_k)$  index the qubits on which  $C_r$  acts nontrivially. With respect to a particular choice of  $(t, \pi)$ , if we define strings  $y_i, z_i \in \Sigma^N$  for each  $i \in \{i_1, \dots, i_k\}$  so that  $\pi(y_i z_i) = u_i$ , then these two conditions will necessarily be met:

1.  $y_i \in \mathcal{D}_N$  for every  $i \in \{i_1, \dots, i_k\}$ , and
2.  $\langle z_{i_1} \cdots z_{i_k} \mid C_r^{\otimes N} \mid t_{i_1} \cdots t_{i_k} \rangle \neq 0$ .

Moreover, in the case that  $r$  determines a Hamiltonian term challenge, the event that  $y_i \in \mathcal{D}_N^1$  for at least one index  $i \in \{i_1, \dots, i_k\}$  is equivalent to  $\xi$  passing this challenge. Thus, in the case that  $v = 0 \cdots 0$ , the process described in Figure 6.12 outputs the bit 1 with precisely the probability that an honest prover and verifier pair would result in acceptance, assuming the prover's initial state is  $\xi$  and  $r$  is selected as a random string determining the challenge.

Now let us assume that  $v$  is a nonzero string, and let us consider two cases: the first is that the Hamming weight  $|v|_1$  of  $v$  satisfies  $|v|_1 < K$  for  $K$  being the minimum Hamming weight of a nonzero codeword in  $\mathcal{D}_N$ , and the second case is that  $|v|_1 \geq K$ .

If it is the case that  $|v|_1 < K$ , then there are two possible ways that the value of the predicate  $R_r$  could change in comparison to the case  $v = 0 \cdots 0$ . In both cases, if there is a change, it must be from 1 to 0, caused by conditions 1 or 2 above being violated. The first case is that one or more bits in one of the codewords  $y_{i_1}, \dots, y_{i_k}$  are flipped, causing condition 1 to be violated. The second case is that a measurement outcome for the trap qubits is obtained that potentially violates condition 2. Note that it is not possible that condition 1 remains satisfied while the Hamiltonian term challenge condition that  $y_i \in \mathcal{D}_N^1$  for at least one index  $i \in \{i_1, \dots, i_k\}$  changes, as such a change would require at least  $K$  bit-flips to cause a logical change in valid codewords. It is unimportant for the purposes of the analysis to determine the probability with which one of the two conditions becomes violated, except to observe that it is independent of  $\xi$ . (In somewhat more detail, the string  $v$  may be written as  $v = v_{i_1} \cdots v_{i_k}$ , and the probability that neither of the two conditions is affected is given by the probability that  $\pi^{-1}(v_i)$  places no 1's within the first  $N$  bits or over a trap qubit left in a standard basis state within the second  $N$  bits, for a random choice of  $\pi$  and for each  $i \in \{i_1, \dots, i_k\}$ .)

If it is the case that  $|v|_1 \geq K$ , then there is a possibility that, in comparison to the functioning of the process for  $v = 0 \cdots 0$ , the Hamiltonian term challenge condition that  $y_i \in \mathcal{D}_N^1$  for at least one index  $i \in \{i_1, \dots, i_k\}$  could be affected. That is,  $v$  has enough Hamming weight to affect the logical values represented by the codewords  $y_{i_1}, \dots, y_{i_k}$ . However, as we will show, the assumption that  $|v|_1 \geq K$  necessarily leads to a negligible probability that the second condition remains satisfied—for a string  $v$  having Hamming weight  $K$  or higher, the probability that none of the traps is sprung is exponentially small. In order to argue that this is so, we require the following simple lemma.

**LEMMA 6.1.** *Let  $k$  be a positive integer, let  $C$  be a Clifford operation on  $k$  qubits, and let  $j \in \{1, \dots, k\}$ . There exists a string  $t \in \{0, +, \odot\}^k$ , a bit  $a \in \Sigma$ , and pure states  $|\phi_0\rangle$  and  $|\phi_1\rangle$  on  $j-1$  qubits and  $k-j$  qubits, respectively, so that*

$$(6.7) \quad C|t\rangle = |\phi_0\rangle|a\rangle|\phi_1\rangle.$$

*Equivalently, there is a choice of  $t$  so that the  $j$ th qubit of  $C|t\rangle$  is left in a standard basis state.*

*Proof.* The lemma is equivalent to the existence of a string  $t$  so that  $|t\rangle$  is an eigenvector of the operator

$$(6.8) \quad C^*(\mathbb{1}^{\otimes(j-1)} \otimes Z \otimes \mathbb{1}^{\otimes(k-j)})C.$$

As the Clifford group normalizes the Pauli group, the operator (6.8) is a scalar multiple of a tensor product of Pauli operators and identity operators. The lemma follows from the observation that  $t$  may be chosen so that each  $|t_1\rangle, \dots, |t_k\rangle$  is an eigenvector of the Pauli operator in the corresponding position.  $\square$

By this lemma, one finds that for a random choice of  $t \in \{0, +, \odot\}^{kN}$ , and for any  $k$ -qubit Clifford operation  $C$  applied transversally to  $|t\rangle$ , each qubit is left in a standard basis state with probability at least  $3^{-k}$ , and for any choice of  $N$  or fewer qubits acted on by distinct Clifford operations these events are independent. In greater detail, if the qubits

$$(6.9) \quad (Z_1^1, \dots, Z_1^k), \dots, (Z_N^1, \dots, Z_N^k)$$

are initialized to the state  $|t\rangle$  for  $t \in \{0, +, \odot\}^{kN}$  chosen uniformly at random, and the  $k$ -qubit Clifford operation  $C$  is applied independently to each  $k$ -tuple of qubits, then each qubit is left in a standard basis state with probability at least  $3^{-k}$ , and the states of the  $k$ -tuples of qubits are independent.

Now we return to the analysis for a string  $v$  of length  $2kN$  having Hamming weight at least  $K$ . By virtue of the fact just mentioned, it is straightforward to obtain a negligible upper bound on the probability for the process described in Figure 6.12 to output 1. As this event requires that a random choice of the permutation  $\pi$  leaves none of the 1-bits of  $v$  in positions corresponding to trap qubits left in standard basis states by the transversal action of  $C_r$ , we find that the probability to output 1 is exponentially small in  $K$ . In particular, this probability is at most

$$(6.10) \quad \left(1 - \frac{1}{3^{k+1}}\right)^{K/k} = \exp(-\varepsilon(k)K),$$

where  $\varepsilon(k)$  denotes a positive real number depending on  $k$  (which the reader will recall is constant and may be taken to be  $k = 5$ ) but not  $K$ .

From a consideration of the two cases just presented, we may conclude the following. Suppose as before that  $\xi_0$  and  $\xi_1$  are  $n$ -qubit states that may be substituted for  $\xi$  in Figure 6.12, and that the probabilities  $p_0$  and  $p_1$  for these states to pass the challenge determined by a fixed choice of  $r$  have negligible difference. Let us write  $q_0(v)$  and  $q_1(v)$ , respectively, to denote the probability that the process described in Figure 6.12 outputs 1. As noted before, it holds that  $p_0 = q_0(0 \cdots 0)$  and  $p_1 = q_1(0 \cdots 0)$ . For any choice of  $v$  satisfying  $|v|_1 < K$ , we have that  $q_0(v) = \beta(v)q_0(0 \cdots 0)$  and  $q_1(v) = \beta(v)q_1(0 \cdots 0)$  for  $\beta(v) \in (0, 1)$  that is independent of  $\xi_0$  and  $\xi_1$ . Finally, for any choice of  $v$  satisfying  $|v|_1 \geq K$ , we have that  $q_0(v)$  and  $q_1(v)$  are both negligible. It therefore follows that the difference  $|q_0(v) - q_1(v)|$  is negligible in all cases, which completes the proof.

**7. Conclusion.** This paper gives a zero-knowledge proof system for any problem in QMA assuming the existence of a quantum computationally concealing and unconditionally binding commitment scheme. Such a commitment scheme can be obtained

assuming quantum-secure one-way permutations [1] (or injections more generally). It also appears feasible to use a commitment scheme with an *interactive* commit phase, such as Naor's two-message commitment scheme [47] based on a pseudorandom generator. This would reduce the zero-knowledge protocol to a quantum-secure one-way function [34, 50, 61], and we leave this for further verification. We conclude with a few open questions and future directions.

1. Our proof system inherits the soundness error of straightforward verification procedure for the local Clifford–Hamiltonian problem, which is to randomly select a Hamiltonian term and perform a measurement corresponding to it. When an arbitrary QMA problem is reduced to the local Hamiltonian problem, the resulting soundness error may potentially be large (polynomially bounded away from 1). Can a zero-knowledge proof system for any QMA problem be obtained with small soundness error while maintaining the other features of our proof system (e.g., constant round of communications)?

We note that if a prover has polynomially many copies of a valid quantum witness, then a parallel repetition of our proof system may yield a constant round zero-knowledge proof system having small soundness error for any QMA problem—but this would require a parallel repetition result concerning zero-knowledge proof systems for NP secure against quantum attacks. Analogous results for zero-knowledge proofs for NP against classical attacks are known [18, 24], but they involve sophisticated rewinding arguments for which known quantum rewinding techniques do not seem to be applicable.

2. Are there natural formalizations of *proofs of quantum knowledge*? Roughly speaking, one would expect such a notion to require that whenever a prover is able to prove the validity of a statement, one could construct a knowledge extractor that can extract a quantum witness given access to such a prover. (Unruh [53] has formulated a notion of *quantum proofs of knowledge* that refers to the extraction of a classical witness from a possibly malicious quantum prover, but here we are referring to the extraction of a quantum witness.) It seems plausible that our proof system could be adapted to such a notion, although we have not investigated this in depth.
3. Finally, we make one further remark on an abstract view of our proof system. Classically speaking, one can imagine a “commit-and-open” primitive where a sender commits to a message  $m$ , and later opens sufficient information so that a receiver can test a property  $\mathcal{P}(\cdot)$  on  $m$ , and nothing more. For example,  $\mathcal{P}$  can be an NP-relation  $R(x, \cdot)$  that checks if message  $m$  is a valid witness. This can be implemented easily by a standard commitment scheme and during the opening phase, the sender and receiver run a zero-knowledge proof of  $R(x, m) = 1$  instead of the standard opening. Our proof system, which combines a commitment scheme and classical zero-knowledge proofs for NP, can be viewed as a quantum analogue. Namely, we commit to a witness state and open just enough information to verify that some reduced density of the witness state falls into a specific subspace. We can only deal with properties of a very special form, and it is an interesting direction for future work to generalize and find applications of this sort of primitive.



**Acknowledgments.** We thank Michael Beverland, Sevag Gharibian, David Gosset, Yi-Kai Liu, and Bei Zeng for helpful conversations.

## REFERENCES




- [1] M. ADCOCK AND R. CLEVE, *A quantum Goldreich-Levin theorem with cryptographic applications*, in Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 2285, Springer, Berlin, 2002, pp. 323–334, [https://doi.org/10.1007/3-540-45841-7\\_26](https://doi.org/10.1007/3-540-45841-7_26).
- [2] D. AHARONOV, M. BEN-OR, AND E. EBAN, *Interactive proofs for quantum computations*, in Innovations in Computer Science, ACM, New York, 2010, pp. 453–469.
- [3] D. AHARONOV, A. KITAIEV, AND N. NISAN, *Quantum circuits with mixed states*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 20–30, <https://doi.org/10.1145/276698.276708>.
- [4] A. AMBAINIS, M. MOSCA, A. TAPP, AND R. DE WOLF, *Private quantum channels*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2000, pp. 547–553, <https://doi.org/10.1109/SFCS.2000.892142>.
- [5] H. BARNUM, C. CRÉPEAU, D. GOTTESMAN, A. SMITH, AND A. TAPP, *Authentication of quantum messages*, in Proceedings of the 43th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2002, pp. 449–458, <https://doi.org/10.1109/SFCS.2002.1181969>.
- [6] M. BEN-OR, C. CRÉPEAU, D. GOTTESMAN, A. HASSIDIM, AND A. SMITH, *Secure multiparty quantum computation with (only) a strict honest majority*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2006, pp. 249–260, <https://doi.org/10.1109/FOCS.2006.68>.
- [7] M. BEN-OR, O. GOLDREICH, S. GOLDWASSER, J. HÅSTAD, J. KILIAN, S. MICALI, AND P. ROGAWAY, *Everything provable is provable in zero-knowledge*, in Advances in Cryptology – CRYPTO 1988, Lecture Notes in Comput. Sci. 403, Springer, Berlin, 1990, pp. 37–56, [https://doi.org/10.1007/0-387-34799-2\\_4](https://doi.org/10.1007/0-387-34799-2_4).
- [8] M. BLUM, *Coin flipping by telephone a protocol for solving impossible problems*, ACM SIGACT News, 15 (1983), pp. 23–27, <https://doi.org/10.1145/1008908.1008911>.
- [9] S. BRAVYI, *Efficient algorithms for a quantum analogue of 2-SAT*, Contemp. Math., 536 (2011), pp. 33–48, <https://doi.org/10.1090/conm/536/10552>.
- [10] A. BROADBENT, *How to verify a quantum computation*, Theory Comput., 14 (2018), pp. 1–37.
- [11] A. BROADBENT, G. GUTOSKI, AND D. STEBILA, *Quantum one-time programs*, in Advances in Cryptology – CRYPTO 2013, Lecture Notes in Comput. Sci. 8043, Springer, Berlin, 2013, pp. 344–360, [https://doi.org/10.1007/978-3-642-40084-1\\_20](https://doi.org/10.1007/978-3-642-40084-1_20).
- [12] A. BROADBENT, Z. JI, F. SONG, AND J. WATROUS, *Zero-knowledge proof systems for QMA*, in Proceedings of the 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Piscataway, NJ, 2016, pp. 31–40.
- [13] A. CHAILLOUX AND I. KERENIDIS, *Increasing the power of the verifier in quantum zero knowledge*, in IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, Wadern, Germany, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.
- [14] I. DAMGÅRD, S. FEHR, AND L. SALVAIL, *Zero-knowledge proofs and string commitments withstanding quantum attacks*, in Advances in Cryptology – CRYPTO 2004, Lecture Notes in Comput. Sci. 3152, Springer, Berlin, 2004, pp. 254–272, [https://doi.org/10.1007/978-3-540-28628-8\\_16](https://doi.org/10.1007/978-3-540-28628-8_16).
- [15] I. DAMGÅRD AND C. LUNEMANN, *Quantum-secure coin-flipping and applications*, in Advances in Cryptology – ASIACRYPT 2009, Lecture Notes in Comput. Sci. 5912, Springer, Berlin, 2009, pp. 52–69, [https://doi.org/10.1007/978-3-642-10366-7\\_4](https://doi.org/10.1007/978-3-642-10366-7_4).
- [16] F. DUPUIS, J. B. NIELSEN, AND L. SALVAIL, *Secure two-party quantum evaluation of unitaries against specious adversaries*, in Advances in Cryptology – CRYPTO 2010, Lecture Notes in Comput. Sci. 6223, Springer, Berlin, 2010, pp. 685–706, [https://doi.org/10.1007/978-3-642-14623-7\\_37](https://doi.org/10.1007/978-3-642-14623-7_37).
- [17] F. DUPUIS, J. B. NIELSEN, AND L. SALVAIL, *Actively secure two-party evaluation of any quantum operation*, in Advances in Cryptology – CRYPTO 2012, Lecture Notes in Comput. Sci. 7417, Springer, Berlin, 2012, pp. 794–811, [https://doi.org/10.1007/978-3-642-32009-5\\_46](https://doi.org/10.1007/978-3-642-32009-5_46).
- [18] U. FEIGE AND A. SHAMIR, *Zero knowledge proofs of knowledge in two rounds*, in Advances in Cryptology – CRYPTO 1989, Lecture Notes in Comput. Sci. 435, Springer, New York, 1990, pp. 526–544, [https://doi.org/10.1007/0-387-34805-0\\_46](https://doi.org/10.1007/0-387-34805-0_46).

- [19] J. F. FITZSIMONS, M. HAJDIŠEK, AND T. MORIMAE, *Post hoc verification with a single prover*, Phys. Rev. Lett., 120 (2018), 040501, <https://doi.org/10.1103/PhysRevLett.120.040501>.
- [20] J. FITZSIMONS, Z. JI, T. VIDICK, AND H. YUEN, *Quantum proof systems for iterated exponential time, and beyond*, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019, pp. 473–480, <https://doi.org/10.1145/3313276.3316343>.
- [21] C. A. FUCHS AND A. PERES, *Quantum-state disturbance versus information gain: Uncertainty relations for quantum information*, Phys. Rev. A(3), 53 (1996), pp. 2038–2045, <https://doi.org/10.1103/PhysRevA.53.2038>.
- [22] O. GOLDBREICH, *Foundations of Cryptography I: Basic Tools*, Cambridge University Press, Cambridge, 2001, <https://doi.org/10.1017/CBO9780511546891>.
- [23] O. GOLDBREICH, *Foundations of Cryptography II: Basic Applications*, Cambridge University Press, Cambridge, 2004, <https://doi.org/10.1017/CBO9780511721656>.
- [24] O. GOLDBREICH AND A. KAHAN, *How to construct constant-round zero-knowledge proof systems for NP*, J. Cryptology, 9 (1996), pp. 167–189, <https://doi.org/10.1007/BF00208001>.
- [25] O. GOLDBREICH, S. MICALI, AND A. WIGDERSON, *How to play ANY mental game*, in Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 218–229, <https://doi.org/10.1145/28395.28420>.
- [26] O. GOLDBREICH, S. MICALI, AND A. WIGDERSON, *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems*, Journal of the ACM, 38 (1991), pp. 690–728, <https://doi.org/10.1145/116825.116852>.
- [27] O. GOLDBREICH AND Y. OREN, *Definitions and properties of zero-knowledge proof systems*, J. Cryptology, 7 (1994), pp. 1–32, <https://doi.org/10.1007/BF00195207>.
- [28] S. GOLDWASSER, S. MICALI, AND C. RACKOFF, *The knowledge complexity of interactive proof systems*, SIAM J. Comput., 18 (1989), pp. 186–208, <https://doi.org/10.1137/0218012>.
- [29] S. GOLDWASSER AND M. SIPSER, *Private coins versus public coins in interactive proof systems*, in Proceedings of the 18th Annual ACM Symposium on Theory of Computing, ACM, New York, 1986, pp. 59–68, <https://doi.org/10.1145/12130.12137>.
- [30] D. GOSSET AND D. NAGAJ, *Quantum 3-SAT is QMA<sub>1</sub>-complete*, SIAM J. Comput., 45 (2016), pp. 1080–1128, <https://doi.org/10.1137/140957056>.
- [31] D. GOTTESMAN, *The Heisenberg representation of quantum computers*, in Group 22: Proceedings of the 22nd International Colloquium on Group Theoretical Methods in Physics, International Press, Cambridge, MA, 1998, pp. 32–43.
- [32] S. HALLGREN, A. KOLLA, P. SEN, AND S. ZHANG, *Making classical honest verifier zero knowledge protocols secure against quantum attacks*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II, Lecture Notes in Comput. Sci. 5126, Springer, Berlin, 2008, pp. 592–603, [https://doi.org/10.1007/978-3-540-70583-3\\_48](https://doi.org/10.1007/978-3-540-70583-3_48).
- [33] S. HALLGREN, A. SMITH, AND F. SONG, *Classical cryptographic protocols in a quantum world*, Int. J. Quantum Inf., 13 (2015), 1550028, <https://doi.org/10.1142/S0219749915500288>.
- [34] J. HÅSTAD, R. IMPAGLIAZZO, L. A. LEVIN, AND M. LUBY, *A pseudorandom generator from any one-way function*, SIAM J. Comput., 28 (1999), pp. 1364–1396, <https://doi.org/10.1137/S0097539793244708>.
- [35] R. IMPAGLIAZZO, *A personal view of average-case complexity*, in Proceedings of 10th Annual IEEE Structure in Complexity Theory Conference, IEEE Computer Society, Los Alamitos, CA, 1995, pp. 134–147, <https://doi.org/10.1109/SCT.1995.514853>.
- [36] Z. JI, *Compression of quantum multi-prover interactive proofs*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, ACM, New York, 2017, pp. 289–302, <https://doi.org/10.1145/3055399.3055441>.
- [37] J. KEMPE, A. KITAEV, AND O. REGEV, *The complexity of the local Hamiltonian problem*, SIAM J. Comput., 35 (2006), pp. 1070–1097, <https://doi.org/10.1137/S0097539704445226>.
- [38] J. KEMPE AND O. REGEV, *3-local Hamiltonian is QMA-complete*, Quantum Inf. Comput., 3 (2003), pp. 258–264, <http://portal.acm.org/citation.cfm?id=2011541>.
- [39] A. Y. KITAEV, *Quantum computations: Algorithms and error correction*, Russian Math. Surveys, 52 (1997), pp. 1191–1249, <http://stacks.iop.org/0036-0279/52/i=6/a=R02>.
- [40] A. Y. KITAEV, A. H. SHEN, AND M. N. VYALYI, *Classical and Quantum Computation*, Grad. Stud. Math. 47, Amer. Math. Soc., Providence, RI, 2002.
- [41] Y.-K. LIU, *Consistency of local density matrices is QMA-complete*, in Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Lecture Notes in Comput. Sci. 4110, Springer, Berlin, 2006, pp. 438–449, [https://doi.org/10.1007/11830924\\_40](https://doi.org/10.1007/11830924_40).

- [42] C. LUNEMANN AND J. B. NIELSEN, *Fully simulatable quantum-secure coin-flipping and applications*, in Progress in Cryptology – AFRICACRYPT 2011, Lecture Notes in Comput. Sci. 6737, Springer, Berlin, 2011, pp. 21–40, [https://doi.org/10.1007/978-3-642-21969-6\\_2](https://doi.org/10.1007/978-3-642-21969-6_2).
- [43] C. MARRIOTT AND J. WATROUS, *Quantum Arthur-Merlin games*, Comput. Complexity, 14 (2005), pp. 122–152, <https://doi.org/10.1007/s00037-005-0194-x>.
- [44] T. MORIMAE, M. HAYASHI, H. NISHIMURA, AND K. FUJII, *Quantum Merlin-Arthur with Clifford Arthur*, Quantum Inf. Comput., 15 (2015), pp. 1420–1430.
- [45] T. MORIMAE, D. NAGAJ, AND N. SCHUCH, *Quantum proofs can be verified using only single-qubit measurements*, Phys. Rev. A(3), 93 (2016), 022326, <https://doi.org/10.1103/PhysRevA.93.022326>.
- [46] D. NAGAJ, P. WOCJAN, AND Y. ZHANG, *Fast amplification of QMA*, Quantum Inf. Comput., 9 (2009), pp. 1053–1068.
- [47] M. NAOR, *Bit commitment using pseudorandomness*, J. Cryptology, 4 (1991), pp. 151–158, <https://doi.org/10.1007/BF00196774>.
- [48] M. NIELSEN AND I. CHUANG, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
- [49] P. W. SHOR, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509, <https://doi.org/10.1137/S0097539795293172>.
- [50] F. SONG, *A note on quantum security for post-quantum cryptography*, in Proceedings of the 6th International Workshop on Post-Quantum Cryptography, Lecture Notes in Comput. Sci. 8772, Springer, Cham, Switzerland, 2014, pp. 246–265, [https://doi.org/10.1007/978-3-319-11659-4\\_15](https://doi.org/10.1007/978-3-319-11659-4_15).
- [51] A. STEANE, *Multi-particle interference and quantum error correction*, Proc. Royal Soc. A, 452 (1996), pp. 2551–2577, <https://doi.org/10.1098/rspa.1996.0136>.
- [52] T. VIDICK AND T. ZHANG, *Classical Zero-Knowledge Arguments for Quantum Computations*, preprint, arXiv:1902.05217, 2019.
- [53] D. UNRUH, *Quantum proofs of knowledge*, in Advances in Cryptology – EUROCRYPT 2012, Lecture Notes in Comput. Sci. 7237, Springer, Heidelberg, Germany, 2012, pp. 135–152.
- [54] J. VAN DE GRAAF, *Towards a Formal Definition of Security for Quantum Protocols*, Ph.D. thesis, Université de Montréal, Montreal, 1997.
- [55] J. WATROUS, *Limits on the power of quantum statistical zero-knowledge*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2002, pp. 459–468, <https://doi.org/10.1109/SFCS.2002.1181970>.
- [56] J. WATROUS, *PSPACE has constant-round quantum interactive proof systems*, Theoret. Comput. Sci., 292 (2003), pp. 575–588, [https://doi.org/10.1016/S0304-3975\(01\)00375-9](https://doi.org/10.1016/S0304-3975(01)00375-9).
- [57] J. WATROUS, *Quantum computational complexity*, in Encyclopedia of Complexity and Systems Science, Springer, New York, 2009, pp. 7174–7201, [https://doi.org/10.1007/978-0-387-30440-3\\_428](https://doi.org/10.1007/978-0-387-30440-3_428).
- [58] J. WATROUS, *Zero-knowledge against quantum attacks*, SIAM J. Comput., 39 (2009), pp. 25–58, <https://doi.org/10.1137/060670997>.
- [59] J. WATROUS, *Guest column: An introduction to quantum information and quantum circuits*, ACM SIGACT News, 42 (2011), pp. 52–67, <https://doi.org/10.1145/1998037.1998053>.
- [60] W. K. WOOTTERS AND W. H. ZUREK, *A single quantum cannot be cloned*, Nature, 299 (1982), pp. 802–803, <https://doi.org/10.1038/299802a0>.
- [61] M. ZHANDRY, *How to construct quantum random functions*, in Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, IEEE, Piscataway, NJ, 2012, pp. 679–687, <https://doi.org/10.1109/FOCS.2012.37>.



# Pseudorandom Quantum States

Zhengfeng Ji<sup>1</sup>() , Yi-Kai Liu<sup>2,3</sup>() , and Fang Song<sup>4</sup>()

<sup>1</sup> Centre for Quantum Software and Information, School of Software,  
Faculty of Engineering and Information Technology,  
University of Technology Sydney, Ultimo, NSW, Australia

`Zhengfeng.Ji@uts.edu.au`

<sup>2</sup> Applied and Computational Mathematics Division,  
National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA

`yi-kai.liu@nist.gov`

<sup>3</sup> Joint Center for Quantum Information and Computer Science (QuICS),  
University of Maryland, College Park, MD, USA

<sup>4</sup> Computer Science Department, Portland State University, Portland, OR, USA  
`fang.song@pdx.edu`

**Abstract.** We propose the concept of pseudorandom quantum states, which appear random to any quantum polynomial-time adversary. It offers a *computational* approximation to perfectly random quantum states analogous in spirit to cryptographic pseudorandom generators, as opposed to *statistical* notions of quantum pseudorandomness that have been studied previously, such as quantum  $t$ -designs analogous to  $t$ -wise independent distributions.

Under the assumption that quantum-secure one-way functions exist, we present efficient constructions of pseudorandom states, showing that our definition is achievable. We then prove several basic properties of pseudorandom states, which show the utility of our definition. First, we show a cryptographic no-cloning theorem: no efficient quantum algorithm can create additional copies of a pseudorandom state, when given polynomially-many copies as input. Second, as expected for random quantum states, we show that pseudorandom quantum states are highly entangled on average. Finally, as a main application, we prove that any family of pseudorandom states naturally gives rise to a private-key quantum money scheme.

## 1 Introduction

Pseudorandomness is a foundational concept in modern cryptography and theoretical computer science. A distribution  $\mathcal{D}$ , e.g., over a set of strings or functions, is called *pseudorandom* if no computationally-efficient observer can distinguish between an object sampled from  $\mathcal{D}$ , and a truly random object sampled from the uniform distribution [10, 56, 63]. Pseudorandom objects, such as pseudorandom generators (PRGs), pseudorandom functions (PRFs) and pseudorandom permutations (PRPs) are fundamental cryptographic building blocks, such as in the design of stream ciphers, block ciphers and message authentication

codes [23, 24, 27, 37, 53]. Pseudorandomness is also essential in algorithm design and complexity theory such as derandomization [32, 47].

The law of quantum physics asserts that truly random bits can be generated easily even with untrusted quantum devices [15, 41]. Is pseudorandomness, a seemingly weaker notion of randomness, still relevant in the context of quantum information processing? The answer is yes. By a simple counting argument, one needs exponentially many bits even to specify a truly random function on  $n$ -bit strings. Hence, in the *computational* realm, pseudorandom objects that offer efficiency as well as other unique characteristics and strengths are indispensable.

A fruitful line of work on pseudorandomness in the context of quantum information science has been about quantum  $t$ -designs and unitary  $t$ -designs [4, 11, 12, 16, 17, 26, 33, 40, 43–45, 59, 69]. However, while these objects are often called “pseudorandom” in the mathematical physics literature, they are actually analogous to  $t$ -wise independent random variables in theoretical computer science. Our focus in this work is a notion of *computational* pseudorandomness, and in particular suits (complexity-theoretical) cryptography.

The major difference between  $t$ -wise independence and cryptographic pseudorandomness is the following. In the case of  $t$ -wise independence, the observer who receives the random-looking object may be computationally unbounded, but only *a priori* (when the random-looking object is constructed) fixed number  $t$  samples are given. Thus, quantum  $t$ -designs satisfy an “information-theoretic” or “statistical” notion of security. In contrast, in the case of cryptographic pseudorandomness, the observer who receives the random-looking object is assumed to be computationally efficient, in that it runs in probabilistic polynomial time for an arbitrary polynomial that is chosen by the observer, *after* the random-looking object has been constructed. This leads to a “computational” notion of security, which typically relies on some complexity-theoretic assumption, such as the existence of one-way functions.

In general, these two notions,  $t$ -wise independence and cryptographic pseudorandomness, are incomparable. In some ways, the setting of cryptographic pseudorandomness imposes stronger restrictions on the observer, since it assumes a bound on the observer’s total computational effort (say, running in probabilistic polynomial time). In other ways, the setting of  $t$ -wise independence imposes stronger restrictions on the observer, since it forces the observer to make a limited number of non-adaptive “queries,” specified by the parameter  $t$ , which is usually a constant or a fixed polynomial. In addition, different distance measures are often used, e.g., trace distance or diamond norm, versus computational distinguishability.

Cryptographic pseudorandomness in quantum information, which has received relatively less study, mostly connects with quantum money and post-quantum cryptography. Pseudorandomness is used more-or-less implicitly in quantum money, to construct quantum states that look complicated to a dishonest party, but have some hidden structure that allows them to be verified by the bank [1–3, 39, 68]. In post-quantum cryptography, one natural question is whether the classical constructions such as PRFs and PRPs remain secure against quantum attacks. This is a challenging task as, for example, a quantum

adversary may query the underlying function or permutation in *superposition*. Fortunately, people have so far restored several positive results. Assuming a one-way function that is hard to invert for polynomial-time quantum algorithms, we can attain quantum-secure PRGs as well as PRFs [27, 65]. Furthermore, one can construct quantum-secure PRPs from quantum-secure PRFs using various *shuffling* constructions [57, 67].

In this work, we study pseudorandom *quantum* objects such as quantum states and unitary operators. Quantum states (in analogy to strings) and unitary operations (in analogy to functions) form continuous spaces, and the Haar measure is considered the perfect randomness on the spaces of quantum states and unitary operators. A basic question is:

*How to define and construct computational pseudorandom approximations of Haar randomness, and what are their applications?*

*Our contributions.* We propose definitions of pseudorandom quantum states (PRS's) and pseudorandom unitary operators (PRUs), present efficient constructions of PRS's, demonstrate basic properties such as no-cloning and high entanglement of pseudorandom states, and showcase the construction of private-key quantum money schemes as one of the applications.

1. We propose a suitable definition of *quantum pseudorandom states*.

We employ the notion of quantum *computational indistinguishability* to define quantum pseudorandom states. Loosely speaking, we consider a collection of quantum states  $\{|\phi_k\rangle\}$  indexed by  $k \in \mathcal{K}$ , and require that no efficient quantum algorithm can distinguish between  $|\phi_k\rangle$  for a random  $k$  and a state drawn according to the Haar measure. However, as a unique consideration in the quantum setting, we need to be cautious about *how many copies* of the input state are available to an adversary.

Classically, this is a vacuous concern for defining a pseudorandom distribution on strings, since one can freely produce many copies of the input string. The quantum no-cloning theorem, however, forbids copying an unknown quantum state in general. Pseudorandom states in terms of *single-copy* indistinguishability have been discussed in the literature (see for example [13] and a recent study [14]). Though this single-copy definition may be suitable for certain cryptographic applications, it also loses many properties of Haar random states as a purely classical distributions already satisfies the definition<sup>1</sup>.

Therefore we require that no adversary can tell a difference even given any *polynomially many* copies of the state. This subsumes the single-copy version and is strictly stronger. We gain from it many interesting properties, such as the no-cloning property and entanglement property for pseudorandom states as discussed later in the paper.

---

<sup>1</sup> For example, a uniform distribution over the computational basis state  $\{|k\rangle\}$  has an identical density matrix as a Haar random state and satisfy the single-shot definition of PRS. But distinguishing them becomes easy as soon as we have more than one copies. These states also do not appear to be hard to clone or possess entanglement.

2. We present concrete efficient constructions of PRS's with the minimal assumption that quantum-secure one-way functions exist.

Our construction uses any quantum-secure PRF  $= \{\text{PRF}_k\}_{k \in \mathcal{K}}$  and computes it into the phases of a uniform superposition state (see Eq. (8)). We call such family of PRS the *random phase states*. This family of states can be efficiently generated using the quantum Fourier transform and a phase kick-back trick. We prove that this family of state is pseudorandom by a hybrid argument. By the quantum security of PRF, the family is computationally indistinguishable from a similar state family defined by truly random functions.

We then prove that, this state family corresponding to truly random functions is statistically indistinguishable from Haar random states. Finally, by the fact that PRF exists assuming quantum-secure one-way functions, we can base our PRS construction on quantum-secure one-way functions.

We note that Aaronson [1, Theorem 3] has described a similar family of states, which uses some polynomial function instead of a PRF in the phases. In that construction, however, the size of the state family depends on (i.e., has to grow with) the adversary's number of queries that the family wants to tolerate. It therefore fails to satisfy our definition, in which any polynomial number queries independent of the family are permitted.

3. We prove *cryptographic no-cloning theorems* for PRS's, and they give a simple and generic construction of private-key quantum money schemes based on any PRS.

We prove that a PRS remains pseudorandom, even if we additionally give the distinguisher an oracle that reflects about the given state (i.e.,  $O_\phi := \mathbb{1} - 2|\phi\rangle\langle\phi|$ ). This establishes the equivalence between the standard and a strong definition of PRS's. Technically, this is proved using the fact that with polynomially many copies of the state, one can approximately simulate the reflection oracle  $O_\phi$ .

We obtain general *cryptographic no-cloning theorems* of PRS's both with and without the reflection oracle. The theorems roughly state that given any polynomially many copies of pseudorandom states, no polynomial-time quantum algorithm can produce even one more copy of the state. We call them cryptographic no-cloning theorems due to the computational nature of our PRS. The proofs of these theorems use SWAP tests in the reduction from a hypothetical cloning algorithm to an efficient distinguishing algorithm violating the definition of PRS's.

Using the strong pseudorandomness and the cryptographic no-cloning theorem with reflection oracle, we show that any PRS immediately gives a *private-key quantum money scheme*. While much attention has been focused on public-key quantum money [1–3, 39, 68], we emphasize that private-key quantum money is already non-trivial. Early schemes for private-key quantum money due to Wiesner and others were not *query secure*, and could be broken by online attacks [9, 20, 38, 61]. Aaronson and Christiano finally showed a query-secure scheme in 2012, which achieves information-theoretic security in the random oracle model, and computational security in the standard model [2]. They used a specific construction based on hidden sub-



space states, whereas our construction (which is also query-secure) is more generic and can be based on any PRS. The freedom to choose and tweak the underlying pseudorandom functions or permutations in the PRS may motivate and facilitate the construction of public-key quantum money schemes in future work.

4. We show that pseudorandom states are highly entangled.

It is known that a Haar random state is entangled with high probability. We establish a similar result for any family of pseudorandom states. Namely, the states in any PRS family are entangled on average. It is shown that the expected Schmidt rank for any PRS is superpolynomial in  $\kappa$  and that the expected min entropy and von Neumann entropy are of the order  $\omega(\log \kappa)$  where  $\kappa$  is the security parameter. This is yet another evidence of the suitability of our definition.

The proof again rests critically on that our definition grants multiple copies to the distinguisher—if the expected entanglement is low, then SWAP test with respect to the corresponding subsystems of two copies of the state will serve as a distinguisher that violates the definition.

5. We propose a definition of *quantum pseudorandom unitary operators* (PRUs).

We also present candidate constructions of PRUs (without a proof of security), by extending our techniques for constructing PRS's.

Loosely speaking, these candidate PRUs resemble unitary  $t$ -designs that are constructed by interleaving random permutations with the quantum Fourier transform [26], or by interleaving random diagonal unitaries with the Hadamard transform [43, 44], and iterating this construction several times. We conjecture that a PRU can be obtained in this way, using only a constant number of iterations. This is in contrast to unitary  $t$ -designs, where a parameter counting argument suggests that the number of iterations must grow with  $t$ . This conjecture is motivated by examples such as the Luby-Rackoff construction of a pseudorandom permutation using multi-round Feistel network built using a PRF.

**Table 1.** Summary of various notions that approximate true randomness

	Classical	Quantum
<i>True randomness</i>	Uniform distribution	Haar measure
<i>t-wise independence</i>	$t$ -wise independent random variables	Quantum $t$ -designs
<i>Pseudorandomness</i>	PRGs PRFs, PRPs	( <i>this work</i> ) PRS's PRUs

*Discussion.* We summarize the mentioned variants of randomness in Table 1. The focus of this work is mostly about PRS's and we briefly touch upon PRUs. We



view our work as an initial step and anticipate further fundamental investigation inspired by our notion of pseudorandom states and unitary operators.

We mention some immediate open problems. First, can we prove the security of our candidate PRU constructions? The techniques developed in quantum unitary designs [12, 26] seem helpful. Second, are quantum-secure one-way functions necessary for the construction of PRS's? Third, can we establish security proofs for more candidate constructions of PRS's? Different constructions may have their own special properties that may be useful in different settings. It is also interesting to explore whether our quantum money construction may be adapted to a public-key money scheme under reasonable cryptographic assumptions. Finally, the entanglement property we prove here refers to the standard definitions of entanglement. If we approach the concept of pseudo-entanglement as a quantum analogue of pseudo-entropy for a distribution [7], can we improve the quantitative bounds?

We point out a possible application in physics. PRS's may be used in place of high-order quantum  $t$ -designs, giving a performance improvement in certain applications. For example, pseudorandom states can be used to construct toy models of quantum *thermalization*, where one is interested in quantum states that can be prepared efficiently via some dynamical process, yet have “generic” or “typical” properties as exemplified by Haar-random pure states, for instance [51]. Using  $t$ -designs with polynomially large  $t$ , one can construct states that are “generic” in an information-theoretic sense [35]. Using PRS, one can construct states that satisfy a weaker property: they are computationally indistinguishable from “generic” states, for a polynomial-time observer.

In these applications, PRS states may be more physically plausible than high-order quantum  $t$ -designs, because PRS states can be prepared in a shorter time, e.g., using a polylogarithmic-depth quantum circuit, based on known constructions for low-depth PRFs [6, 46].

## 2 Preliminaries

### 2.1 Notions

For a finite set  $\mathcal{X}$ ,  $|\mathcal{X}|$  denotes the number of elements in  $\mathcal{X}$ . We use the notion  $\mathcal{Y}^{\mathcal{X}}$  to denote the set of all functions  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . For finite set  $\mathcal{X}$ , we use  $x \leftarrow \mathcal{X}$  to mean that  $x$  is drawn uniformly at random from  $\mathcal{X}$ . The permutation group over elements in  $\mathcal{X}$  is denoted as  $S_{\mathcal{X}}$ . We use  $\text{poly}(\kappa)$  to denote the collection of polynomially bounded functions of the security parameter  $\kappa$ , and use  $\text{negl}(\kappa)$  to denote negligible functions in  $\kappa$ . A function  $\epsilon(\kappa)$  is *negligible* if for all constant  $c > 0$ ,  $\epsilon(\kappa) < \kappa^{-c}$  for large enough  $\kappa$ .

In this paper, we use a *quantum register* to name a collection of qubits that we view as a single unit. Register names are represented by capital letters in a *sans serif* font. We use  $S(\mathcal{H})$ ,  $D(\mathcal{H})$ ,  $U(\mathcal{H})$  and  $L(\mathcal{H})$  to denote the set of pure quantum states, density operators, unitary operators and bounded linear operators on space  $\mathcal{H}$  respectively. An ensemble of states  $\{(p_i, \rho_i)\}$  represents a system prepared in  $\rho_i$  with probability  $p_i$ . If the distribution is uniform, we write

the ensemble as  $\{\rho_i\}$ . The adjoint of matrix  $M$  is denoted as  $M^*$ . For matrix  $M$ ,  $|M|$  is defined to be  $\sqrt{M^*M}$ . The operator norm  $\|M\|$  of matrix  $M$  is the largest eigenvalue of  $|M|$ . The trace norm  $\|M\|_1$  of  $M$  is the trace of  $|M|$ . For two operators  $M, N \in L(\mathcal{H})$ , the Hilbert-Schmidt inner product is defined as

$$\langle M, N \rangle = \text{tr}(M^*N).$$

A quantum channel is a physically admissible transformation of quantum states. Mathematically, a quantum channel

$$\mathcal{E} : L(\mathcal{H}) \rightarrow L(\mathcal{K})$$

is a completely positive, trace-preserving linear map.

The trace distance of two quantum states  $\rho_0, \rho_1 \in D(\mathcal{H})$  is

$$\text{TD}(\rho_0, \rho_1) \stackrel{\text{def}}{=} \frac{1}{2} \|\rho_0 - \rho_1\|_1. \quad (1)$$

It is known (Holevo-Helstrom theorem [29, 30]) that for a state drawn uniformly at random from the set  $\{\rho_0, \rho_1\}$ , the optimal distinguish probability is given by

$$\frac{1 + \text{TD}(\rho_0, \rho_1)}{2}.$$

Define number  $N = 2^n$  and set  $\mathcal{X} = \{0, 1, \dots, N-1\}$ . The quantum Fourier transform on  $n$  qubits is defined as

$$F = \frac{1}{\sqrt{N}} \sum_{x, y \in \mathcal{X}} \omega_N^{xy} |x\rangle\langle y|. \quad (2)$$

It is a well-known fact in quantum computing that  $F$  can be implemented in time  $\text{poly}(n)$ .

For Hilbert space  $\mathcal{H}$  and integer  $m$ , we use  $\vee^m \mathcal{H}$  to denote the symmetric subspace of  $\mathcal{H}^{\otimes m}$ , the subspace of states that are invariant under permutations of the subsystems. Let  $N$  be the dimension of  $\mathcal{H}$  and let  $\mathcal{X}$  be the set  $\{0, 1, \dots, N-1\}$  such that  $\mathcal{H}$  is the span of  $\{|x\rangle\}_{x \in \mathcal{X}}$ . For any  $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \mathcal{X}^m$ , let  $m_j$  be the number of  $j$  in  $\mathbf{x}$  for  $j \in \mathcal{X}$ . Define state

$$|\mathbf{x}; \text{Sym}\rangle = \sqrt{\frac{\prod_{j \in \mathcal{X}} m_j!}{m!}} \sum_{\sigma} |x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(m)}\rangle. \quad (3)$$

The summation runs over all possible permutations  $\sigma$  that give different tuples  $(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(m)})$ . Equivalently, we have

$$|\mathbf{x}; \text{Sym}\rangle = \frac{1}{\sqrt{m! \prod_{j \in \mathcal{X}} m_j!}} \sum_{\sigma \in S_m} |x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(m)}\rangle. \quad (4)$$

The coefficients in the front of the above two equations are normalization constants. The set of states

$$\{|\mathbf{x}; \text{Sym}\rangle\}_{\mathbf{x} \in \mathcal{X}^m} \quad (5)$$

forms an orthonormal basis of the symmetric subspace  $\vee^m \mathcal{H}$  [58, Proposition 7.2]. This implies that the dimension of the symmetric subspace is

$$\binom{N+m-1}{m}.$$

Let  $\Pi_m^{\text{Sym}}$  be the projection onto the symmetric subspace  $\vee^m \mathcal{H}$ . For a permutation  $\sigma \in S_m$ , define operator

$$W_\sigma = \sum_{x_1, x_2, \dots, x_m \in \mathcal{X}} |x_{\sigma^{-1}(1)}, x_{\sigma^{-1}(2)}, \dots, x_{\sigma^{-1}(m)}\rangle \langle x_1, x_2, \dots, x_m|.$$

The following identity will be useful [58, Proposition 7.1]

$$\Pi_m^{\text{Sym}} = \frac{1}{m!} \sum_{\sigma \in S_m} W_\sigma. \quad (6)$$

Let  $\mu$  be the Haar measure on  $S(\mathcal{H})$ , it is known that [25, Proposition 6]

$$\int (|\psi\rangle\langle\psi|)^{\otimes m} d\mu(\psi) = \binom{N+m-1}{m}^{-1} \Pi_m^{\text{Sym}}. \quad (7)$$

## 2.2 Cryptography

In this section, we recall several definitions and results from cryptography that is necessary for this work.

Pseudorandom functions (PRF) and pseudorandom permutations (PRP) are important constructions in classical cryptography. Intuitively, they are families of functions or permutations that looks like truly random functions or permutations to polynomial-time machines. In the quantum case, we need a strong requirement that they still look random even to polynomial-time quantum algorithms.

**Definition 1 (Quantum-Secure Pseudorandom Functions and Permutations).** Let  $\mathcal{K}, \mathcal{X}, \mathcal{Y}$  be the key space, the domain and range, all implicitly depending on the security parameter  $\kappa$ . A keyed family of functions  $\{PRF_k : \mathcal{X} \rightarrow \mathcal{Y}\}_{k \in \mathcal{K}}$  is a quantum-secure pseudorandom function (QPRF) if for any polynomial-time quantum oracle algorithm  $\mathcal{A}$ ,  $PRF_k$  with a random  $k \leftarrow \mathcal{K}$  is indistinguishable from a truly random function  $f \leftarrow \mathcal{Y}^{\mathcal{X}}$  in the sense that:

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{PRF_k}(1^\kappa) = 1] - \Pr_{f \leftarrow \mathcal{Y}^{\mathcal{X}}} [\mathcal{A}^f(1^\kappa) = 1] \right| = \text{negl}(\kappa).$$

Similarly, a keyed family of permutations  $\{PRP_k \in S_{\mathcal{X}}\}_{k \in \mathcal{K}}$  is a quantum-secure pseudorandom permutation (QPRP) if for any quantum algorithm  $\mathcal{A}$  making at most polynomially many queries,  $PRP_k$  with a random  $k \leftarrow \mathcal{K}$  is indistinguishable from a truly random permutation in the sense that:

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{PRP_k}(1^\kappa) = 1] - \Pr_{P \leftarrow S_{\mathcal{X}}} [\mathcal{A}^P(1^\kappa) = 1] \right| = \text{negl}(\kappa).$$

In addition, both  $PRF_k$  and  $PRP_k$  are polynomial-time computable (on a classical computer).

**Fact 1.** *QPRFs and QPRPs exist if quantum-secure one-way functions exist.*

Zhandry proved the existence of QPRFs assuming the existence of one-way functions that are hard to invert even for quantum algorithms [65]. Assuming QPRF, one can construct QPRP using various *shuffling* constructions [57, 67]. Since a random permutation and a random function is indistinguishable by efficient quantum algorithms [64, 66], existence of QPRP is hence equivalent to existence of QPRF.

### 3 Pseudorandom Quantum States

In this section, we will discuss the definition and constructions of pseudorandom quantum states.

#### 3.1 Definition of Pseudorandom States

Intuitively speaking, a family pseudorandom quantum states are a set of random states  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$  that is indistinguishable from Haar random quantum states.

The first idea on defining pseudorandom states can be the following. Without loss of generality, we consider states in  $S(\mathcal{H})$  where  $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$  is the Hilbert space for  $n$ -qubit systems. We are given either a state randomly sampled from the set  $\{|\phi_k\rangle \in \mathcal{H}\}_{k \in \mathcal{K}}$  or a state sampled according to the Haar measure on  $S(\mathcal{H})$ , and we require that no efficient quantum algorithm will be able to tell the difference between the two cases.

However, this definition does not seem to grasp the quantum nature of the problem. First, the state family where each  $|\phi_k\rangle$  is a uniform random bit string will satisfy the definition—in both cases, the mixed states representing the ensemble are  $\mathbb{1}/2^n$ . Second, many of the applications that we can find for PRS's will not hold for this definition.

Instead, we require that the family of states looks random even if polynomially many copies of the state are given to the distinguishing algorithm. We argue that this is the more natural way to define pseudorandom states. One can see that this definition also naturally generalizes the definition of pseudorandomness in the classical case to the quantum setting. In the classical case, asking for more copies of a string is always possible and one does not bother making this explicit in the definition. This of course also rules out the example of classical random bit strings we discussed before. Moreover, this strong definition, once established, is rather flexible to use when studying the properties and applications of pseudorandom states.

**Definition 2 (Pseudorandom Quantum States (PRS's)).** *Let  $\kappa$  be the security parameter. Let  $\mathcal{H}$  be a Hilbert space and  $\mathcal{K}$  the key space, both parameterized by  $\kappa$ . A keyed family of quantum states  $\{|\phi_k\rangle \in S(\mathcal{H})\}_{k \in \mathcal{K}}$  is **pseudorandom**, if the following two conditions hold:*

1. (**Efficient generation**). There is a polynomial-time quantum algorithm  $G$  that generates state  $|\phi_k\rangle$  on input  $k$ . That is, for all  $k \in \mathcal{K}$ ,  $G(k) = |\phi_k\rangle$ .
2. (**Pseudorandomness**). Any polynomially many copies of  $|\phi_k\rangle$  with the same random  $k \in \mathcal{K}$  is **computationally indistinguishable** from the same number of copies of a Haar random state. More precisely, for any efficient quantum algorithm  $\mathcal{A}$  and any  $m \in \text{poly}(\kappa)$ ,

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}(|\phi_k\rangle^{\otimes m}) = 1] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{A}(|\psi\rangle^{\otimes m}) = 1] \right| = \text{negl}(\kappa),$$

where  $\mu$  is the Haar measure on  $S(\mathcal{H})$ .

### 3.2 Constructions and Analysis

In this section, we give an efficient construction of pseudorandom states which we call random phase states. We will prove that this family of states satisfies our definition of PRS's. There are other interesting and simpler candidate constructions, but the family of random phase states is the easiest to analyze.

Let  $\text{PRF} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$  be a quantum-secure pseudorandom function with key space  $\mathcal{K}$ ,  $\mathcal{X} = \{0, 1, 2, \dots, N-1\}$  and  $N = 2^n$ .  $\mathcal{K}$  and  $N$  are implicitly functions of the security parameter  $\kappa$ . The family of pseudorandom states of  $n$  qubits is defined

$$|\phi_k\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathcal{X}} \omega_N^{\text{PRF}_k(x)} |x\rangle, \quad (8)$$

for  $k \in \mathcal{K}$  and  $\omega_N = \exp(2\pi i/N)$ .

**Theorem 1.** *For any QPRF  $\text{PRF} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$ , the family of states  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$  defined in Eq. (8) is a PRS.*

*Proof.* First, we prove that the state can be efficiently prepared with a single query to  $\text{PRF}_k$ . As  $\text{PRF}_k$  is efficient, this proves the efficient generation property.

The state generation algorithm works as follows. First, it prepares a state

$$\frac{1}{N} \sum_{x \in \mathcal{X}} |x\rangle \sum_{y \in \mathcal{X}} \omega_N^y |y\rangle.$$

This can be done by applying  $H^{\otimes n}$  to the first register initialized in  $|0\rangle$  and the quantum Fourier transform to the second register in state  $|1\rangle$ .

Then the algorithm calls  $\text{PRF}_k$  on the first register and subtract the result from the second register, giving state

$$\frac{1}{N} \sum_{x \in \mathcal{X}} |x\rangle \sum_{y \in \mathcal{X}} \omega_N^y |y - \text{PRF}_k(x)\rangle.$$

The state can be rewritten as

$$\frac{1}{N} \sum_{x \in \mathcal{X}} \omega_N^{\text{PRF}_k(x)} |x\rangle \sum_{y \in \mathcal{X}} \omega_N^y |y\rangle.$$

Therefore, the effect of this step is to transform the first register to the required form and leaving the second register intact.

Next, we prove the pseudorandomness property of the family. For this purpose, we consider three hybrids. In the first hybrid  $H_1$ , the state will be  $|\phi_k\rangle^{\otimes m}$  for a uniform random  $k \in \mathcal{K}$ . In the second hybrid  $H_2$ , the state is  $|f\rangle^{\otimes m}$  for truly random functions  $f \in \mathcal{X}^{\mathcal{X}}$  where

$$|f\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathcal{X}} \omega_N^{f(x)} |x\rangle.$$

In the third hybrid  $H_3$ , the state is  $|\psi\rangle^{\otimes m}$  for  $|\psi\rangle$  chosen according to the Haar measure.

By the definition of the quantum-secure pseudorandom functions for PRF, we have for any polynomial-time quantum algorithm  $\mathcal{A}$  and any  $m \in \text{poly}(\kappa)$ ,

$$|\Pr[\mathcal{A}(H_1) = 1] - \Pr[\mathcal{A}(H_2) = 1]| = \text{negl}(\kappa).$$

By Lemma 1, we have for any algorithm  $\mathcal{A}$  and  $m \in \text{poly}(\kappa)$ ,

$$|\Pr[\mathcal{A}(H_2) = 1] - \Pr[\mathcal{A}(H_3) = 1]| = \text{negl}(\kappa).$$

This completes the proof by triangle inequality.

**Lemma 1.** *For function  $f : \mathcal{X} \rightarrow \mathcal{X}$ , define quantum state*

$$|f\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathcal{X}} \omega_N^{f(x)} |x\rangle.$$

*For  $m \in \text{poly}(\kappa)$ , the state ensemble  $\{|f\rangle^{\otimes m}\}$  is statistically indistinguishable from  $\{|\psi\rangle^{\otimes m}\}$  for Haar random  $|\psi\rangle$ .*

*Proof.* Let  $m \in \text{poly}(\kappa)$  be the number of copies of the state. We have

$$\mathbb{E}_f \left[ (|f\rangle\langle f|)^{\otimes m} \right] = \frac{1}{N^m} \sum_{\mathbf{x} \in \mathcal{X}^m, \mathbf{y} \in \mathcal{X}^m} \mathbb{E}_f \omega_N^{f(x_1) + \dots + f(x_m) - [f(y_1) + \dots + f(y_m)]} |\mathbf{x}\rangle\langle \mathbf{y}|,$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_m)$ . For later convenience, define density matrix

$$\rho^m = \mathbb{E}_f \left[ (|f\rangle\langle f|)^{\otimes m} \right].$$

We will compute the entries of  $\rho^m$  explicitly.

For  $\mathbf{x} = (x_1, x_2, \dots, x_m) \in \mathcal{X}^m$ , let  $m_j$  be the number of  $j$  in  $\mathbf{x}$  for  $j \in \mathcal{X}$ . Obviously, one has  $\sum_{j \in \mathcal{X}} m_j = m$ . Note that we have omitted the dependence of  $m_j$  on  $\mathbf{x}$  for simplicity. Recall the basis states defined in Eq. (4)

$$|\mathbf{x}; \text{Sym}\rangle = \frac{1}{\sqrt{\left(\prod_{j \in \mathcal{X}} m_j!\right) m!}} \sum_{\sigma \in S_m} |x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(m)}\rangle.$$

For  $\mathbf{x}, \mathbf{y} \in \mathcal{X}^m$ , let  $m_j$  be the number of  $j$  in  $\mathbf{x}$  and  $m'_j$  be the number of  $j$  in  $\mathbf{y}$ .

We can compute the entries of  $\rho^m$  as

$$\begin{aligned} & \langle \mathbf{x}; \text{Sym} | \rho^m | \mathbf{y}; \text{Sym} \rangle \\ &= \frac{m!}{N^m \sqrt{\left(\prod_{j \in \mathcal{X}} m_j!\right) \left(\prod_{j \in \mathcal{X}} m'_j!\right)}} \mathbb{E}_f \left[ \exp \left( \frac{2\pi i}{N} \sum_{l=1}^m (f(x_l) - f(y_l)) \right) \right]. \end{aligned}$$

When  $\mathbf{x}$  is not a permutation of  $\mathbf{y}$ , the summation  $\sum_{l=1}^m (f(x_l) - f(y_l))$  is a summation of terms  $\pm f(z_j)$  for distinct values  $z_j$ . As  $f$  is a truly random function,  $f(z_j)$  is uniformly random and independent of  $f(z_{j'})$  for  $z_j \neq z_{j'}$ . So it is not hard to verify that the entry is nonzero only if  $\mathbf{x}$  is a permutation of  $\mathbf{y}$ . These nonzero entries are on the diagonal of  $\rho^m$  in the basis of  $\{|\mathbf{x}; \text{Sym}\rangle\}$ . These diagonal entries are

$$\langle \mathbf{x}; \text{Sym} | \rho^m | \mathbf{x}; \text{Sym} \rangle = \frac{m!}{N^m \prod_{j \in \mathcal{X}} m_j!}.$$

Let  $\rho_\mu^m$  be the density matrix of a random state  $|\psi\rangle^{\otimes m}$ , for  $|\psi\rangle$  chosen from the Haar measure  $\mu$ . From Eqs. (5) and (7), we have that

$$\rho_\mu^m = \binom{N+m-1}{m}^{-1} \sum_{\mathbf{x}; \text{Sym}} |\mathbf{x}; \text{Sym}\rangle \langle \mathbf{x}; \text{Sym}|.$$

We need to prove

$$\text{TD}(\rho^m, \rho_\mu^m) = \text{negl}(\kappa).$$

Define

$$\delta_{\mathbf{x}; \text{Sym}} = \frac{m!}{N^m \prod_{j \in \mathcal{X}} m_j!} - \binom{N+m-1}{m}^{-1}.$$

Then

$$\text{TD}(\rho^m, \rho_\mu^m) = \frac{1}{2} \sum_{\mathbf{x}; \text{Sym}} |\delta_{\mathbf{x}; \text{Sym}}|.$$

The ratio of the two terms in  $\delta_{\mathbf{x}; \text{Sym}}$  is

$$\frac{m! \binom{N+m-1}{m}}{N^m \prod_{j \in \mathcal{X}} m_j!} = \frac{\prod_{l=0}^{m-1} \left(1 + \frac{l}{N}\right)}{\prod_{j \in \mathcal{X}} m_j!}.$$

For sufficient large security parameter  $\kappa$ , the ratio is larger than 1 only if  $\prod_{j \in \mathcal{X}} m_j! = 1$ , which corresponds to  $\mathbf{x}$ 's whose entries are all distinct. As there are  $\binom{N}{m}$  such  $\mathbf{x}$ 's, we can calculate the trace distance as

$$\begin{aligned} \text{TD}(\rho^m, \rho_\mu^m) &= \binom{N}{m} \left[ \frac{m!}{N^m} - \binom{N+m-1}{m}^{-1} \right] \\ &= \frac{N(N-1) \cdots (N-m+1)}{N^m} - \frac{N(N-1) \cdots (N-m+1)}{(N+m-1) \cdots N}. \end{aligned}$$

As first term is less than 1 and is at least

$$(1 - \frac{1}{N}) \cdots (1 - \frac{m-1}{N}) \geq 1 - \frac{1+2+\cdots+(m-1)}{N}$$

For our choices of  $m \in \text{poly}(\kappa)$  and  $N \in 2^{\text{poly}(\kappa)}$ , this term is  $1 - \text{negl}(\kappa)$  for sufficiently large security parameter  $\kappa$ . Similar analysis applies to the second term and this completes the proof.

### 3.3 Comparison with Related Work

We remark that a similar family of states was considered in [1] (Theorem 3). However, the size of the state family there depends on a parameter  $d$  which should be larger than the sum of the number of state copies and the number of queries. In our construction, the key space is fixed for a given security parameter, which may be advantageous for various applications.

We mention several other candidate constructions of PRS's and leave detailed analysis of them to future work. A construction closely related to the random phase states in Eq. (8) uses random  $\pm 1$  phases,

$$|\phi_k\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathcal{X}} (-1)^{\text{PRF}_k(x)} |x\rangle.$$

Intuitively, this family is less random than the random phase states in Eq. (8) and the corresponding density matrix  $\rho^m$  has small off-diagonal entries, making the proof more challenging. The other family of candidate states on  $2n$  qubits takes the form

$$|\phi_k\rangle = \frac{1}{\sqrt{N}} \text{PRP}_k \left[ \sum_{x \in \mathcal{X}} |x\rangle \otimes |0^n\rangle \right].$$

In this construction, the state is an equal superposition of a random subset of size  $2^n$  of  $\{0, 1\}^{2n}$  and PRP is any pseudorandom permutation over the set  $\{0, 1\}^{2n}$ . We call this the *random subset states* construction.

Finally, we remark that under plausible cryptographic assumptions our PRS constructions can be implemented using shallow quantum circuits of polylogarithmic depth. To see this, note that there exist PRFs that can be computed in polylogarithmic depth [6], which are based on lattice problems such as “learning with errors” (LWE) [52], and are believed to be secure against quantum computers. These PRFs can be used directly in our PRS construction. (Alternatively, one can use low-depth PRFs that are constructed from more general assumptions, such as the existence of trapdoor one-way permutations [46].)

This shows that PRS states can be prepared in surprisingly small depth, compared to quantum state  $t$ -designs, which generally require at least linear depth when  $t$  is a constant greater than 2, or polynomial depth when  $t$  grows polynomially with the number of qubits [4, 12, 40, 43]. (Note, however, that for  $t = 2$ , quantum state 2-designs can be generated in logarithmic depth [16].) Moreover, PRS states are sufficient for many applications where high-order  $t$ -designs are used [35, 51], provided that one only requires states to be *computationally* (not statistically) indistinguishable from Haar-random.



## 4 Cryptographic No-cloning Theorem and Quantum Money

A fundamental fact in quantum information theory is that unknown or random quantum states cannot be cloned [18, 48, 50, 60, 62]. The main topic of this section is to investigate the cloning problem for pseudorandom states. As we will see, even though pseudorandom states can be efficiently generated, they do share the no-cloning property of generic quantum states.

Let  $\mathcal{H}$  be the Hilbert space of dimension  $N$  and  $m < m'$  be two integers. The numbers  $N, m, m'$  depend implicitly on a security parameter  $\kappa$ . We will assume that  $N$  is exponential in  $\kappa$  and  $m \in \text{poly}(\kappa)$  in the following discussion.

We first recall the fact that for Haar random state  $|\psi\rangle \in \mathcal{S}(\mathcal{H})$ , the success probability of producing  $m'$  copies of the state given  $m$  copies is negligibly small. Let  $\mathcal{C}$  be a cloning channel that on input  $(|\psi\rangle\langle\psi|)^{\otimes m}$  tries to output a state that is close to  $(|\psi\rangle\langle\psi|)^{\otimes m'}$  for  $m' > m$ . The expected success probability of  $\mathcal{C}$  is measured by

$$\int \left\langle (|\psi\rangle\langle\psi|)^{\otimes m'}, \mathcal{C}((|\psi\rangle\langle\psi|)^{\otimes m}) \right\rangle d\mu(\psi).$$

It is known that [60], for all cloning channel  $\mathcal{C}$ , this success probability is bounded by

$$\binom{N+m-1}{m} \bigg/ \binom{N+m'-1}{m'},$$

which is  $\text{negl}(\kappa)$  for our choices of  $N, m, m'$ .

We establish a no-cloning theorem for PRS's which says that no efficient quantum cloning procedure exists for a general PRS. The theorem is called the cryptographic no-cloning theorem because of its deep roots in pseudorandomness in cryptography.

**Theorem 2 (Cryptographic No-cloning Theorem).** *For any PRS family  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$ ,  $m \in \text{poly}(\kappa)$ ,  $m < m'$  and any polynomial-time quantum algorithm  $\mathcal{C}$ , the success cloning probability*

$$\mathbb{E}_{k \in \mathcal{K}} \left\langle (|\phi_k\rangle\langle\phi_k|)^{\otimes m'}, \mathcal{C}((|\phi_k\rangle\langle\phi_k|)^{\otimes m}) \right\rangle = \text{negl}(\kappa).$$

*Proof.* Assume on the contrary that there is a polynomial-time quantum cloning algorithm  $\mathcal{C}$  such that the success cloning probability of producing  $m+1$  from  $m$  copies is  $\kappa^{-c}$  for some constant  $c > 0$ . We will construct a polynomial-time distinguisher  $\mathcal{D}$  that violates the definition of PRS's. Distinguisher  $\mathcal{D}$  will draw  $2m+1$  copies of the state, call  $\mathcal{C}$  on the first  $m$  copies, and perform the SWAP test on the output of  $\mathcal{C}$  and the remaining  $m+1$  copies. It is easy to see that  $\mathcal{D}$  outputs 1 with probability  $(1 + \kappa^{-c})/2$  if the input is from PRS, while if the input is Haar random, it outputs 1 with probability  $(1 + \text{negl}(\kappa))/2$ . Since  $\mathcal{C}$  is polynomial-time, it follows that  $\mathcal{D}$  is also polynomial-time. This is a contradiction with the definition of PRS's and completes the proof.

#### 4.1 A Strong Notion of PRS and Equivalence to PRS

In this section, we show that, somewhat surprisingly, PRS in fact implies a seemingly stronger notion, where indistinguishability needs to hold even if a distinguisher additionally has access to an oracle that reflects about the given state. There are at least a couple of motivations to consider an augmented notion. Firstly, unlike a classical string, a quantum state is inherently *hidden*. Give a quantum register prepared in some state (i.e., a physical system), we can only choose some observable to measure which just reveals partial information and will collapse the state in general. Therefore, it is meaningful to consider offering a distinguishing algorithm more information *describing* the given state, and the reflection oracle comes naturally. Secondly, this stronger notion is extremely useful in our application of quantum money schemes, and could be interesting elsewhere too.

More formally, for any state  $|\phi\rangle \in \mathcal{H}$ , define an oracle  $O_\phi := \mathbb{1} - 2|\phi\rangle\langle\phi|$  that reflects about  $|\phi\rangle$ .

**Definition 3 (Strongly Pseudorandom Quantum States).** *Let  $\mathcal{H}$  be a Hilbert space and  $\mathcal{K}$  be the key space.  $\mathcal{H}$  and  $\mathcal{K}$  depend on the security parameter  $\kappa$ . A keyed family of quantum states  $\{|\phi_k\rangle \in \mathcal{S}(\mathcal{H})\}_{k \in \mathcal{K}}$  is **strongly pseudo-random**, if the following two conditions hold:*

1. (**Efficient generation**). *There is a polynomial-time quantum algorithm  $G$  that generates state  $|\phi_k\rangle$  on input  $k$ . That is, for all  $k \in \mathcal{K}$ ,  $G(k) = |\phi_k\rangle$ .*
2. (**Strong Pseudorandomness**). *Any polynomially many copies of  $|\phi_k\rangle$  with the same random  $k \in \mathcal{K}$  is **computationally indistinguishable** from the same number of copies of a Haar random state. More precisely, for any efficient quantum oracle algorithm  $\mathcal{A}$  and any  $m \in \text{poly}(\kappa)$ ,*

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{O_{\phi_k}}(|\phi_k\rangle^{\otimes m}) = 1] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{A}^{O_\psi}(|\psi\rangle^{\otimes m}) = 1] \right| = \text{negl}(\kappa),$$

where  $\mu$  is the Haar measure on  $\mathcal{S}(\mathcal{H})$ .

Note that since the distinguisher  $\mathcal{A}$  is polynomial-time, the number of queries to the reflection oracle ( $O_{\phi_k}$  or  $O_\psi$ ) is also polynomially bounded.

We prove the advantage that a reflection oracle may give to a distinguisher is limited. In fact, standard PRS implies strong PRS, and hence they are equivalent.

**Theorem 3.** *A family of states  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$  is strongly pseudorandom if and only if it is (standard) pseudorandom.*

*Proof.* Clearly a strong PRS is also a standard PRS by definition. It suffice to prove that any PRS is also strongly pseudorandom.

Suppose for contradiction that there is a distinguishing algorithm  $\mathcal{A}$  that breaks the strongly pseudorandom condition. Namely, there exists  $m \in \text{poly}(\kappa)$  and constant  $c > 0$  such that for sufficiently large  $\kappa$ ,

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{O_{\phi_k}}(|\phi_k\rangle^{\otimes m}) = 1] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{A}^{O_\psi}(|\psi\rangle^{\otimes m}) = 1] \right| = \varepsilon(\kappa) \geq \kappa^{-c}.$$

We assume  $\mathcal{A}$  makes  $q \in \text{poly}(\kappa)$  queries to the reflection oracle. Then, by Theorem 4, there is an algorithm  $\mathcal{B}$  such that for any  $l$

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{O_{\phi_k}}(|\phi_k\rangle^{\otimes m})] - \Pr_{k \leftarrow \mathcal{K}} [\mathcal{B}(|\phi_k\rangle^{\otimes(m+l)})] \right| \leq \frac{2q}{\sqrt{l+1}},$$

and

$$\left| \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{A}^{O_\psi}(|\psi\rangle^{\otimes m})] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{B}(|\psi\rangle^{\otimes(m+l)})] \right| \leq \frac{2q}{\sqrt{l+1}}.$$

By triangle inequality, we have

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{B}(|\phi_k\rangle^{\otimes(m+l)})] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{B}(|\psi\rangle^{\otimes(m+l)})] \right| \geq \kappa^{-c} - \frac{4q}{\sqrt{l+1}}.$$

Choosing  $l = 64q^2\kappa^{2c} \in \text{poly}(\kappa)$ , we have

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{B}(|\phi_k\rangle^{\otimes(m+l)})] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{B}(|\psi\rangle^{\otimes(m+l)})] \right| \geq \kappa^{-c}/2,$$

which is a contradiction with the definition of PRS for  $\{|\phi_k\rangle\}$ . Therefore, we conclude that PRS and strong PRS are equivalent.

We now show a technical ingredient that allows us to simulate the reflection oracle about a state by using multiple copies of the given state. This result is inspired by a similar theorem proved by Ambainis et al. [5, Lemma 42]. Our simulation applies the reflection about the standard symmetric subspace, as opposed to a reflection operation about a particular subspace in [5], on the multiple copies of the given state, which we know how to implement efficiently.

**Theorem 4.** *Let  $|\psi\rangle \in \mathcal{H}$  be a quantum state. Define oracle  $O_\psi = \mathbb{1} - 2|\psi\rangle\langle\psi|$  to be the reflection about  $|\psi\rangle$ . Let  $|\xi\rangle$  be a state not necessarily independent of  $|\psi\rangle$ . Let  $\mathcal{A}^{O_\psi}$  be an oracle algorithm that makes  $q$  queries to  $O_\psi$ . For any integer  $l > 0$ , there is a quantum algorithm  $\mathcal{B}$  that makes no queries to  $O_\psi$  such that*

$$\text{TD}(\mathcal{A}^{O_\psi}(|\xi\rangle), \mathcal{B}(|\psi\rangle^{\otimes l} \otimes |\xi\rangle)) \leq \frac{q\sqrt{2}}{\sqrt{l+1}}.$$

Moreover, the running time of  $\mathcal{B}$  is polynomial in that of  $\mathcal{A}$  and  $l$ .

*Proof.* Consider a quantum register  $\mathsf{T}$ , initialized in the state  $|\Theta\rangle_{\mathsf{T}} = |\psi\rangle^{\otimes l} \in \mathcal{H}^{\otimes l}$ . Let  $\Pi$  be the projection onto the symmetric subspace  $\vee^{l+1}\mathcal{H} \subset \mathcal{H}^{\otimes(l+1)}$ , and let  $R = \mathbb{1} - 2\Pi$  be the reflection about the symmetric subspace.

Assume without loss of generality that algorithm  $\mathcal{A}$  is unitary and only performs measurements at the end. We define algorithm  $\mathcal{B}$  to be the same as  $\mathcal{A}$ , except that when  $\mathcal{A}$  queries  $O_\psi$  on register  $\mathsf{D}$ ,  $\mathcal{B}$  applies the reflection  $R$  on the collection of quantum registers  $\mathsf{D}$  and  $\mathsf{T}$ . We first analyze the corresponding states after the first oracle call to  $O_\psi$  in algorithms  $\mathcal{A}$  and  $\mathcal{B}$ ,

$$|\Psi_A\rangle = O_\psi(|\phi\rangle_{\mathsf{D}}) \otimes |\Theta\rangle_{\mathsf{T}}, \quad |\Psi_B\rangle = R(|\phi\rangle_{\mathsf{D}} \otimes |\Theta\rangle_{\mathsf{T}}).$$

For any two states  $|x\rangle, |y\rangle \in \mathcal{H}$ , we have

$$\begin{aligned}
(\langle x| \otimes \langle \Theta|) R(|y\rangle \otimes |\Theta\rangle) &= \langle x|y\rangle - 2 \mathbb{E}_{\pi \in S_{l+1}} (\langle x| \otimes \langle \Theta|) W_\pi (|y\rangle \otimes |\Theta\rangle) \\
&= \langle x|y\rangle - \frac{2}{l+1} \langle x|y\rangle - \frac{2l}{l+1} \langle x|\psi\rangle \langle \psi|y\rangle \\
&= \frac{l-1}{l+1} \langle x|y\rangle - \frac{2l}{l+1} \langle x|\psi\rangle \langle \psi|y\rangle,
\end{aligned}$$

where the first step uses the identity in Eq. (6) and the second step follows by observing that the probability of a random  $\pi \in S_{l+1}$  mapping 1 to 1 is  $1/(l+1)$ . These calculations imply that,

$$(\mathbb{1} \otimes \langle \Theta|) R(\mathbb{1} \otimes |\Theta\rangle) = \frac{l-1}{l+1} \mathbb{1} - \frac{2l}{l+1} |\psi\rangle \langle \psi|.$$

We can compute the inner product of the two states  $|\Psi_A\rangle$  and  $|\Psi_B\rangle$  as

$$\begin{aligned}
\langle \Psi_A | \Psi_B \rangle &= \text{tr} \left( (|\phi\rangle \otimes |\Theta\rangle) (\langle \phi| \otimes \langle \Theta|) (O_\psi \otimes \mathbb{1}) R \right) \\
&= \text{tr} \left( |\phi\rangle \langle \phi| O_\psi (\mathbb{1} \otimes \langle \Theta|) R (\mathbb{1} \otimes |\Theta\rangle) \right) \\
&= \text{tr} \left( |\phi\rangle \langle \phi| (\mathbb{1} - 2|\psi\rangle \langle \psi|) \left( \frac{l-1}{l+1} \mathbb{1} - \frac{2l}{l+1} |\psi\rangle \langle \psi| \right) \right) \\
&= \frac{l-1}{l+1} + \frac{2l}{l+1} |\langle \phi|\psi\rangle|^2 - \frac{2(l-1)}{l+1} |\langle \phi|\psi\rangle|^2 \\
&= \frac{l-1}{l+1} + \frac{2}{l+1} |\langle \phi|\psi\rangle|^2 \\
&\geq 1 - \frac{2}{l+1}.
\end{aligned}$$

This implies that

$$\| |\Psi_A\rangle - |\Psi_B\rangle \| \leq \frac{2}{\sqrt{l+1}}.$$

Let  $|\Psi_A^q\rangle$  and  $|\Psi_B^q\rangle$  be the final states of algorithm  $\mathcal{A}$  and  $\mathcal{B}$  before measurement respectively. Then by induction on the number of queries, we have

$$\| |\Psi_A^q\rangle - |\Psi_B^q\rangle \| \leq \frac{2q}{\sqrt{l+1}}.$$

This concludes the proof by noticing that

$$\text{TD}(|\Psi_A^q\rangle, |\Psi_B^q\rangle) \leq \| |\Psi_A^q\rangle - |\Psi_B^q\rangle \|.$$

Finally, we show that if  $\mathcal{A}$  is polynomial-time, then so is  $\mathcal{B}$ . Based on the construction of  $\mathcal{B}$ , it suffices to show that the reflection  $R$  is efficiently implementable for any polynomially large  $l$ . Here we use a result by Barenco et al. [8] which provides an efficient implementation for the projection  $\Pi$  onto  $\vee^{l+1} \mathcal{H}$ . More precisely, they design a quantum circuit of size  $O(\text{poly}(l, \log \dim \mathcal{H}))$  that

implements a unitary  $U$  such that  $U|\phi\rangle = \sum_j |\xi_j\rangle|j\rangle$  on  $\mathcal{H}^{\otimes(l+1)} \otimes \mathcal{H}'$  for an auxiliary space  $\mathcal{H}'$  of dimension  $O(l!)$ . Here  $|\xi_0\rangle = \Pi|\phi\rangle$  corresponds to the projection of  $|\phi\rangle$  on the symmetric subspace. With  $U$ , we can implement the reflection  $R$  as  $U^*SU$  where  $S$  is the unitary that introduces a minus sign conditioned on the second register being 0.

$$S|\Psi\rangle|j\rangle = \begin{cases} -|\Psi\rangle|j\rangle & \text{if } j = 0, \\ |\Psi\rangle|j\rangle & \text{otherwise.} \end{cases}$$

## 4.2 Quantum Money from PRS

Using Theorem 3, we can improve Theorem 2 to the following version. The proof is omitted as it is very similar to that for Theorem 2 and uses the complexity-theoretic no-cloning theorem [1, 2] for Haar random states.

**Theorem 5 (Cryptographic no-cloning Theorem with Oracle).** *For any PRS  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$ ,  $m \in \text{poly}(\kappa)$ ,  $m < m'$  and any polynomial-time quantum query algorithm  $\mathcal{C}$ , the success cloning probability*

$$\mathbb{E}_{k \in \mathcal{K}} \left\langle (|\phi_k\rangle\langle\phi_k|)^{\otimes m'}, \mathcal{C}^{O_{\phi_k}}((|\phi_k\rangle\langle\phi_k|)^{\otimes m}) \right\rangle = \text{negl}(\kappa).$$

A direct application of this no-cloning theorem is that it gives rise to new constructions for private-key quantum money. As one of the earliest findings in quantum information [9, 61], quantum money schemes have received revived interests in the past decade (see e.g. [1, 3, 20, 21, 39, 42]). First, we recall the definition of quantum money scheme adapted from [2].

**Definition 4 (Quantum Money Scheme).** *A private-key quantum money scheme  $\mathcal{S}$  consists of three algorithms:*

- *KeyGen*, which takes as input the security parameter  $1^\kappa$  and randomly samples a private key  $k$ .
- *Bank*, which takes as input the private key  $k$  and generates a quantum state  $|\$ \rangle$  called a **banknote**.
- *Ver*, which takes as input the private key  $k$  and an alleged banknote  $|\zeta\rangle$ , and either accepts or rejects.

*The money scheme  $\mathcal{S}$  has **completeness error**  $\varepsilon$  if  $\text{Ver}(k, |\$ \rangle)$  accepts with probability at least  $1 - \varepsilon$  for all valid banknote  $|\$ \rangle$ .*

*Let **Count** be the money counter that output the number of valid banknotes when given a collection of (possibly entangled) alleged banknotes  $|\zeta_1, \zeta_2, \dots, \zeta_r\rangle$ . Namely, **Count** will call **Ver** on each banknotes and return the number of times that **Ver** accepts. The money scheme  $\mathcal{S}$  has **soundness error**  $\delta$  if for any polynomial-time counterfeiter  $C$  that maps  $q$  valid banknotes  $|\$ _1\rangle, \dots, |\$ _q\rangle$  to  $r$  alleged banknotes  $|\zeta_1, \dots, \zeta_r\rangle$  satisfies*

$$\Pr[\text{Count}(k, C(|\$ _1\rangle, \dots, |\$ _q\rangle)) > q] \leq \delta.$$

*The scheme  $\mathcal{S}$  is **secure** if it has completeness error  $\leq 1/3$  and negligible soundness error.*

For any  $\text{PRS} = \{|\phi_k\rangle\}_{k \in \mathcal{K}}$  with key space  $\mathcal{K}$ , we can define a private-key quantum money scheme  $\mathcal{S}_{\text{PRS}}$  as follows:

- $\text{KeyGen}(1^\kappa)$  randomly outputs  $k \in \mathcal{K}$ .
- $\text{Bank}(k)$  generates the banknote  $|\$ \rangle = |\phi_k\rangle$ .
- $\text{Ver}(k, \rho)$  applies the projective measurement that accepts  $\rho$  with probability  $\langle \phi_k | \rho | \phi_k \rangle$ .

We remark that usually the money state  $|\$ \rangle$  takes the form  $|\$ \rangle = |s, \psi_s\rangle$  where the first register contains a classical serial number. Our scheme, however, does not require the use of the serial numbers. This simplification is brought to us by the strong requirement that any polynomial copies of  $|\phi_k\rangle$  are indistinguishable from Haar random states.

**Theorem 6.** *The private-key quantum money scheme  $\mathcal{S}_{\text{PRS}}$  is secure for all PRS.*

*Proof.* It suffices to prove the soundness of  $\mathcal{S}_{\text{PRS}}$  is negligible. Assume to the contrary that there is a counterfeiter  $C$  such that

$$\Pr[\text{Count}(k, C(|\phi_k\rangle^{\otimes q})) > q] \geq \kappa^{-c}$$

for some constant  $c > 0$  and sufficiently large  $\kappa$ . From the counterfeiter  $C$ , we will construct an oracle algorithm  $\mathcal{A}^{O_{\phi_k}}$  that maps  $q$  copies of  $|\phi_k\rangle$  to  $q + 1$  copies with noticeable probability and this leads to a contradiction with Theorem 5.

The oracle algorithm  $\mathcal{A}$  first runs  $C$  and implement the measurement

$$\left\{ M^0 = \mathbb{1} - |\phi_k\rangle\langle\phi_k|, M^1 = |\phi_k\rangle\langle\phi_k| \right\}$$

on each copy of the money state  $C$  outputs. This measurement can be implemented by attaching an auxiliary qubit initialized in  $(|0\rangle + |1\rangle)/\sqrt{2}$  and call the reflection oracle  $O_\phi$  conditioned on the qubit being at 1 and performs the  $X$  measurement on this auxiliary qubit. This gives  $r$ -bit of outcome  $\mathbf{x} \in \{0, 1\}^r$ . If  $\mathbf{x}$  has Hamming weight at least  $q + 1$ , algorithm  $\mathcal{A}$  outputs any  $q + 1$  registers that corresponds to outcome 1; otherwise, it outputs  $|0\rangle^{\otimes(q+1)}$ . By the construction of  $\mathcal{A}$ , it succeeds in cloning  $q + 1$  money states from  $q$  copies with probability at least  $\kappa^{-c}$ .

Our security proof of the quantum money scheme is arguably simpler than that in [2]. In [2], to prove their hidden subspace money scheme is secure, one needs to develop the so called inner-product adversary method to show the worst-case query complexity for the hidden subspace states and use a random self-reducible argument to establish the average-case query complexity. In our case, it follows almost directly from the cryptographic no-cloning theorem with oracles. The quantum money schemes derived from PRS's enjoy many nice features of the hidden subspace scheme. Most importantly, they are also *query-secure* [2], meaning that the bank can simply return the money state back to the user after verification.

It is also interesting to point out that quantum money states are not necessarily pseudorandom states. The hidden subspace state [2], for example, do not satisfy our definition of PRS as one can measure polynomially many copies of the state in the computational basis and recover a basis for the hidden subspace with high probability.

## 5 Entanglement of Pseudorandom Quantum States

In this section, we study the entanglement property of pseudorandom quantum states. Our result shows that any PRS consists of states that have high entanglement on average.

The entanglement property of a bipartite pure quantum state is well understood and is completely determined by the Schmidt coefficients of a bipartite state (see e.g. [31]). Any state  $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$  on system  $A$  and  $B$  can be written as

$$|\psi\rangle = \sum_{j=1}^R \sqrt{\lambda_j} |\psi_A^j\rangle \otimes |\psi_B^j\rangle,$$

where  $\lambda_j > 0$  for all  $1 \leq j \leq R$  and the states  $|\psi_A^j\rangle$  (and  $|\psi_B^j\rangle$ ) form a set of orthonormal states on  $A$  (and  $B$  respectively). Here, the positive real numbers  $\lambda_j$ 's are the Schmidt coefficients and  $R$  is the Schmidt rank of state  $|\psi\rangle$ . Let  $\rho_A$  be the reduced density matrix of  $|\psi\rangle$  on system  $A$ , then  $\lambda_j$  is the nonzero eigenvalues of  $\rho_A$ . Entanglement can be measured by the Schmidt rank  $R$  or entropy-like quantities derived from the Schmidt coefficients. We consider the quantum  $\alpha$ -Rényi entropy of  $\rho_A$

$$S_\alpha(\rho_A) := \frac{1}{1-\alpha} \log \left( \sum_{j=1}^R \lambda_j^\alpha \right).$$

When  $\alpha \rightarrow 1$ ,  $S_\alpha$  coincides with the von Neumann entropy of  $\rho_A$

$$S(\rho_A) = - \sum_{j=1}^R \lambda_j \log \lambda_j.$$

When  $\alpha \rightarrow \infty$ ,  $S_\alpha$  coincides with the quantum min entropy of  $\rho_A$

$$S_{\min}(\rho_A) = -\log \|\rho_A\| = -\log \lambda_{\max},$$

where  $\lambda_{\max}$  is the largest eigenvalue of  $\rho_A$ . For  $\alpha = 2$ , the entropy  $S_2$  is the quantum analogue of the collision entropy.

For Haar random state  $|\psi\rangle \sim \mu(\mathcal{H}_A \otimes \mathcal{H}_B)$  where the dimensions of  $\mathcal{H}_A$  and  $\mathcal{H}_B$  are  $d_A$  and  $d_B$  respectively, the Page conjecture [49] proved in [22, 54, 55] states that for  $d_A \leq d_B$ , the average entanglement entropy is explicitly given as

$$\mathbb{E} S(\rho_A) = \frac{1}{\ln 2} \left[ \left( \sum_{j=d_B+1}^{d_A d_B} \frac{1}{j} \right) - \frac{d_B - 1}{2d_A} \right] > \log d_A - O(1).$$

That is, the Haar random states are highly entangled on average and, in fact, a typical Haar random state is almost maximumly entangled. A more detailed discussion on this phenomena is give in [28, 34]. The following theorem and its corollary tell us that pseudorandom states are also entangled on average though the quantitative bound is much weaker.

**Theorem 7.** *Let  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$  be a family of PRS with security parameter  $\kappa$ . Consider partitions of the state  $|\phi_k\rangle$  into systems  $A$  and  $B$  consisting of  $n_A$  and  $n_B$  qubits each where both  $n_A$  and  $n_B$  are polynomial in the security parameter. Let  $\rho_k$  be the reduced density matrix on system  $A$ . Then,*

$$\mathbb{E}_k \text{tr}(\rho_k^2) = \text{negl}(\kappa).$$

*Proof.* Assume to the contrary that

$$\mathbb{E}_k \text{tr}(\rho_k^2) \geq \kappa^{-c}$$

for some constant  $c > 0$  and sufficiently large  $\kappa$ . We will construct a distinguisher  $\mathcal{A}$  that tells the family of state  $\{|\phi_k\rangle\}$  apart from the Haar random states.

Consider the SWAP test performed on the system  $A$  of two copies of  $|\phi_k\rangle$ . The test accepts with probability

$$\frac{1 + \text{tr}(\rho_k^2)}{2}.$$

Let distinguisher  $\mathcal{A}$  be the above SWAP test, we have

$$\begin{aligned} & \left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}(|\phi_k\rangle^{\otimes 2}) = 1] - \Pr_{|\psi\rangle \leftarrow \mu} [\mathcal{A}(|\psi\rangle^{\otimes 2}) = 1] \right| \\ &= \frac{1}{2} \left| \mathbb{E}_k \text{tr}(\rho_k^2) - \mathbb{E}_\mu \text{tr}(\rho_\psi^2) \right| \geq \kappa^{-c}/4, \end{aligned}$$

for sufficiently large  $\kappa$ . The last step follows by a formula of Lubkin [36]

$$\mathbb{E}_{|\psi\rangle \leftarrow \mu} \text{tr}(\rho_\psi^2) = \frac{d_A + d_B}{d_A d_B + 1} = \frac{2^{n_A} + 2^{n_B}}{2^{n_A + n_B} + 1} = \text{negl}(\kappa).$$

**Corollary 1.** *Let  $\{|\phi_k\rangle\}_{k \in \mathcal{K}}$  be a family of PRS with security parameter  $\kappa$ . Consider partitions of the state  $|\phi_k\rangle$  into systems  $A$  and  $B$  consisting of  $n_A$  and  $n_B$  qubits each where both  $n_A$  and  $n_B$  are polynomial in the security parameter. We have*

1. *Let  $R_k$  be the Schmidt rank of state  $|\phi_k\rangle$  under the  $A, B$  partition, then  $\mathbb{E}_k R_k \geq \kappa^c$  for all constant  $c > 0$  and sufficiently large  $\kappa$ .*
2.  *$\mathbb{E}_k S_{\min}(\rho_k) = \omega(\log \kappa)$  and  $\mathbb{E}_k S(\rho_k) = \omega(\log \kappa)$ .*



*Proof.* The first item follows from the fact that

$$\mathrm{tr}(\rho_k^2) \geq \frac{1}{R_k}.$$

where  $R_k$  is the Schmidt rank of state  $|\phi_k\rangle$ . The second item for the min entropy follows by Jensen's inequality and

$$\mathrm{tr}(\rho_k^2) \geq \lambda_{\max}^2.$$

Finally, the bound on the expected entanglement entropy follows by the fact that min entropy is the smallest  $\alpha$ -Rényi entropy for all  $\alpha > 0$ .

## 6 Pseudorandom Unitary Operators (PRUs)

### 6.1 Definitions

Our notion of pseudorandom states readily extends to distributions over unitary operators. Let  $\mathcal{H}$  be a Hilbert space and let  $\mathcal{K}$  a key space, both of which depend on a security parameter  $\kappa$ . Let  $\mu$  be the Haar measure on the unitary group  $U(\mathcal{H})$ .

**Definition 5.** A family of unitary operators  $\{U_k \in U(\mathcal{H})\}_{k \in \mathcal{K}}$  is **pseudorandom**, if two conditions hold:

1. (**Efficient computation**). There is an efficient quantum algorithm  $Q$ , such that for all  $k$  and any  $|\psi\rangle \in S(\mathcal{H})$ ,  $Q(k, |\psi\rangle) = U_k|\psi\rangle$ .
2. (**Pseudorandomness**).  $U_k$  with a random key  $k$  is **computationally indistinguishable** from a Haar random unitary operator. More precisely, for any efficient quantum algorithm  $\mathcal{A}$  that makes at most polynomially many queries to the oracle,

$$\left| \Pr_{k \leftarrow \mathcal{K}} [\mathcal{A}^{U_k}(1^\kappa) = 1] - \Pr_{U \leftarrow \mu} [\mathcal{A}^U(1^\kappa) = 1] \right| = \mathrm{negl}(\kappa).$$

The extensive literature on approximation of Haar randomness on unitary groups concerns with unitary *designs* [12, 19], which are statistical approximations to the Haar random distribution up to a fixed  $t$ -th moment. Our notion of pseudorandom unitary operators in terms of computational indistinguishability, in addition to independent interest, supplements and could substitute for unitary designs in various applications.

### 6.2 Candidate Constructions

Clearly, given a pseudorandom unitary family  $\{U_k\}$ , it immediately gives pseudorandom states as well (e.g.,  $\{U_k|0\rangle\}$ ). On the other hand, our techniques for constructing pseudorandom states can be extended to give candidate constructions for pseudorandom unitary operators (PRUs) in the following way. Let

$\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ . Assume we have a pseudorandom function  $\text{PRF} : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$ , with domain  $\mathcal{X} = \{0, 1, 2, \dots, N-1\}$  and  $N = 2^n$ . Using the phase kick-back technique, we can implement the unitary transformation  $T_k \in \text{U}(\mathcal{H})$  that maps

$$T_k : |x\rangle \mapsto \omega_N^{\text{PRF}_k(x)} |x\rangle, \quad \omega_N = \exp(2\pi i/N). \quad (9)$$

Our pseudorandom states were given by  $|\phi_k\rangle = T_k H^{\otimes n} |0\rangle$ , where  $H^{\otimes n}$  denotes the  $n$ -qubit Hadamard transform. We conjecture that by repeating the operation  $T_k H^{\otimes n}$  a constant number of times (with different keys  $k$ ), we get a PRU. This resembles the construction of unitary  $t$ -designs in [43, 44].

Alternatively, one can give a candidate construction for PRUs based on pseudorandom permutations (PRPs) as follows. First, let  $\text{PRP}_k$  be a pseudorandom permutation (with key  $k \in \mathcal{K}$ ) acting on  $\{0, 1\}^n$ , and suppose we have efficient quantum circuits that compute the permutation  $P_k : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus \text{PRP}_k(x)\rangle$  as well as its inverse  $R_k : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus \text{PRP}_k^{-1}(x)\rangle$  (where  $\oplus$  denotes the bit-wise xor operation). Then we can compute the permutation in-place by applying the following sequence of operations:

$$\begin{aligned} |x\rangle|0\rangle &\xrightarrow{P_k} |x\rangle|\text{PRP}_k(x)\rangle \\ &\xrightarrow{\text{SWAP}} |\text{PRP}_k(x)\rangle|x\rangle \\ &\xrightarrow{R_k} |\text{PRP}_k(x)\rangle|0\rangle. \end{aligned} \quad (10)$$

For simplicity, let us denote this operation by  $S_k : |x\rangle \mapsto |\text{PRP}_k(x)\rangle$  (ignoring the second register, which stays in the state  $|0\rangle$ ). Now we can consider repeating the operation  $S_k H^{\otimes n}$  several times (with different keys  $k$ ), as a candidate for a PRU. Note that this resembles the construction of unitary  $t$ -designs in [26].

It is an interesting challenge to prove that these constructions actually yield PRUs. For the special case of non-adaptive adversaries, one could try to use the proof techniques of [26, 43, 44] for unitary  $t$ -designs. For the general case, where the adversary can make adaptive queries to the pseudorandom unitary, new proof techniques seem to be needed. Finally, we can consider combining all of these ingredients (the pseudorandom operations  $S_k$  and  $T_k$ , and the Hadamard transform) to try to obtain more efficient constructions of PRUs.

## References

1. Aaronson, S.: Quantum copy-protection and quantum money. In: Proceedings of the Twenty-Fourth Annual IEEE Conference on Computational Complexity (CCC 2009), pp. 229–242. IEEE Computer Society (2009). <https://doi.org/10.1109/CCC.2009.42>
2. Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. In: Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, STOC 2012, pp. 41–60. ACM, New York (2012). <https://doi.org/10.1145/2213977.2213983>

3. Aaronson, S., Farhi, E., Gosset, D., Hassidim, A., Kelner, J., Lutomirski, A.: Quantum money. *Commun. ACM* **55**(8), 84–92 (2012). <https://doi.org/10.1145/2240236.2240258>
4. Ambainis, A., Emerson, J.: Quantum  $t$ -designs:  $t$ -wise independence in the quantum world. In: *Proceedings of the Twenty-Second Annual IEEE Conference on Computational Complexity (CCC 2007)*, pp. 129–140, June 2007
5. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: the hardness of quantum rewinding. In: *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 474–483. IEEE Computer Society (2014). <https://doi.org/10.1109/FOCS.2014.57>. Full version at <https://arxiv.org/abs/1404.6898>
6. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) *EUROCRYPT 2012*. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-29011-4\\_42](https://doi.org/10.1007/978-3-642-29011-4_42)
7. Barak, B., Shaltiel, R., Wigderson, A.: Computational analogues of entropy. In: Arora, S., Jansen, K., Rolim, J.D.P., Sahai, A. (eds.) *APPROX/RANDOM-2003*. LNCS, vol. 2764, pp. 200–215. Springer, Heidelberg (2003). [https://doi.org/10.1007/978-3-540-45198-3\\_18](https://doi.org/10.1007/978-3-540-45198-3_18)
8. Barenco, A., Berthiaume, A., Deutsch, D., Ekert, A., Jozsa, R., Macchiavello, C.: Stabilization of quantum computations by symmetrization. *SIAM J. Comput.* **26**(5), 1541–1557 (1997). <https://doi.org/10.1137/S0097539796302452>
9. Bennett, C.H., Brassard, G., Breidbart, S., Wiesner, S.: Quantum cryptography, or unforgeable subway tokens. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) *Advances in Cryptology*, pp. 267–275. Springer, Boston, MA (1983). [https://doi.org/10.1007/978-1-4757-0602-4\\_26](https://doi.org/10.1007/978-1-4757-0602-4_26)
10. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Comput.* **13**(4), 850–864 (1984). <https://doi.org/10.1137/0213053>
11. Brandão, F.G.S.L., Harrow, A.W., Horodecki, M.: Efficient quantum pseudorandomness. *Phys. Rev. Lett.* **116**, 170502 (2016). <https://doi.org/10.1103/PhysRevLett.116.170502>
12. Brandão, F.G.S.L., Harrow, A.W., Horodecki, M.: Local random quantum circuits are approximate polynomial-designs. *Commun. Math. Phys.* **346**(2), 397–434 (2016). <https://doi.org/10.1007/s00220-016-2706-8>
13. Bremner, M.J., Mora, C., Winter, A.: Are random pure states useful for quantum computation? *Phys. Rev. Lett.* **102**, 190502 (2009). <https://doi.org/10.1103/PhysRevLett.102.190502>
14. Chen, Y.H., Chung, K.M., Lai, C.Y., Vadhan, S.P., Wu, X.: Computational notions of quantum min-entropy. [arXiv:1704.07309](https://arxiv.org/abs/1704.07309) (2017)
15. Chung, K.M., Shi, Y., Wu, X.: Physical randomness extractors: generating random numbers with minimal assumptions. *arXiv preprint* [arXiv:1402.4797](https://arxiv.org/abs/1402.4797) (2014)
16. Cleve, R., Leung, D., Liu, L., Wang, C.: Near-linear constructions of exact unitary 2-designs. *Quantum Inf. Comput.* **16**(9&10), 721–756 (2016). <http://www.rintonpress.com/xxqic16/qic-16-910/0721-0756.pdf>
17. Dankert, C., Cleve, R., Emerson, J., Livine, E.: Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A* **80**, 012304 (2009). <https://doi.org/10.1103/PhysRevA.80.012304>
18. Dieks, D.: Communication by EPR devices. *Phys. Lett. A* **92**(6), 271–272 (1982)

19. Emerson, J., Weinstein, Y.S., Saraceno, M., Lloyd, S., Cory, D.G.: Pseudo-random unitary operators for quantum information processing. *Science* **302**(5653), 2098–2100 (2003)
20. Farhi, E., Gosset, D., Hassidim, A., Lutomirski, A., Nagaj, D., Shor, P.: Quantum state restoration and single-copy tomography for ground states of hamiltonians. *Phys. Rev. Lett.* **105**, 190503 (2010). <https://doi.org/10.1103/PhysRevLett.105.190503>
21. Farhi, E., Gosset, D., Hassidim, A., Lutomirski, A., Shor, P.: Quantum money from knots. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS 2012*, pp. 276–289. ACM, New York (2012). <https://doi.org/10.1145/2090236.2090260>
22. Foong, S.K., Kanno, S.: Proof of Page’s conjecture on the average entropy of a subsystem. *Phys. Rev. Lett.* **72**, 1148–1151 (1994). <https://doi.org/10.1103/PhysRevLett.72.1148>
23. Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions (extended abstract). In: Blakley, G.R., Chaum, D. (eds.) *CRYPTO 1984*. LNCS, vol. 196, pp. 276–288. Springer, Heidelberg (1985). [https://doi.org/10.1007/3-540-39568-7\\_22](https://doi.org/10.1007/3-540-39568-7_22)
24. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. *J. ACM* **33**(4), 792–807 (1986). <https://doi.org/10.1145/6490.6503>
25. Harrow, A.W.: The church of the symmetric subspace. [arXiv:1308.6595](https://arxiv.org/abs/1308.6595) (2013)
26. Harrow, A.W., Low, R.A.: Efficient quantum tensor product expanders and  $k$ -designs. In: Dinur, I., Jansen, K., Naor, J., Rolim, J. (eds.) *APPROX/RANDOM-2009*. LNCS, vol. 5687, pp. 548–561. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-03685-9\\_41](https://doi.org/10.1007/978-3-642-03685-9_41)
27. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. *SIAM J. Comput.* **28**(4), 1364–1396 (1999)
28. Hayden, P., Leung, D.W., Winter, A.: Aspects of generic entanglement. *Commun. Math. Phys.* **265**(1), 95–117 (2006). <https://doi.org/10.1007/s00220-006-1535-6>
29. Helstrom, C.W.: Detection theory and quantum mechanics. *Inf. Control* **10**(3), 254–291 (1967)
30. Holevo, A.S.: An analogue of statistical decision theory and noncommutative probability theory. *Tr. Mosk. Matematicheskogo Obshchestva* **26**, 133–149 (1972)
31. Horodecki, R., Horodecki, P., Horodecki, M., Horodecki, K.: Quantum entanglement. *Rev. Mod. Phys.* **81**, 865–942 (2009). <https://doi.org/10.1103/RevModPhys.81.865>
32. Impagliazzo, R., Wigderson, A.:  $P = BPP$  if  $E$  requires exponential circuits: derandomizing the XOR lemma. In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, STOC 1997*, pp. 220–229. ACM, New York (1997). <https://doi.org/10.1145/258533.258590>
33. Kueng, R., Gross, D.: Qubit stabilizer states are complex projective 3-designs. [arXiv:1510.02767](https://arxiv.org/abs/1510.02767) (2015)
34. Liu, Z.W., Lloyd, S., Zhu, E.Y., Zhu, H.: Entropic scrambling complexities. [arXiv:1703.08104](https://arxiv.org/abs/1703.08104) (2017)
35. Low, R.A.: Large deviation bounds for  $k$ -designs. *Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **465**(2111), 3289–3308 (2009). <http://rspa.royalsocietypublishing.org/content/465/2111/3289>
36. Lubkin, E.: Entropy of an  $n$ -system from its correlation with a  $k$ -reservoir. *J. Math. Phys.* **19**(5), 1028–1031 (1978)
37. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.* **17**(2), 373–386 (1988)

38. Lutomirski, A.: An online attack against Wiesner's quantum money. [arXiv:1010.0256](#) (2010)
39. Lutomirski, A., Aaronson, S., Farhi, E., Gosset, D., Hassidim, A., Kelner, J., Shor, P.: Breaking and making quantum money: toward a new quantum cryptographic protocol. In: Proceedings of the Innovations in Theoretical Computer Science Conference, ITCS 2010, pp. 20–31. Tsinghua University Press (2010)
40. Mezher, R., Ghalbouni, J., Dgheim, J., Markham, D.: Efficient quantum pseudorandomness with simple graph states. [arXiv:1709.08091](#) (2017)
41. Miller, C.A., Shi, Y.: Robust protocols for securely expanding randomness and distributing keys using untrusted quantum devices. *J. ACM (JACM)* **63**(4), 33 (2016)
42. Mosca, M., Stebila, D.: Quantum coins. In: Bruen, A.A., Wehlau, D.L. (eds.) *Error-Correcting Codes, Finite Geometries and Cryptography. Contemporary Mathematics*, vol. 523, pp. 35–47. American Mathematical Society, Providence (2010). <http://www.ams.org/bookstore?fn=20&arg1=commseries&ikey=CONM-523>
43. Nakata, Y., Hirche, C., Koashi, M., Winter, A.: Efficient quantum pseudorandomness with nearly time-independent Hamiltonian dynamics. *Phys. Rev. X* **7**, 021006 (2017). <https://doi.org/10.1103/PhysRevX.7.021006>
44. Nakata, Y., Hirche, C., Morgan, C., Winter, A.: Unitary 2-designs from random X- and Z-diagonal unitaries. *J. Math. Phys.* **58**(5), 052203 (2017). <https://doi.org/10.1063/1.4983266>
45. Nakata, Y., Koashi, M., Murao, M.: Generating a state t-design by diagonal quantum circuits. *New J. Phys.* **16**(5), 053043 (2014). <http://stacks.iop.org/1367-2630/16/i=5/a=053043>
46. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.* **58**(2), 336–375 (1999). <https://doi.org/10.1006/jcss.1998.1618>
47. Nisan, N., Wigderson, A.: Hardness vs randomness. *J. Comput. Syst. Sci.* **49**(2), 149–167 (1994). [https://doi.org/10.1016/S0022-0000\(05\)80043-1](https://doi.org/10.1016/S0022-0000(05)80043-1)
48. Ortigoso, J.: Twelve years before the quantum no-cloning theorem. [arXiv:1707.06910](#) (2017)
49. Page, D.N.: Average entropy of a subsystem. *Phys. Rev. Lett.* **71**, 1291–1294 (1993). <https://doi.org/10.1103/PhysRevLett.71.1291>
50. Park, J.L.: The concept of transition in quantum mechanics. *Found. Phys.* **1**, 23–33 (1970)
51. Popescu, S., Short, A.J., Winter, A.: Entanglement and the foundations of statistical mechanics. *Nat. Phys.* **2**(11), 754 (2006)
52. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. *J. ACM (JACM)* **56**(6), 34 (2009)
53. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, pp. 387–394. ACM (1990)
54. Sánchez-Ruiz, J.: Simple proof of Page's conjecture on the average entropy of a subsystem. *Phys. Rev. E* **52**, 5653–5655 (1995). <https://doi.org/10.1103/PhysRevE.52.5653>
55. Sen, S.: Average entropy of a quantum subsystem. *Phys. Rev. Lett.* **77**, 1–3 (1996). <https://doi.org/10.1103/PhysRevLett.77.1>
56. Shamir, A.: On the generation of cryptographically strong pseudorandom sequences. *ACM Trans. Comput. Syst.* **1**(1), 38–44 (1983). <https://doi.org/10.1145/357353.357357>

57. Song, F.: Quantum-secure pseudorandom permutations, June 2017. Blog post. <http://qcc.fangsong.info/2017-06-quantumprp/>
58. Watrous, J.: The Theory of Quantum Information. Cambridge University Press, Cambridge (2018, to be published). A draft copy is available at <https://cs.uwaterloo.ca/~watrous/TQI/>
59. Webb, Z.: The Clifford group forms a unitary 3-design. Quantum Inf. Comput. **16**(15&16), 1379–1400 (2016). <http://www.rintonpress.com/xxqic16/qic-16-1516/1379-1400.pdf>
60. Werner, R.F.: Optimal cloning of pure states. Phys. Rev. A **58**, 1827–1832 (1998). <https://doi.org/10.1103/PhysRevA.58.1827>
61. Wiesner, S.: Conjugate coding. SIGACT News **15**(1), 78–88 (1983). Original manuscript written Circa 1970
62. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature **299**, 802–803 (1982)
63. Yao, A.C.: Theory and application of trapdoor functions. In: 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), pp. 80–91, November 1982
64. Yuen, H.: A quantum lower bound for distinguishing random functions from random permutations. Quantum Inf. Comput. **14**(13–14), 1089–1097 (2014). <http://dl.acm.org/citation.cfm?id=2685166>
65. Zhandry, M.: How to construct quantum random functions. In: FOCS 2012, pp. 679–687. IEEE (2012). <http://eprint.iacr.org/2012/182>
66. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Inf. Comput. **15**(7&8) (2015). <http://arxiv.org/abs/1312.1027>
67. Zhandry, M.: A note on quantum-secure PRPs (2016). <https://eprint.iacr.org/2016/1076>
68. Zhandry, M.: Quantum lightning never strikes the same state twice. iACR eprint 2017/1080 (2017)
69. Zhu, H.: Multiqubit Clifford groups are unitary 3-designs. [arXiv:1510.02619](https://arxiv.org/abs/1510.02619) (2015)

# Quantum Security of NMAC and Related Constructions

## PRF Domain Extension Against Quantum attacks

Fang Song<sup>1</sup>( ) and Aaram Yun<sup>2</sup>( )

<sup>1</sup> Portland State University, Portland, USA  
fang.song@pdx.edu

<sup>2</sup> Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea  
aaramyun@unist.ac.kr

**Abstract.** We prove the security of NMAC, HMAC, AMAC, and the cascade construction with fixed input-length as *quantum-secure* pseudo-random functions (PRFs). Namely, they are indistinguishable from a random oracle against any polynomial-time quantum adversary that can make quantum superposition queries. In contrast, many blockcipher-based PRFs including CBC-MAC were recently broken by quantum superposition attacks.

Classical proof strategies for these constructions do not generalize to the quantum setting, and we observe that they sometimes even fail completely (e.g., the universal-hash then PRF paradigm for proving security of NMAC). Instead, we propose a direct hybrid argument as a new proof strategy (both classically and quantumly). We first show that a quantum-secure PRF is secure against key-recovery attacks, and remains secure under random leakage of the key. Next, as a key technical tool, we extend the oracle indistinguishability framework of Zhandry in two directions: we consider distributions on *functions* rather than strings, and we also consider a relative setting, where an additional oracle, possibly correlated with the distributions, is given to the adversary as well. This enables a hybrid argument to prove the security of NMAC. Security proofs for other constructions follow similarly.

**Keywords:** Cascade construction · NMAC · HMAC · Augmented cascade · AMAC · PRF domain extension · Quantum query · Quantum security · Post-quantum cryptography

## 1 Introduction

After Shor proposed his celebrated quantum algorithm for solving integer factorization and discrete logarithms efficiently, it became apparent that once practical quantum computers become reality, a large part of public-key cryptography, including elliptic curve cryptography and RSA, will be completely broken. Therefore, research in *post-quantum cryptography* has been emerging: new cryptographic algorithms are designed which can still run on conventional classical computers, but their security holds against potential quantum attacks.



There are two possible approaches for modeling quantum attacks in post-quantum cryptography. One is to assume a quantum attacker who has only quantum computational capabilities. In other words, a classical attacker who has a quantum computer in its garage. Such an attacker can run quantum algorithms, but its interaction with the environment remains classical. In such an adversarial model, while some important classical proof techniques do not carry over such as rewinding [16, 19], there are also many examples of existing security proofs that go through relatively easily as long as we switch to hardness assumptions which are not broken by quantum computers [14].

On the other hand, we can be more conservative, and design cryptographic schemes secure against quantum attackers who have not only quantum computational capabilities, but are also capable of interacting quantumly with the environment. In other words, such an attacker can access the cryptographic primitive under attack in quantum *superposition*. Such a scheme would be secure not only now, but also in the far future when quantum computing and quantum networking technologies become prevalent and ubiquitous, and could be also used as a subprotocol in larger quantum computing protocols. We take this adversarial model in this work and refer to this security notion as *quantum security* [20].

Proving quantum security is notoriously challenging. Classically, when an adversary has access to an oracle, each query examines only one point in the domain of the oracle, and that fact is often used crucially in classical security proofs. On the other hand, when an adversary can make superposed queries, each query can potentially probe all points in the input domain in superposition. Therefore, for example, one cannot perform lazy sampling when simulating such an oracle. In fact, there are schemes which are secure classically but fail to be quantum-secure. For example, Kuwakado and Morii showed that three-round Luby-Rackoff cipher [10] and Even-Mansour cipher [11] do not have quantum security, even though they are secure classically.

Later in a series of works [5, 6, 20], the quantum security of several basic primitives, such as PRFs, MACs and signatures, was proved. However one important question was still largely unclear, as Boneh and Zhandry noted [6]:

Can we construct a quantum-secure PRF for a large domain from a quantum-secure PRF for a small domain? In particular, do the CBC-MAC or NMAC constructions give quantum-secure PRFs?

Unfortunately, in Crypto 2016, Kaplan et al. showed that many popular MACs and authenticated encryption schemes are not quantum-secure [9]. For example, CBC-MAC is shown to be insecure when the adversary is allowed to make quantum queries, even when the underlying blockcipher is quantum-secure, and the number of blocks are fixed. Since it is known that a quantum-secure PRF is also quantum-secure as a MAC [5], this shows that CBC-MAC is not a quantum-secure PRF, and the same is true for many other blockcipher-based MACs attacked in the paper. Similar results were independently discovered by Santoli and Schaffner in [13]. This brings us to the basic question:

*Is domain extension for PRFs possible in the quantum setting?*



## 1.1 Our Contributions

In this paper, we give a positive answer to this question. Our discovery is that NMAC and related schemes like HMAC, AMAC, and the (fixed-length) cascade construction are quantum-secure as PRFs. Together with results in [9], our work provides almost a complete picture on the PRF domain extension problem in the quantum world. We highlight some of our main proof ideas and contributions, followed by a gentle technical overview.

- **A general framework for oracle-indistinguishability of function distributions.** All constructions consist of iterated evaluations of the basic PRF, and the output from previous round is used as the *key* to determine the PRF in the next round. This is essentially giving multiple PRF oracles  $F(k_i, \cdot)$  with independent keys  $k_i$  to the adversary. Luckily since the number of oracles is polynomially bounded classically (i.e., number of adversary’s queries), this does not give the adversary more power by a simple hybrid argument relating to the standard PRF indistinguishability. However, when we allow quantum-accessible oracles, in effect, the adversary can query in quantum superposition exponentially many PRF oracles each with an independently random key. Our first technical contribution shows that, the standard notion of quantum-secure PRF implies this seemingly stronger notion, which enables us to prove security of the cascade construction (for fixed-length inputs) already. More generally we view this as *oracle-indistinguishability* of distributions over *functions*. Therefore we extend Zhandry’s work to this setting and show equivalence between ordinary and oracle indistinguishability. We further generalize it, for applications in NMAC for example, to the setting that some additional oracle possibly dependent on the two distributions under consideration is also given to the adversary (we call this relative oracle-indistinguishability).
- **Direct hybrid argument for NMAC and variants.** NMAC and other variants can be viewed as “encrypted” version of the cascade construction by evaluating the output from cascade by another function (e.g., PRF under an independent key). Classical security proofs usually proceed by reducing to some property of its inner cascade. For example the famous “hash-then-PRF” paradigm states that the composition of a (*computationally*) *almost universal hash function* with a *PRF* gives a secure PRF with larger domain. Bellare [1] shows that the cascade construction is indeed computationally almost universal, and the composition theorem implies that NMAC is a secure PRF immediately. However, it is easy to see that this would not work in the quantum world; there are many universal hash functions with nontrivial periods, and if we start with such a periodic universal hash function, any hash-then-PRF construction inherits that period, which can be detected efficiently by quantum Fourier sampling. Therefore, one cannot prove the quantum security of hash-then-PRF constructions by relying solely on the (computationally almost) universality and the PRF security. Another approach by Gaži, Pietrzak, and Rybár [7] proves the security of NMAC by reducing it to the security of the cascade construction against *prefix-free* queries. However the notion of

prefix-free does not have a natural counterpart in the regime of quantum superposition queries. Instead, we prove the security of NMAC by a direct hybrid argument based on our relative oracle-indistinguishability framework for function distributions. We stress that this also provides an alternative (and cleaner in our opinion) proof for *classical security* as well.

- **Further properties of quantum-secure PRFs.** In proving the security of these constructions, we also give further characterizations and strengthened properties of PRFs. Specifically, we show that a quantum-secure PRF is also secure against key-recovery attacks, and in addition a PRF remains indistinguishable from a random oracle even if the PRF key is leaked in some restricted way. While the corresponding classical results are more or less straightforward, they face considerable difficulties to carry through quantumly. We hence demonstrate more examples and tools of quantum proof techniques where classical security can be “lifted” to quantum security.

*Technical Overview.* NMAC is a construction producing a variable-input-length PRF  $\text{NMAC}[f]$ , given a secure PRF  $f : \{0, 1\}^c \times \{0, 1\}^b \rightarrow \{0, 1\}^c$  (with  $b \geq c$ )<sup>1</sup>. Here, the first input argument is the key  $k \in \{0, 1\}^c$ , the second input argument is the message block  $x \in \{0, 1\}^b$ , and the output  $f(k, x) = y \in \{0, 1\}^c$  has the same bit length as the key. NMAC turns this  $f$  into a PRF with the key length of  $2c$ , the output length of  $c$ , and the unbounded input length by

$$\text{NMAC}[f]((k_1, k_2), x_1 \dots x_l) := f(k_2, \text{Casc}[f](k_1, x_1 \dots x_l) \| 0^{b-c}),$$

where  $\text{Casc}[f]$  is the *cascade construction* given as

$$\text{Casc}[f](k, x_1 \dots x_l) = f(\dots f(f(k, x_1), x_2), \dots, x_l).$$

To explain our methods, first let us discuss the cascade construction. It is well-known that the cascade construction would not be a secure PRF if messages of variable lengths are allowed. For example, an adversary may query  $y = \text{Casc}[f](k, x_1) = f(k, x_1)$ , and compute  $f(y, x_2) = \text{Casc}[f](k, x_1 x_2)$ , then query  $\text{Casc}[f](k, x_1 x_2)$  to check if the queried oracle is  $\text{Casc}[f]$  or a true random function. To prevent such an *extension attack*, one obvious way is to fix the number of blocks. More generally, one can prove security against prefix-free adversaries, who never make queries  $m$  and  $m'$  where  $m$  is a proper prefix of  $m'$ . In fact, the cascade construction is proved to be secure in this sense in [4]. To achieve full security, one would process the output of the cascade construction further, and this would give us schemes like NMAC/HMAC or AMAC.

**Quantum security of fixed-length cascade.** For quantum security, there seems no natural analogue of prefix-freeness in presence of quantum superposed

<sup>1</sup> To be precise, the definition of NMAC given here is a simplified version which is not exactly the same as the original definition given in [3], which for example can handle messages whose lengths are not divisible by the block length  $b$ . However, the differences do not affect the security, so previous works on NMAC, like [1, 7], also analyzed this simplified version.

queries. Instead, we consider fixed-input-length cascade  $\text{Casc}_l[f]$ , processing messages of total block length  $l$ , for arbitrary but fixed  $l$ .

It is easy to observe that, when  $b = 1$ , the  $l$ -fold cascade  $\text{Casc}_l[f]$  is the same as the Goldreich-Goldwasser-Micali construction [8] of a PRF out of a secure PRG. Zhandry in [20] proved that if the underlying PRG is secure against polynomial-time quantum adversaries, then the GGM construction remains quantum-secure. In fact, a PRG is equivalent to a PRF with a polynomial-size domain, therefore Zhandry's proof almost immediately applies to  $\text{Casc}_l[f]$  with such a small-sized PRF  $f$ . But, to remove the small-domain restriction, we need more work.

To get a sense of the general difficulty of proving quantum security, we briefly review the classical GGM proof. Roughly speaking, two hybrid arguments are used to construct a distinguisher for the underlying PRG from a distinguisher for the GGM construction; one hybrid argument is over the bit-length of the message inputs of the GGM PRF, and the other is over the individual queries made by the adversary. When trying to adapt the classical proof to quantum security, the first hybrid is not at all problematic, but the second hybrid is not usable; since the adversary in general makes many superposed queries which examine all bitstrings of the given length, the fact that only polynomially-many bitstrings are examined by queries of the adversary is no longer true in the quantum setting.

Zhandry resolves this, by observing that the second hybrid is in fact not necessary, and instead the first hybrid can be carried out by relying on the *oracle security* of the underlying PRG. Suppose  $D$  is a distribution on a set  $\mathcal{Y}$ . Let us define  $D^{\mathcal{X}}$  as a distribution of functions of form  $\mathcal{X} \rightarrow \mathcal{Y}$  where for each  $x \in \mathcal{X}$ , a function value  $y \in \mathcal{Y}$  is chosen independently according to  $D$ . Then, two distributions  $D_1$  and  $D_2$  are said to be *oracle-indistinguishable*, if  $D_1^{\mathcal{X}}$  and  $D_2^{\mathcal{X}}$  are indistinguishable for all  $\mathcal{X}$ . We also say that a PRG  $G$  is *oracle-secure*, if its output distribution is oracle-indistinguishable from the uniform random distribution. This notion expresses indistinguishability of possibly exponentially many independent samples from the PRG (indexed by each  $x \in \mathcal{X}$ ) and possibly exponentially many uniform random numbers, and oracle indistinguishability together with the first hybrid argument gives the security proof of GGM, both classically and quantumly. In the classical case, the oracle indistinguishability can be proved via a hybrid argument over the total number of adversarial queries, since at most polynomially many of the samples will be examined by the adversary. On the other hand, in the quantum case, a completely different approach is needed, which is given by Zhandry's "small-range distributions".

Returning to the cascade construction, we need to work with PRFs instead of PRGs. We may follow the same outline of the proof for the GGM construction, except we need oracle security of PRFs. Hence, we adapt the notion of oracle indistinguishability to function distributions. When  $D$  is a distribution of functions of form  $\mathcal{X} \rightarrow \mathcal{Y}$ , then for any set  $\mathcal{Z}$ , we define  $D^{\mathcal{Z}}$  as the distribution of functions of form  $f : \mathcal{Z} \times \mathcal{X} \rightarrow \mathcal{Y}$ , sampled by choosing  $f(z) \leftarrow D$  independently for each  $z \in \mathcal{Z}$ . (Note that we are using the 'currying' isomorphism here, regarding  $f$  as  $f : \mathcal{Z} \rightarrow \mathcal{Y}^{\mathcal{X}}$ .) Then, the oracle indistinguishability of  $D_1$  and  $D_2$

can be defined as indistinguishability of  $D_1^{\mathcal{Z}}$  and  $D_2^{\mathcal{Z}}$  for every set  $\mathcal{Z}$ . We prove oracle security of secure PRFs also by the small-range distributions.

**Quantum security of NMAC.** We prove the security of NMAC by a direct hybrid argument, adapting the hybrid argument for the cascade construction, rather than reducing to some property of the inner cascade in the classical literature. We start by the standard procedure of swapping the outer instance of the PRF  $f$  with a random oracle  $H$ ; now the modified scheme is  $H(\text{Casc}[f](k, x_1 \dots x_l)) = H(f(\dots f(f(k, x_1), x_2), \dots, x_l))$ . Using a hybrid argument, we would like to repeatedly swap inner instances of the PRF  $f$  with true random functions, until only the true random function remains. However, we need a stronger security notion for the PRF  $f$  to do this: while the random oracle  $H$  prevents the fatal extension attack, still, queries of different block lengths would leak some information on the inner state of PRF instances. In particular, an adversary can make a single-block query  $x$  to obtain  $H(f(k, x))$ , and make a zero-block query to obtain  $H(k)$ . Here, the hash value  $H(k)$  of the secret key  $k$  is leaked by the random oracle  $H$ , and this prevents using the indistinguishability of the PRF  $f$ . What we need is that  $f(k, \cdot)$  should remain pseudorandom even when  $H(k)$  is leaked and the random oracle  $H$  is accessible. We call this property the *security under random leakage*. Nonetheless, we prove that a quantum-secure PRF remains quantum-secure under random leakage, and therefore we do not need to impose this additional condition on a PRF.

To carry out the hybrid argument, however, we need another augmentation to the oracle indistinguishability: while our NMAC security proof itself is in the standard model, a random oracle  $H$  is introduced during the security proof, and the PRF security under random leakage is inherently a security notion in the (quantum) random oracle model. Hence we introduce and study oracle indistinguishability of function distributions, *relative to a random oracle  $H$* . The function distributions may be in general dependent on the random oracle  $H$ , and an adversary always in addition has access to  $H$  to attack indistinguishability or oracle indistinguishability. The tools we introduced so far are enough to enable us to complete the hybrid argument and prove quantum security of NMAC finally.

**Quantum security of augmented cascade and AMAC.** In [2], Bellare, Bernstein, and Tessaro prove PRF security of AMAC. In fact, they analyze ACSC, which is the *augmented cascade*. We can say that ACSC is to AMAC as NMAC is to HMAC. In ACSC, the output of the usual cascade construction  $\text{Casc}[f]$  is further processed by a keyless output transform  $\text{Out}$ , which is typically truncation:  $\text{Out}(b_1 \dots b_c) = b_1 \dots b_r$  for some  $r < c$ . They show that the augmented cascade is a secure PRF, if  $f$  is secure under  $\text{Out}$ -leakage, that is,  $f(k, \cdot)$  remains pseudorandom even when  $\text{Out}(k)$  is leaked. In this paper, using oracle indistinguishability of functions, we also prove that ACSC is quantum-secure if  $f$  is secure under  $\text{Out}$ -leakage.

*Organization.* Sect. 2 introduces basic notations and definitions. We develop our technical tool of oracle distribution for function distributions in Sect. 3.

Combined with the further properties of PRFs we establish in Sect. 4, we prove quantum security of NMAC and other constructions in Sect. 5.

## 2 Preliminaries

### 2.1 Notations and Conventions

In this paper, all constructions and security notions are *implicitly* asymptotic: many quantities and objects are parametrized by the main security parameter  $\lambda$ , but for simplicity, we will often omit writing the dependency on  $\lambda$  explicitly. Although it is in reality a family of sets  $\{\mathcal{X}_\lambda\}_\lambda$ , we write it simply as  $\mathcal{X}_\lambda$ , or even just  $\mathcal{X}$ . Similarly, a function  $f : \mathcal{X} \rightarrow \mathcal{Y}$  in such a case is really a family  $\{f_\lambda : \mathcal{X}_\lambda \rightarrow \mathcal{Y}_\lambda\}_\lambda$  of functions. We also omit the size input  $1^\lambda$  from arguments of polynomial-time computable functions.

A quantity  $p = p(\lambda)$  is *polynomially bounded*, if  $p(\lambda) = O(\lambda^d)$  for some  $d > 0$ . We denote this as  $p(\lambda) = \text{poly}(\lambda)$ , or even,  $p = \text{poly}()$ . Similarly, a quantity  $\epsilon = \epsilon(\lambda)$  is *negligible*, if  $\epsilon(\lambda) \leq 2^{-\omega(\log \lambda)}$ . We denote this as  $\epsilon(\lambda) = \text{negl}(\lambda)$ , or even,  $\epsilon = \text{negl}()$ .

If  $D$  is a distribution, then  $x \leftarrow D$  means  $x$  is sampled according to  $D$ . Also, if  $\mathcal{X}$  is a set, then  $x \leftarrow \mathcal{X}$  means that  $x$  is sampled from  $\mathcal{X}$  uniform randomly.

For any  $r \in \mathbb{N}$ , we define  $[r] := \{0, 1, \dots, r-1\}$ .

Let  $\mathcal{X}$  and  $\mathcal{Y}$  be two sets. We denote by  $\mathcal{Y}^\mathcal{X}$  the set of all functions from  $\mathcal{X}$  to  $\mathcal{Y}$ . We sometimes call it the *function space* from  $\mathcal{X}$  to  $\mathcal{Y}$ .

In this paper, we are mostly interested in quantum security. Unless explicitly mentioned otherwise, by an *adversary*, we always mean a polynomial-time quantum algorithm which may have access to some oracles, to which it can make polynomially many quantum superposed queries. Similarly, when we mention ‘security’, unless it is in a context describing previous works and comparing them with ours, it means quantum security. On the other hand, by an ‘algorithm’, we always mean a classical algorithm, unless mentioned otherwise.

### 2.2 I.i.d Samples of Functions

Following Zhandry [20], we introduce the notation  $D^\mathcal{X}$  as follows.

**Definition 2.1 (Indexed family of i.i.d. samples).** *Let  $D$  be a probability distribution over a set  $\mathcal{Y}$ , and let  $\mathcal{X}$  be another set. Then, we denote by  $D^\mathcal{X}$  the probability distribution over  $\mathcal{Y}^\mathcal{X}$ , defined such that,  $f$  is sampled according to  $D^\mathcal{X}$  if and only if  $f(x)$  is sampled according to  $D$ , independently for each  $x \in \mathcal{X}$ .*

In other words, if  $f \leftarrow D^\mathcal{X}$ , then  $\{f(x)\}_{x \in \mathcal{X}}$  is an indexed family of i.i.d. samples, where each  $f(x)$  is distributed according to  $D$ .

Suppose  $D$  is a distribution over  $\mathcal{Y}^\mathcal{X}$ . Since  $\mathcal{Y}^\mathcal{X}$  itself is just a set, the previous definition is applicable. Let us clarify this as the following definition.

**Definition 2.2 (Indexed family of i.i.d. samples of functions).** *Let  $D$  be a probability distribution over  $\mathcal{Y}^{\mathcal{X}}$ , and let  $\mathcal{Z}$  be another set. We define the distribution  $D^{\mathcal{Z}}$  of functions  $f \in (\mathcal{Y}^{\mathcal{X}})^{\mathcal{Z}}$  as in Definition 2.1; if  $f$  is sampled according to  $D^{\mathcal{Z}}$ , then  $f(z) \in \mathcal{Y}^{\mathcal{X}}$  is sampled according to  $D$ , independently for each  $z \in \mathcal{Z}$ .*

Then, evaluating  $f(z) \in \mathcal{Y}^{\mathcal{X}}$  on  $x \in \mathcal{X}$  will give a value  $f(z)(x) = y \in \mathcal{Y}$ . Considering the ‘currying’ isomorphism  $(\mathcal{Y}^{\mathcal{X}})^{\mathcal{Z}} \cong \mathcal{Y}^{\mathcal{Z} \times \mathcal{X}}$ , we may regard  $D^{\mathcal{Z}}$  as a distribution over  $\mathcal{Y}^{\mathcal{Z} \times \mathcal{X}}$ , writing  $f(z, x)$ , instead of  $f(z)(x)$ . We will use the two perspectives interchangeably.

In this paper, although our results are *not* in the quantum random oracle model, during the security proofs, we mostly work in the quantum random oracle model. In other words, all players, including the adversary, are given oracle access to a uniform random function  $H : \mathcal{A} \rightarrow \mathcal{B}$ , and various constructions depend on  $H$ . Therefore, we need to consider the case when a distribution  $D$  over  $\mathcal{Y}^{\mathcal{X}}$  depends on  $H$ , that is,  $D$  and the uniform distribution of  $H$  are both marginal distributions of a joint distribution. Therefore, we give a definition of  $D^{\mathcal{Z}}$ , relative to a random oracle  $H$ :

**Definition 2.3 (Indexed family of relative i.i.d. samples of functions).** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, that is, a uniform random function in  $\mathcal{B}^{\mathcal{A}}$ . And let  $D$  be a probability distribution over  $\mathcal{Y}^{\mathcal{X}}$  which depends on  $H$ , and let  $\mathcal{Z}$  be another set. We define the distribution  $D_H^{\mathcal{Z}}$  relative to  $H$  as follows. To jointly sample  $f$  from  $D_H^{\mathcal{Z}}$  and also a particular  $h : \mathcal{A} \rightarrow \mathcal{B}$  as realization of the random variable  $H$ , first sample  $h \leftarrow \mathcal{B}^{\mathcal{A}}$  uniform randomly, and form  $D|h$ , which is the conditional distribution of  $D$  conditioned on the event  $H = h$ . Finally, sample  $f \leftarrow (D|h)^{\mathcal{Z}}$ . When the dependence on the random oracle  $H$  is clear, we abuse the notation and simply write  $D^{\mathcal{Z}}$ , instead of  $D_H^{\mathcal{Z}}$ .*

In other words, when we are in the quantum random oracle model, at first a function  $h$  is sampled uniformly, as a realization of the random variable  $H$ . When a distribution  $D$  is dependent on  $H$ , then sampling  $f \leftarrow D^{\mathcal{Z}}$  means that,  $f(z)$  is independently sampled from  $D|h$ , for each  $z \in \mathcal{Z}$ .

### 2.3 Various Security Notions of PRFs

First, let us define the syntax of the pseudorandom function as follows:

**Definition 2.4 (Pseudorandom function).** *A pseudorandom function (PRF) is a polynomial-time computable function  $f$  of form  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ . We call the sets  $\mathcal{K}$ ,  $\mathcal{X}$ ,  $\mathcal{Y}$  as the key space, the domain, and the codomain of  $f$ , respectively.*

The domain of a PRF may be of fixed size or arbitrarily large. For a blockcipher,  $\mathcal{X}$  would be  $\{0, 1\}^n$  for some  $n$ . On the other hand, for HMAC, the domain  $\mathcal{X}$  is the set of all bitstrings, or bitstrings up to some large fixed length.

In this paper, we are concerned with polynomial-time quantum adversaries who can make quantum superposed queries to their oracles. Therefore, our standard definition of PRF security is as follows:



**Definition 2.5 (Quantum security of PRF).** Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a PRF. We say that  $f$  is secure, if for any adversary  $A$ , we have the following:

$$\mathbf{Adv}_f^{\text{prf}}(A) := \left| \Pr[A^{f(k, \cdot)}() = 1] - \Pr[A^\rho() = 1] \right| = \text{negl}(),$$

where  $k \leftarrow \mathcal{K}$ ,  $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$  are uniformly and independently random.

That is, sampling  $k \leftarrow \mathcal{K}$  and letting  $F$  as  $F(x) := f(k, x)$ , any quantum adversary cannot distinguish  $F$  from a true random function  $\rho : \mathcal{X} \rightarrow \mathcal{Y}$ .

Here,  $\mathbf{Adv}_f^{\text{prf}}(A)$  is the *advantage* of  $A$  in distinguishing  $f(k, \cdot)$  from  $\rho$ , and if  $f$  is a secure PRF, then the advantage is negligible for any adversary  $A$ .

Sometimes, we may want less than the full PRF security against distinguishing attack, and only require the following:

**Definition 2.6 (Security of PRF against key recovery).** Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a PRF. We say that  $f$  is secure against key recovery, if for any adversary  $A$ , we have the following:

$$\mathbf{Adv}_f^{\text{prf-kr}}(A) := \Pr[A^{f(k, \cdot)}() = k] = \text{negl}(),$$

where  $k \leftarrow \mathcal{K}$  is uniformly random.

Classically, it is well known, and indeed trivial to prove that a secure PRF is also secure against key recovery. However, in the quantum world, it is less trivial than classically. We discuss this more in Sect. 4.

Finally, let us present a stronger security notion for PRF, which will be crucial later when we prove the security of NMAC.

**Definition 2.7 (Security of PRF under random leakage).** Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a PRF. We say that  $f$  is secure under random leakage, if for any set  $\mathcal{W}$  and any adversary  $A$ , we have the following:

$$\mathbf{Adv}_f^{\text{prf-rl}}(A) := \left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{\rho, H}(w) = 1] \right| = \text{negl}(),$$

where  $k \leftarrow \mathcal{K}$ ,  $w \leftarrow \mathcal{W}$ ,  $H \leftarrow \mathcal{W}^{\mathcal{K}}$ ,  $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$  are uniform, independent random.

The above notion is related to the leakage-resilient cryptography. Here, the PRF key  $k$  is leaked once, via the leakage function  $H(\cdot)$ . But, this leakage is very weak; the adversary does not choose  $H$ , which is just a random oracle.

## 2.4 NMAC and Related Constructions

In this subsection, we give definitions of NMAC and other hash-based PRFs which we study in this paper.

**Cascade construction.** Suppose that  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  is a PRF where the codomain is the same as the key space  $\mathcal{K}$ . We define the *l-fold cascade* of  $f$ ,

denoted by  $\text{Casc}_l[f] : \mathcal{K} \times \mathcal{X}^l \rightarrow \mathcal{K}$ , as follows: given  $k \in \mathcal{K}$  and  $x_1, \dots, x_l \in \mathcal{X}$ , we define a sequence of values  $y_0, \dots, y_l \in \mathcal{K}$ , recursively.

$$\begin{aligned} y_0 &:= k, \\ y_i &= f(y_{i-1}, x_i), \quad \text{for } i = 1, \dots, l. \end{aligned}$$

Then, the cascade PRF is given as the last value  $y_l$ .

$$\text{Casc}_l[f](k, x_1 \dots x_l) := y_l.$$

In other words,

$$\text{Casc}_l[f](k, x_1 \dots x_l) = f(\dots f(f(k, x_1), x_2), \dots, x_l).$$

From the definition of  $\text{Casc}_l[f]$ , we see  $\text{Casc}_0[f] : \mathcal{K} \times \mathcal{X}^0 \rightarrow \mathcal{K}$  is given as

$$\text{Casc}_0[f](k, \epsilon) = k,$$

where  $\epsilon \in \mathcal{X}^0$  is the empty string of length 0.

**NMAC.** Suppose that  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  is a PRF where the codomain is the same as the key space  $\mathcal{K}$ . Here, we assume that  $|\mathcal{K}| \leq |\mathcal{X}|$ . The *NMAC* of  $f$ , denoted by  $\text{NMAC}[f] : \mathcal{K}^2 \times \mathcal{X}^* \rightarrow \mathcal{K}$  is defined as

$$\text{NMAC}[f]((k_1, k_2), x_1 \dots x_m) := f(k_2, \text{pad}(\text{Casc}_m[f](k_1, x_1 \dots x_m))),$$

where  $\text{pad} : \mathcal{K} \rightarrow \mathcal{X}$  is a simple injective ‘padding function’. Typically, when  $\mathcal{X} = \{0, 1\}^b$  and  $\mathcal{K} = \{0, 1\}^c$ , then  $\text{pad}(k) = k \| 0^{b-c}$ , but the choice of  $\text{pad}$  does not affect the security of NMAC.

**Augmented cascade.** Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  be a PRF where the codomain is the same as the key space  $\mathcal{K}$ , and let  $\text{Out} : \mathcal{K} \rightarrow \mathcal{Y}$  be an unkeyed function. Then, the *augmented cascade*  $\text{ACSC}[f, \text{Out}] : \mathcal{K} \times \mathcal{X}^* \rightarrow \mathcal{Y}$  is

$$\text{ACSC}[f, \text{Out}](k, x_1 \dots x_m) := \text{Out}(\text{Casc}_m[f](k, x_1 \dots x_m)).$$

## 2.5 Implementing Oracles

Here, we are going to discuss which function distributions can be ‘efficiently implemented’. One possible answer is the following:

**Definition 2.8.** Let  $D$  be a function distribution over  $\mathcal{Y}^{\mathcal{X}}$ . We say that  $D$  is efficiently samplable, if there exists a set  $\mathcal{R}$  and a polynomial-time deterministic algorithm  $D.\text{eval} : \mathcal{R} \times \mathcal{X} \rightarrow \mathcal{Y}$ , such that sampling  $f \in \mathcal{Y}^{\mathcal{X}}$  according to the distribution  $D$  can be done by sampling  $r \leftarrow \mathcal{R}$  and defining  $f$  by  $f(x) := D.\text{eval}(r, x)$ . We also require that  $\log |\mathcal{R}| = \text{poly}()$ .

In other words, we may sample a function  $f \leftarrow D$ , by sampling  $r \leftarrow \mathcal{R}$ .

One typical example of an efficiently samplable distribution is  $\text{PRF}_f$  over  $\mathcal{Y}^{\mathcal{X}}$  of a PRF  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ . Here,  $\mathcal{R} = \mathcal{K}$ , and  $\text{PRF}_f.\text{eval} = f$ .



Zhandry shows that in fact we can efficiently ‘implement’ function distributions which are not necessarily efficiently samplable. One such example is the uniform distribution over  $\mathcal{Y}^{\mathcal{X}}$ . While it is not efficiently samplable in the above sense, still, given any adversary  $A$  making at most  $q$  quantum superposed queries, it is possible to implement the uniform distribution for the adversary  $A$  perfectly. This is due to Theorem 3.1 of [21]. Here, we give a slightly extended version as follows, whose proof we defer to the full version of our paper [15], due to page limitation.

**Theorem 2.9.** *Let  $A$  be an adversary having oracle access to  $O_1, \dots, O_t$ , and makes at most  $q_i$  quantum queries to  $O_i \in \mathcal{Y}_i^{\mathcal{X}_i}$  for  $i = 1, \dots, t$ . If we draw  $O_i$  from any joint distribution for  $i = 1, \dots, t$ , then for every  $v$ , the quantity  $\Pr[A^{O_1, \dots, O_t}() = v]$  is a linear combination of the quantities*

$$\Pr[\forall i \in \{1, \dots, t\}, \forall j \in \{1, \dots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}]$$

for all possible settings of the values  $x_j^{(i)} \in \mathcal{X}$  and  $y_j^{(i)} \in \mathcal{Y}$ .

Hence if  $D, D'$  are distributions over  $\mathcal{Y}^{\mathcal{X}}$  which are  $2q$ -wise equivalent, i.e.,

$$\Pr_{O \leftarrow D}[\forall i \in \{1, \dots, 2q\}, O(x_i) = y_i] = \Pr_{O \leftarrow D'}[\forall i \in \{1, \dots, 2q\}, O(x_i) = y_i],$$

for any distinct  $x_1, \dots, x_{2q} \in \mathcal{X}$  and any  $y_1, \dots, y_{2q} \in \mathcal{Y}$ , then when  $A$  makes at most  $q$  queries to its oracle, for any output value  $v$  of  $A$ , we have

$$\Pr_{O \leftarrow D}[A^O() = v] = \Pr_{O \leftarrow D'}[A^O() = v].$$

In particular, for any adversary making at most  $q$  quantum queries, the uniform random function  $U \in \mathcal{Y}^{\mathcal{X}}$  can be efficiently ‘implemented’ by any  $2q$ -wise independent function family. We use the following standard fact (for example, see p. 72 of [18]):

**Proposition 2.10.** *For every  $n, m, k$ , there exists a family of  $k$ -wise independent functions  $\mathcal{H} = \{h : \{0, 1\}^n \rightarrow \{0, 1\}^m\}$  such that, choosing a function  $h$  from  $\mathcal{H}$  takes  $k \cdot \max\{n, m\}$  random bits, and evaluating  $h \in \mathcal{H}$  takes time  $\text{poly}(n, m, k)$ .*

Therefore, implementing a uniform distribution in  $\mathcal{Y}^{\mathcal{X}}$  for any adversary making  $q$  quantum queries requires sampling  $2q \cdot \max\{\log |\mathcal{X}|, \log |\mathcal{Y}|\}$  bits, and answering one query takes time  $\text{poly}(\log |\mathcal{X}|, \log |\mathcal{Y}|, q)$ .

Let us propose the following definition which captures both efficiently samplable distributions and uniform distributions.

**Definition 2.11.** *Let  $D$  be a function distribution over  $\mathcal{Y}^{\mathcal{X}}$ . We say that  $D$  is bounded samplable, if there exists a set  $\mathcal{R}^{(q)}$  for each  $q$  and a polynomial-time deterministic algorithm  $D.\text{eval} : 1^* \times \bigcup_q \mathcal{R}^{(q)} \times \mathcal{X} \rightarrow \mathcal{Y}$  such that, if we sample  $f \in \mathcal{Y}^{\mathcal{X}}$  according to the distribution  $D$ , and sample  $f' \in \mathcal{Y}^{\mathcal{X}}$  by sampling  $r \leftarrow \mathcal{R}^{(q)}$  and defining  $f'$  by  $f'(x) := D.\text{eval}(1^q, r, x)$ , then two random functions  $f$  and  $f'$  are  $2q$ -wise equivalent. Also, we require that  $\log |\mathcal{R}^{(q)}| = \text{poly}(\lambda, q)$ .*

If  $D$  is bounded samplable, then a function  $f$  can be sampled according to  $D$  by sampling  $r \leftarrow \mathcal{R}^{(q)}$ , and it can be evaluated by  $f(x) = D.eval(1^q, r, x)$ . The resulting distribution may not be identical to  $D$ , but would be enough to ‘fool’ any adversary making at most  $q$  queries. The following lemma is obvious.

**Lemma 2.12.** *For any  $\mathcal{X}, \mathcal{Y}$ , the uniform distribution over  $\mathcal{Y}^{\mathcal{X}}$  is bounded samplable.*

Moreover, we can see from Theorem 2.9 that, when  $A$  has access to several oracles  $O_1, \dots, O_t$  sampled according to  $D_1, \dots, D_t$ , and if they are *independent*, then if the distributions  $D_i$  are all bounded samplable, then they can be ‘implemented’ separately: sampling  $f_i \leftarrow D_i$  can be done by sampling  $r_i \leftarrow \mathcal{R}_i^{(q)}$ , and letting  $f_i(x) = D_i.eval(1^{q_i}, r_i, x)$  for all  $i = 1, \dots, t$ , since we have

$$\begin{aligned} & \Pr[\forall i \in \{1, \dots, t\}, \forall j \in \{1, \dots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}] \\ &= \prod_{i=1}^t \Pr[\forall j \in \{1, \dots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}] \end{aligned}$$

Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, that is, a uniform random function. For our purpose, we need to ‘relativize’ the efficient samplability and the bounded samplability, with respect to  $H$ . First, let us give the following definitions.

**Definition 2.13.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D_i$  be a distribution over  $\mathcal{Y}_i^{\mathcal{X}_i}$ , for  $i = 1, \dots, t$ . We say that  $D_1, \dots, D_t$  are conditionally independent relative to  $H$ , if for any  $h \in \mathcal{B}^{\mathcal{A}}$ , the distributions  $D_1, \dots, D_t$  are independent, conditioned on the event that  $H = h$ .*

**Definition 2.14.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D, D'$  be distributions over  $\mathcal{Y}^{\mathcal{X}}$ . We say that  $D, D'$  are  $k$ -wise equivalent relative to  $H$ , if*

$$\begin{aligned} & \Pr_{O \leftarrow D}[\forall i \in \{1, \dots, k\}, O(x_i) = y_i \mid H = h] \\ &= \Pr_{O \leftarrow D'}[\forall i \in \{1, \dots, k\}, O(x_i) = y_i \mid H = h], \end{aligned}$$

for any distinct  $x_1, \dots, x_k \in \mathcal{X}$ , any  $y_1, \dots, y_k \in \mathcal{Y}$ , and any  $h \in \mathcal{B}^{\mathcal{A}}$ .

Then, we are ready to define relative versions of efficient samplability and bounded samplability as follows.

**Definition 2.15.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D$  be a distribution over  $\mathcal{Y}^{\mathcal{X}}$ . We say that  $D$  is efficiently samplable relative to  $H$ , if there exists a set  $\mathcal{R}$  and a polynomial-time deterministic oracle algorithm  $D.eval^H : \mathcal{R} \times \mathcal{X} \rightarrow \mathcal{Y}$  such that sampling  $f \in \mathcal{Y}^{\mathcal{X}}$  according to  $D$  can be done by sampling  $r \leftarrow \mathcal{R}$  and defining  $f$  by  $f(x) := D.eval^H(r, x)$ . We also require that  $\log |\mathcal{R}| = \text{poly}()$ .*

**Definition 2.16.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D$  be a distribution over  $\mathcal{Y}^{\mathcal{X}}$ . We say that  $D$  is bounded samplable relative to  $H$ , if there exists a set  $\mathcal{R}^{(q)}$  for each  $q$ , and a polynomial-time deterministic oracle algorithm  $D.eval^H : 1^* \times \bigcup_q \mathcal{R}^{(q)} \times \mathcal{X} \rightarrow \mathcal{Y}$  such that, if we sample  $f \in \mathcal{Y}^{\mathcal{X}}$*

according to  $D$ , and sample  $f' \in \mathcal{Y}^{\mathcal{X}}$  by sampling  $r \leftarrow \mathcal{R}^{(q)}$  and defining  $f'$  by  $f'(x) := D.\text{eval}^H(1^q, r, x)$ , then  $f$  and  $f'$  are  $2q$ -wise equivalent relative to  $H$ . We also require that  $\log |\mathcal{R}^{(q)}| = \text{poly}(\lambda, q)$ .

We have the following lemma about ‘relative implementation’ of an oracle.

**Lemma 2.17.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D$  be a distribution over  $\mathcal{Y}^{\mathcal{X}}$ . Suppose  $D$  is bounded samplable relative to  $H$ , and suppose an adversary  $A^{O,H}$  makes at most  $q$  queries to  $O$ , and at most  $q_H$  queries to  $H$ . Let  $D'$  be the distribution of function  $O$  sampled by sampling  $r \leftarrow \mathcal{R}^{(q)}$  and letting  $O(x) := D.\text{eval}^H(1^q, r, x)$ . Then, we have*

$$\Pr_{O \leftarrow D}[A^{O,H}() = v] = \Pr_{O \leftarrow D'}[A^{O,H}() = v],$$

for any possible output value  $v$  of  $A$ .

Lemma 2.17 is an extension of the previous result that, if a distribution  $D$  is bounded samplable, then for each adversary  $A$ , we can implement  $D$  to completely fool  $A$ . This time, Lemma 2.17 says that if  $D$  is bounded samplable relative to a random oracle  $H$ , then for any adversary  $A$ , we can implement  $D$  to completely fool  $A$ , even when  $D$  is dependent on  $H$  and  $A$  also has access to  $H$ . The proof can be done by simple arguments using conditional probability. Due to page limitation, we defer the proof to the full version of this paper [15].

Similar to the non-relative case, if  $D_1, \dots, D_t$  are all distributions bounded samplable relative to  $H$ , and if they are conditionally independent relative to  $H$ , then it is easy to see that we can implement each oracle  $O_i$  sampled from  $D_i$  separately, which will fool any adversary which has oracle access to not only  $O_1, \dots, O_t$ , but also to the random oracle  $H$ .

Let us give another lemma, to be used later. Note that the definition of the distribution  $D^{\mathcal{Z}}$  and its dependence on  $H$  is given in Definition 2.3.

**Lemma 2.18.** *Suppose that  $D$  is an efficiently samplable distribution over  $\mathcal{Y}^{\mathcal{X}}$  relative to a random oracle  $H : \mathcal{A} \rightarrow \mathcal{B}$ . If  $\mathcal{Z}$  is any set, then  $D^{\mathcal{Z}}$  is bounded samplable relative to  $H$ .*

Lemma 2.18 is generalization of the following: if  $D$  is an efficiently samplable distribution over a set  $\mathcal{Y}$ , then Zhandry points out in [21] that the distribution  $D^{\mathcal{X}}$  can be ‘constructed’ for any set  $\mathcal{X}$ : if  $\mathcal{R}$  is the randomness space for sampling  $D$ , and if  $y = f(r)$  is the element of  $\mathcal{Y}$  sampled using randomness  $r \in \mathcal{R}$ , then we can implement  $O \leftarrow D^{\mathcal{X}}$  by first implementing a random function  $\rho \in \mathcal{R}^{\mathcal{X}}$  and then letting  $O(x) = f(\rho(x))$ . In our terminology, the distribution  $D^{\mathcal{X}}$  is bounded samplable.

Lemma 2.18 says that, when we form  $D^{\mathcal{Z}}$  from an efficiently samplable distribution  $D$  of functions (relative to a random oracle  $H$ ), the result is analogous: the distribution  $D^{\mathcal{Z}}$  is bounded samplable (relative to  $H$ ). The proof is similar, but the fact that we are dealing with functions, and also relative to a random oracle, makes this slightly more complex. Again, we defer the proof to the full version of this paper [15].

### 3 Relative Oracle Indistinguishability of Functions

In this paper, we are primarily interested in distributions of functions. We also consider the case where these distributions may be dependent on a random oracle  $H : \mathcal{A} \rightarrow \mathcal{B}$ , and the adversary has access to  $H$  as well.

Zhandry [20] defines “oracle indistinguishability” of two distributions  $D_1, D_2$  over a set  $\mathcal{Y}$ . We adapt this notion to our case, giving the following definitions.

**Definition 3.1 (Relative indistinguishability of functions).** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D_1, D_2$  be two distributions on  $\mathcal{Y}^{\mathcal{X}}$ , which are conditionally independent relative to  $H$ . Then, we say that  $D_1$  and  $D_2$  are indistinguishable relative to  $H$ , if for any adversary  $A$ , the distinguishing advantage*

$$\mathbf{Adv}_{D_1, D_2, H}^{\text{rel-dist}}(A) := \left| \Pr_{O \leftarrow D_1} [A^{O, H}() = 1] - \Pr_{O \leftarrow D_2} [A^{O, H}() = 1] \right|$$

*is negligible.*

**Definition 3.2 (Relative oracle indistinguishability of functions).** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D_1, D_2$  be two distributions over  $\mathcal{Y}^{\mathcal{X}}$ , which are conditionally independent relative to  $H$ . We say that  $D_1$  and  $D_2$  are oracle-indistinguishable relative to  $H$ , if, for any set  $\mathcal{Z}$ , and any adversary  $A$ , we have the following:*

$$\mathbf{Adv}_{D_1, D_2, \mathcal{Z}, H}^{\text{oracle-rel-dist}}(A) := \left| \Pr_{O \leftarrow D_1^{\mathcal{Z}}} [A^{O, H}() = 1] - \Pr_{O \leftarrow D_2^{\mathcal{Z}}} [A^{O, H}() = 1] \right|$$

*is negligible.*

Note that, when  $\mathcal{A}$  and  $\mathcal{B}$  are singleton sets, the random oracle  $H$  is trivial, and we obtain non-relativized definitions of the above. Moreover we are only interested in the case when  $D_1$  and  $D_2$  are conditionally independent relative to  $H$ , which would make sense since these are definitions of indistinguishability in the quantum random oracle model.

The following is our main result regarding oracle indistinguishability.

**Theorem 3.3.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D_1, D_2$  be two function distributions over  $\mathcal{Y}^{\mathcal{X}}$  for some  $\mathcal{X}, \mathcal{Y}$ . Suppose that both  $D_1^{\mathcal{Z}}$  and  $D_2^{\mathcal{Z}}$  are bounded samplable relative to  $H$ , for any set  $\mathcal{Z}$ . Further, suppose that  $D_1$  and  $D_2$  are conditionally independent relative to  $H$ , and indistinguishable relative to  $H$ . Then, they are oracle-indistinguishable relative to  $H$ .*

Concretely, for any adversary  $A^{O, H}$  making at most  $q$  queries to  $O$  and at most  $q_H$  queries to  $H$ , we can construct an adversary  $A_{\text{rd}}^{O', H}$  satisfying

$$\mathbf{Adv}_{D_1, D_2, \mathcal{Z}, H}^{\text{oracle-rel-dist}}(A) < 12q^{3/2} \sqrt{\mathbf{Adv}_{D_1, D_2, H}^{\text{rel-dist}}(A_{\text{rd}})}.$$

Moreover,  $A_{\text{rd}}^{O', H}$  makes at most  $2q$  queries to  $O'$  and  $q_H + 2(q_{e_1} + q_{e_2})q$  queries to  $H$ . Here,  $q_{e_i}$  is the maximum number of queries to  $H$  needed by one invocation to the evaluation algorithm  $D_i^{\mathcal{Z}}.\text{eval}^H()$ , for  $i = 1, 2$ , respectively.

Theorem 3.3 says that, if two function distributions are indistinguishable (relative to  $H$ ), and if they satisfy some additional conditions, then they are also oracle-indistinguishable (relative to  $H$ ).

Our proof of Theorem 3.3 proceeds similarly as Zhandry's proof of the corresponding result in [20]. Therefore, we are going to defer the complete proof to the full version of this paper [15], but here let us describe some outline of the proof.

To prove oracle indistinguishability of indistinguishable distributions over a set, Zhandry uses 'small-range distribution' [20], given as follows.

**Definition 3.4.** *Given a distribution  $D$  on  $\mathcal{Y}$ , we define  $\text{SR}_r^D(\mathcal{X})$  as the following distribution on functions  $O \in \mathcal{Y}^{\mathcal{X}}$ :*

- For each  $i \in [r]$ , sample a value  $y_i \in \mathcal{Y}$  according to the distribution  $D$ .
- For each  $x \in \mathcal{X}$ , sample a uniform random  $i \in [r]$  and set  $O(x) = y_i$ .

This can be applied to a distribution  $D$  over  $\mathcal{Y}^{\mathcal{X}}$ : since  $\mathcal{Y}^{\mathcal{X}}$  is just a set, surely we may talk about a small-range distribution for  $D$ . Let us make this explicit:

**Definition 3.5.** *Given a function distribution  $D$  on  $\mathcal{Y}^{\mathcal{X}}$ , we define the small-range distribution  $\text{SR}_r^D(\mathcal{Z})$  as the following distribution on functions  $O \in \mathcal{Y}^{\mathcal{Z} \times \mathcal{X}}$ :*

- For each  $i \in [r]$ , sample a function  $f_i \in \mathcal{Y}^{\mathcal{X}}$  according to the distribution  $D$ .
- For each  $z \in \mathcal{Z}$ , sample a uniform random  $i \in [r]$  and set  $O(z) = f_i$ .

Following Definition 2.3, when  $D$  depends on the random oracle  $H$ , we interpret  $\text{SR}_r^D(\mathcal{Z})$  as follows:

**Definition 3.6.** *Given a function distribution  $D$  on  $\mathcal{Y}^{\mathcal{X}}$  depending on a random oracle  $H : \mathcal{A} \rightarrow \mathcal{B}$ , we define the small-range distribution  $\text{SR}_r^D(\mathcal{Z})$  as follows. To jointly sample  $O$  from  $\text{SR}_r^D(\mathcal{Z})$  and also a particular  $h : \mathcal{A} \rightarrow \mathcal{B}$  as realization of the random variable  $H$ , first sample  $h \leftarrow \mathcal{B}^{\mathcal{A}}$  uniform randomly, and form  $D|h$ . Then,*

- For each  $i \in [r]$ , sample a function  $f_i \in \mathcal{Y}^{\mathcal{X}}$  according to  $D|h$ .
- For each  $z \in \mathcal{Z}$ , sample a uniform random  $i \in [r]$  and set  $O(z) = f_i$ .

Then, we have the following theorem.

**Theorem 3.7.** *Let  $H : \mathcal{A} \rightarrow \mathcal{B}$  be a random oracle, and let  $D$  be a function distribution over  $\mathcal{Y}^{\mathcal{X}}$  which is not necessarily independent from  $H$ . Suppose that  $A$  is an adversary making at most  $q$  queries to an oracle  $O \in \mathcal{Y}^{\mathcal{X}}$ , and at most  $q_H$  queries to the random oracle  $H$ . Then, we have*

$$\left| \Pr_{O \leftarrow \text{SR}_r^D(\mathcal{Z})}[A^{O,H}() = 1] - \Pr_{O \leftarrow D^{\mathcal{Z}}}[A^{O,H}() = 1] \right| < \frac{16q^3}{r},$$

for any  $r > 0$ , and any set  $\mathcal{Z}$ .

*Remark 3.8.* Note that Theorem 3.7 holds, whether  $D$  is bounded samplable or not. The bound in the theorem does not depend on  $q_H$  either.

Just like the corresponding result in [20], Theorem 3.7 says that the distribution  $D^{\mathcal{Z}}$ , which is the distribution of an exponentially many independent samples of  $D$  indexed by  $\mathcal{Z}$  is, in fact, indistinguishable from similar collection of samples, this time duplicated from only  $r$  independent samples. Theorem 3.7 also says that the result holds regardless of dependence to a random oracle  $H$ . We give the complete proof in the full version of this paper [15].

In the classical cases, we can prove oracle indistinguishability of two indistinguishable distributions by a hybrid argument over the adversarial queries: even though  $O \leftarrow D^{\mathcal{Z}}$  can be considered as a collection of exponentially many independent samples of  $D$ , if a classical adversary  $A$  makes  $q$  queries  $z_1, \dots, z_q$ , then all  $A$  examines are  $O(z_1), \dots, O(z_q) \leftarrow D$ , and these can be swapped to samples from another indistinguishable distribution  $D'$  one by one.

On the other hand, in the quantum case, each query can be superposed, so the previous approach would not work. Small-range distribution solves this: once we switch to a small-range distribution of size  $r$ , then only  $r$  independent samples from a distribution  $D$  are involved, and they can be swapped to samples from another indistinguishable distribution  $D'$  one by one, and the resulting small-range distribution can be once again switched to  $D'^{\mathcal{Z}}$ . Hence, the proof of Theorem 3.3 is again a standard hybrid argument, which we defer to the full version of this paper [15].

## 4 Security Against Key Recovery and Security Under Random Leakage

In this section, we characterize further properties about a quantum-secure PRF, which will be useful later (to establish quantum security of NMAC for example). We first show that a secure PRF is also secure against key recovery. Using this, we prove that a secure PRF is secure under random leakage as well. This further enables us to study oracle security under random leakage for PRFs.

### 4.1 Security of PRFs Against Key Recovery

First, we have the following theorem:

**Theorem 4.1.** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a secure PRF. Suppose that both the domain and the codomain of  $f$  are superpolynomially large:  $|\mathcal{X}|, |\mathcal{Y}| \geq 2^{\omega(\log \lambda)}$ . Then,  $f$  is also secure against key recovery.*

Concretely, for any adversary  $A^{f(k, \cdot)}$  making at most  $q$  queries to  $f(k, \cdot)$  with uniform random  $k \leftarrow \mathcal{K}$ , we can construct an adversary  $A_d$  that makes at most  $q + 1$  queries such that

$$\mathbf{Adv}_f^{\text{prf-kr}}(A) \leq \mathbf{Adv}_f^{\text{prf}}(A_d) + \frac{1}{|\mathcal{Y}|} + \frac{4q}{\sqrt{|\mathcal{X}|}}.$$



Classically, it is easy to prove that a secure PRF  $f$  is also secure against key recovery: if  $A$  is a classical key recovery attacker, then using  $A$ , we can construct a PRF distinguisher  $B$ :  $B^O$  runs  $A^O$ , while answering any query of  $A$  by its own query. In the end, if  $A$  outputs a candidate  $k$ , then  $B$  uses this  $k$  to determine whether  $O$  is a true random function  $\rho$  or a PRF instance  $f(k, \cdot)$ , by choosing an unqueried point  $z \in \mathcal{X}$  and see if

$$f(k, z) = O(z).$$

If  $O(\cdot) = f(k, \cdot)$  and if  $A$  correctly found the key  $k$ , then the above equation holds. On the other hand, if  $O = \rho$ , then  $O(z)$  is uniform random, independent from  $f(k, z)$ , so the probability that  $f(k, z) = O(z)$  is only  $1/|\mathcal{Y}|$ . This difference in probability can be used to distinguish the two cases.

On the other hand, if  $A$  is a quantum adversary, the case when  $O(\cdot) = f(k, \cdot)$  is essentially the same as in the classical case. However, we may not apply the classical argument when  $O$  is a truly random function since the notion of “unqueried” point no longer makes sense under quantum (e.g., uniform superposition) queries. Therefore, we need a different approach in the quantum world. We defer the proof of Theorem 4.1 to the full version of this paper [15], due to page limitation. Note that it is possible to employ a coarse counting argument to prove key-recovery security, which works against both classical and quantum attacks. But it relies on specific settings of keyspace, domain and codomain, and the bound is typically not as tight as what our strategy can prove.

## 4.2 Security of PRFs Under Random Leakage

We show next that random leakage of the PRF key does not compromise the security of a PRF.

**Theorem 4.2.** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a secure PRF, with  $\mathcal{X}$  and  $\mathcal{Y}$  superpolynomially large. Then,  $f$  is also secure under random leakage.*

*Concretely, for any adversary  $A^{O, H}$  making at most  $q$  queries to  $O$  and  $q_H$  queries to  $H$ , we can construct adversaries  $A_{\text{kr}}$  and  $A_{\text{d}}$  such that*

$$\text{Adv}_f^{\text{prf-rl}}(A) \leq 2q_H \sqrt{\text{Adv}_f^{\text{prf-kr}}(A_{\text{kr}})} + \text{Adv}_f^{\text{prf}}(A_{\text{d}}).$$

*Here, both  $A_{\text{kr}}$  and  $A_{\text{d}}$  make at most  $q$  oracle queries.*

To prove Theorem 4.2, we are going to use the following lemma of Unruh.

**Lemma 4.3 (One-Way to Hiding Lemma of [17]).** *Let  $H : \mathcal{X} \rightarrow \mathcal{Y}$  be a random oracle. Consider an adversary  $A$  making at most  $q$  queries to  $H$ . Let  $B$  be an adversary that on input  $x$  does the following: pick  $i \leftarrow \{1, \dots, q\}$  and  $y \leftarrow \mathcal{Y}$ , run  $A^H(x, y)$  until (just before) the  $i$ th query, then measure the  $i$ th query in the computational basis, and output the outcome. (When  $A$  makes less than  $i$  queries,  $B$  outputs  $\perp \notin \mathcal{X}$ .) Then, we have*

$$\left| \Pr_{x \leftarrow \mathcal{X}}[A^H(x, H(x)) = 1] - \Pr_{\substack{x \leftarrow \mathcal{X} \\ y \leftarrow \mathcal{Y}}}[A^H(x, y) = 1] \right| \leq 2q \sqrt{\Pr_{x \leftarrow \mathcal{X}}[B^H(x) = x]}.$$

Now, we are ready to prove Theorem 4.2:

*Proof.* Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a secure PRF. To show that  $f$  is secure under random leakage, we need to show that for any set  $\mathcal{W}$  and any adversary  $A$ , the advantage  $\mathbf{Adv}_f^{\text{prf-rl}}(A)$  is negligible. Suppose  $A$  makes at most  $q$  queries to  $O$ , and at most  $q_H$  queries to  $H$ . Then,

$$\begin{aligned} \mathbf{Adv}_f^{\text{prf-rl}}(A) &= \left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{\rho, H}(w) = 1] \right| \\ &\leq \left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{f(k, \cdot), H}(w) = 1] \right| \\ &\quad + \left| \Pr[A^{f(k, \cdot), H}(w) = 1] - \Pr[A^{\rho, H}(w) = 1] \right|. \end{aligned}$$

It suffices to bound both terms. First, let us bound

$$\left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{f(k, \cdot), H}(w) = 1] \right|.$$

Let us define the algorithm  $A_1^H(k, w)$  as follows: it runs  $A^{O, H}(w)$  while any  $H$ -query is answered by  $H$ -query of  $A_1$  itself, and any  $O$ -query  $|x\rangle$  is answered by  $|x\rangle|f(k, x)\rangle$ . And when  $A^{O, H}(w)$  eventually halts with an output  $v$ ,  $A_1^H(k, w)$  outputs  $v$  and halts.

So,

$$\begin{aligned} A_1^H(k, H(k)) &= A^{f(k, \cdot), H}(H(k)), \\ A_1^H(k, w) &= A^{f(k, \cdot), H}(w). \end{aligned}$$

From Lemma 4.3, we have

$$\begin{aligned} &\left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{f(k, \cdot), H}(w) = 1] \right| \\ &= \left| \Pr[A_1^H(k, H(k)) = 1] - \Pr[A_1^H(k, w) = 1] \right| \\ &\leq 2q_H \sqrt{\Pr[B_1^H(k) = k]}, \end{aligned}$$

where the algorithm  $B_1^H(k)$  can be described as follows:  $B_1$  picks  $i \leftarrow \{1, \dots, q_H\}$ ,  $w \leftarrow \mathcal{W}$ , and runs  $A_1^H(k, w) = A^{f(k, \cdot), H}(w)$  until the  $i$ th  $H$ -query, then measure the  $i$ th query and output the outcome.

Now, using  $A$ , we construct an adversary  $A_{\text{kr}}$  mounting key recovery attack on  $f$ . The algorithm  $A_{\text{kr}}$  has oracle access to  $f(k, \cdot)$  for uniform random  $k \leftarrow \mathcal{K}$ , and  $A_{\text{kr}}$  works as follows:  $A_{\text{kr}}$  picks  $i \leftarrow \{1, \dots, q_H\}$ ,  $w \leftarrow \mathcal{W}$ , and runs  $A^{f(k, \cdot), H}(w)$ , while implementing  $H : \mathcal{K} \rightarrow \mathcal{W}$  by a  $2q_H$ -wise independent function, until the  $i$ th  $H$ -query, then measure the  $i$ th query and output the outcome.

By construction, we have  $\Pr[A_{\text{kr}}^{f(k, \cdot)}() = k] = \Pr[B_1^H(k) = k]$ . Therefore,

$$\left| \Pr[A^{f(k, \cdot), H}(H(k)) = 1] - \Pr[A^{f(k, \cdot), H}(w) = 1] \right| \leq 2q_H \sqrt{\mathbf{Adv}_f^{\text{prf-kr}}(A_{\text{kr}})}.$$

Note that the adversary  $A_{\text{kr}}$  makes at most  $q$  queries to its oracle  $f(k, \cdot)$ .



Next, let us bound

$$\left| \Pr[A^{f(k,\cdot),H}(w) = 1] - \Pr[A^{\rho,H}(w) = 1] \right|.$$

This is straightforward: using  $A$ , we construct an adversary  $A_d$  attacking PRF security of  $f$ . The algorithm  $A_d$  has oracle access to  $O$ , which can be  $f(k, \cdot)$  or a true random function  $\rho$ . Now, the algorithm  $A_d$  works as follows:  $A_d$  picks  $w \leftarrow \mathcal{W}$ , and runs  $A^{O,H}(w)$ , answering any  $O$ -query of  $A$  by an  $O$ -query of itself, and implementing  $H : \mathcal{K} \rightarrow \mathcal{W}$  by a  $2q_H$ -wise independent function. When  $A$  halts and outputs a value  $v$  eventually,  $A_d$  also halts and outputs  $v$ .

By construction, we have

$$\begin{aligned} & \left| \Pr[A^{f(k,\cdot),H}(w) = 1] - \Pr[A^{\rho,H}(w) = 1] \right| \\ &= \left| \Pr[A_d^{f(k,\cdot)}() = 1] - \Pr[A_d^{\rho}() = 1] \right| = \text{Adv}_f^{\text{prf}}(A_d). \end{aligned}$$

The adversary  $A_d$  also makes at most  $q$  queries. This proves the theorem.  $\square$

### 4.3 Oracle-Secure PRF Under Random Leakage

In order to prove security of NMAC, we are going to use the notion of oracle security under random leakage, which we define as follows.

**Definition 4.4 (Oracle security under random leakage).** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a PRF. We say that  $f$  is oracle-secure under random leakage, if for any sets  $\mathcal{W}, \mathcal{Z}$ , the following holds for any adversary  $A$ :*

$$\text{Adv}_{f,\mathcal{Z},\mathcal{W}}^{\text{os-rl}}(A) := \left| \Pr[A^{O_0,H}() = 1] - \Pr[A^{O_1,H}() = 1] \right| = \text{negl}()$$

where the oracles  $O_0, O_1$  are defined as

$$\begin{aligned} O_0(z, x) &:= (H(\kappa(z)), f(\kappa(z), x)), \\ O_1(z, x) &:= (\rho_1(z), \rho_2(z, x)), \end{aligned}$$

and  $H \leftarrow \mathcal{W}^{\mathcal{K}}, \kappa \leftarrow \mathcal{K}^{\mathcal{Z}}, \rho_1 \leftarrow \mathcal{W}^{\mathcal{Z}}, \rho_2 \leftarrow \mathcal{Y}^{\mathcal{Z} \times \mathcal{X}}$  are chosen uniform randomly, and independently.

We can show that any secure PRF  $f$  is also oracle-secure under random leakage:

**Theorem 4.5.** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a secure PRF, with  $\mathcal{X}$  and  $\mathcal{Y}$  superpolynomially large. Then,  $f$  is also oracle-secure under random leakage.*

*Concretely, for any adversary  $A^{O,H}$  making at most  $q$  queries to  $O$  and at most  $q_H$  queries to  $H$ , we can construct an adversary  $A_{\text{rl}}$  such that*

$$\text{Adv}_{f,\mathcal{Z},\mathcal{W}}^{\text{os-rl}}(A) < 12q^{3/2} \sqrt{\text{Adv}_f^{\text{prf-rl}}(A_{\text{rl}})},$$

where  $A_{\text{rl}}^{O,H}$  makes at most  $2q$  queries to  $O$ , and  $q_H + 2q$  queries to  $H$ .

*Proof.* Consider the distribution  $\text{PRFRL}_f$  over  $(\mathcal{W} \times \mathcal{Y})^{\mathcal{X}}$  which is efficiently samplable relative to  $H$ : the randomness space  $\mathcal{R}$  is just the key space  $\mathcal{K}$  of  $f$ , and the evaluation algorithm is given by  $\text{PRFRL}_f.\text{eval}^H(k, x) := (H(k), f(k, x))$ .

Consider another distribution  $\text{RU}$  over  $(\mathcal{W} \times \mathcal{Y})^{\mathcal{X}}$  of  $f$  defined by  $f(x) := (w, \rho(x))$ , where  $w \leftarrow \mathcal{W}$  and  $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$  are chosen uniform randomly and independently.

It is clear that the oracle security of  $f$  under random leakage is merely restatement of the oracle indistinguishability of  $\text{PRFRL}_f$  and  $\text{RU}$  relative to  $H$ .

Since  $f$  is secure, we may use Theorem 4.2 to show that  $f$  is secure under random leakage, and this is equivalent to indistinguishability of  $\text{PRFRL}_f$  and  $\text{RU}$  relative to  $H$ . Therefore, we are going to use Theorem 3.3 to show that the two are oracle-indistinguishable relative to  $H$ .

By construction,  $\text{PRFRL}_f$  and  $\text{RU}$  are independent, and since  $\text{PRFRL}_f$  is efficiently samplable relative to  $H$ ,  $\text{PRFRL}_f^{\mathcal{Z}}$  is bounded samplable relative to  $H$  for any set  $\mathcal{Z}$ , due to Lemma 2.18. Then, the only thing remaining to be proved to invoke Theorem 3.3 is that  $\text{RU}^{\mathcal{Z}}$  is bounded samplable for any set  $\mathcal{Z}$ . But this is to prove that the distribution of the oracle  $O_1(z, x) = (\rho_1(z), \rho_2(z, x))$  is bounded samplable, which is now trivially true.

Concretely, for any adversary  $A$  attacking oracle security under random leakage of  $f$  making at most  $q$  queries to  $O$  and  $q_H$  queries to  $H$ , by Theorem 3.3, we have

$$\begin{aligned} \mathbf{Adv}_{f, \mathcal{Z}, \mathcal{W}}^{\text{os-rl}}(A) &= \mathbf{Adv}_{\text{PRFRL}_f, \text{RU}, \mathcal{Z}, H}^{\text{oracle-rel-dist}}(A) \\ &< 12q^{3/2} \sqrt{\mathbf{Adv}_{\text{PRFRL}_f, \text{RU}, H}^{\text{rel-dist}}(A_{\text{rd}})}, \end{aligned}$$

for some adversary  $A_{\text{rd}}^{O', H}$  attacking indistinguishability of  $\text{PRFRL}_f$  and  $\text{RU}$  relative to  $H$ , which makes at most  $2q$  queries to  $O'$  and  $q_H + 2(1 + 0)q$  queries to  $H$ , since 1 call to  $H$  is required to implement  $\text{PRFRL}_f^{\mathcal{Z}}$ , and 0 calls to  $H$  are required to implement  $\text{RU}^{\mathcal{Z}}$ .

Now we can trivially turn  $A_{\text{rd}}$  into  $A_{\text{rl}}$  attacking security of  $f$  under random leakage:  $A_{\text{rl}}^{O, H}(w) := A_{\text{rd}}^{w \| O', H}()$ , satisfying  $\mathbf{Adv}_{\text{PRFRL}_f, \text{RU}, H}^{\text{rel-dist}}(A_{\text{rd}}) = \mathbf{Adv}_f^{\text{prf-rl}}(A_{\text{rl}})$ . Like  $A_{\text{rd}}$ ,  $A_{\text{rl}}^{O, H}$  makes at most  $2q$  queries to  $O$  and  $q_H + 2q$  queries to  $H$ .  $\square$

## 5 Security of NMAC and Other Constructions

In this section, we prove the PRF security of cascade, NMAC, HMAC, augmented cascade, and AMAC, using ingredients we have developed so far.

### 5.1 Security of the Cascade

The cascade construction is not secure when queries of different block lengths are allowed. However, if we fix the total number  $l$  of blocks for all messages, then it becomes a quantum-secure PRF. Since its proof is a simpler version of that of NMAC, we only state the theorem below and refer the readers to the proof of NMAC in Sect. 5.2.

**Theorem 5.1 (Security of the cascade construction).** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  be a secure PRF. Then,  $\text{Casc}_l[f]$  is a secure PRF, for any fixed  $l$ .*

*Concretely, for any adversary  $A$  of  $\text{Casc}_l[f]$  making at most  $q$  oracle queries, we can construct an adversary  $A_d$  making at most  $4q$  oracle queries, such that*

$$\text{Adv}_{\text{Casc}_l[f]}^{\text{prf}}(A) \leq 34lq^{3/2} \sqrt{\text{Adv}_f^{\text{prf}}(A_d)}.$$

## 5.2 Security of NMAC

We are now ready to prove the security of NMAC as a quantum PRF.

**Theorem 5.2 (NMAC security).** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  be a secure PRF. Then,  $\text{NMAC}[f]$  is a secure PRF.*

*Concretely, for any adversary  $A$  of  $\text{NMAC}[f]$  making at most  $q$  oracle queries, where each message has at most  $l$  message blocks, we can construct adversaries  $A_d$ ,  $A_{rl}$ , such that*

$$\text{Adv}_{\text{NMAC}[f]}^{\text{prf}}(A) \leq \text{Adv}_f^{\text{prf}}(A_d) + 34(l+1)q^{3/2} \sqrt{\text{Adv}_f^{\text{prf-rl}}(A_{rl})}.$$

*Moreover,  $A_d^O$  makes at most  $q$  queries to  $O$ , and  $A_{rl}^{O,H}$  makes at most  $4q$  queries to  $O$ , and at most  $6q$  queries to  $H$ .*

*Proof.* Let  $A$  be an adversary making at most  $q$  oracle queries, where each message has at most  $l$  message blocks. We are going to define a sequence of games, where in each game,  $A$  has access to an oracle  $O$ . The only difference between the games is how the oracle  $O$  is defined.

Here's our first game  $N$ .

**Game  $N$**  : In this game, the oracle  $O$  is given exactly as  $\text{NMAC}[f]$ : first,  $k_1, k_2 \leftarrow \mathcal{K}$  are picked uniform randomly and independently. Then, for any message  $x_1 \dots x_j$  of  $j$ -blocks ( $j = 0, 1, \dots, l$ ), the oracle  $O$  is defined as

$$O(x_1 \dots x_j) = f(k_2, \text{pad}(f(\dots f(f(k_1, x_1), x_2), \dots, x_j))).$$

In the next game  $G_0$ , the outer instance of the PRF  $f$  is swapped with a random function  $H : \mathcal{K} \rightarrow \mathcal{K}$ .

**Game  $G_0$** : In this game, the oracle  $O$  is given as follows: first,  $k \leftarrow \mathcal{K}$ ,  $H \leftarrow \mathcal{K}^{\mathcal{K}}$  are picked uniform randomly and independently. Then, for any message  $x_1 \dots x_j$  of  $j$ -blocks ( $j = 0, 1, \dots, l$ ), the oracle  $O$  is defined as

$$O(x_1 \dots x_j) = H(f(\dots f(f(k, x_1), x_2), \dots, x_j)).$$

Continuing, for each  $i = 1, \dots, l+1$ , we define games  $G_i$  as follows.

**Game  $G_i$** : In this game, the oracle  $O$  is given as follows: first,  $H \leftarrow \mathcal{K}^{\mathcal{K}}$ ,  $R \leftarrow \mathcal{K}^{\mathcal{X}^i}$ ,  $R_j \leftarrow \mathcal{K}^{\mathcal{X}^j}$  (for  $j = 0, \dots, i-1$ ) are picked uniform randomly and

independently. Then, for any message  $x_1 \dots x_j$  of  $j$ -blocks ( $j = 0, 1, \dots, l$ ), the oracle  $O$  is defined as

$$O(x_1 \dots x_j) = \begin{cases} R_j(x_1 \dots x_j) & \text{if } j = 0, \dots, i-1, \\ H(R(x_1 \dots x_i)) & \text{if } j = i, \\ H(f(\dots f(R(x_1 \dots x_i), x_{i+1}), \dots, x_j)) & \text{if } j = i+1, \dots, l. \end{cases}$$

Note that the game  $G_0$  is in fact a special case of the above games  $G_i$ ; when  $i = 0$ , the definition of  $O$  in the game  $G_i$  degenerates to

$$\begin{aligned} O() &= H(R()), \\ O(x_1 \dots x_j) &= H(f(\dots f(R(), x_1), \dots, x_j)), \end{aligned}$$

where  $k = R() \in \mathcal{K}$  serves as the secret key in the game  $G_0$ .

Also, let's take a special look at the final game  $G_{l+1}$ : we have

$$\begin{aligned} O() &= R_0(), \\ O(x_1) &= R_1(x_1), \\ &\vdots \\ O(x_1 \dots x_l) &= R_l(x_1 \dots x_l). \end{aligned}$$

Therefore, in the game  $G_{l+1}$ , the oracle  $O$  is a true random function defined over the domain  $\bigcup_{i=0}^l \mathcal{X}^i$ .

For any game  $G$  and an adversary  $A$ , let  $G(A)$  be the final output of  $A$  when  $A$  is executed in the game  $G$ . We see that

$$\begin{aligned} \mathbf{Adv}_{\text{NMAC}[f]}^{\text{prf}}(A) &= |\Pr[N(A) = 1] - \Pr[G_{l+1}(A) = 1]| \\ &\leq |\Pr[N(A) = 1] - \Pr[G_0(A) = 1]| \\ &\quad + |\Pr[G_0(A) = 1] - \Pr[G_{l+1}(A) = 1]|. \end{aligned}$$

First, it is easy to see that

$$|\Pr[N(A) = 1] - \Pr[G_0(A) = 1]| \leq \mathbf{Adv}_f^{\text{prf}}(A_d),$$

for some adversary  $A_d$  attacking the PRF security of  $f$ ; we can construct the adversary  $A_d^O$  distinguishing  $f(k, \cdot)$  and  $\rho \leftarrow \mathcal{K}^{\mathcal{X}}$  as follows: the adversary  $A_d^O$  picks  $k' \leftarrow \mathcal{K}$ , and runs  $A$ . For any query  $x_1 \dots x_j$  of  $A$  (for  $j \leq l$ ), return

$$O(\text{pad}(f(\dots f(f(k', x_1), x_2), \dots, x_j))).$$

When  $A$  eventually halts with an output  $v$ ,  $A_d$  also halts with  $v$ .

Now, when  $O(x) = f(k, x)$  for  $k \leftarrow \mathcal{K}$ , the query  $x_1 \dots x_j$  is answered by  $\text{NMAC}[f]$ , and when  $O = \rho \leftarrow \mathcal{K}^{\mathcal{X}}$ , then the function  $H(k) := O(\text{pad}(k))$  is a true random function uniformly random over  $\mathcal{K}^{\mathcal{K}}$ . In this case, the query of the adversary  $A$  is answered exactly like in the game  $G_0$ . In fact,

$$\mathbf{Adv}_f^{\text{prf}}(A_d) = |\Pr[N(A) = 1] - \Pr[G_0(A) = 1]|.$$

Next, we are going to construct an adversary  $A_{\text{os-rl}}$  attacking the oracle security of  $f$  under random leakage, with respect to the set  $\mathcal{X}^{l-1}$  and the random oracle  $H : \mathcal{K} \rightarrow \mathcal{K}$ . The adversary  $A_{\text{os-rl}}^{O',H}$  can be described as follows.

1.  $A_{\text{os-rl}}$  has access to two oracles  $O', H$ , where  $H : \mathcal{K} \rightarrow \mathcal{K}$  is a random oracle, and the oracle  $O' : \mathcal{X}^{l-1} \times \mathcal{X} \rightarrow \mathcal{K} \times \mathcal{K}$  is either  $O'_0(z, x) = (H(\kappa(z)), f(\kappa(z), x))$  or  $O'_1(z, x) = (\rho_1(z), \rho_2(z, x))$ , for uniform random and independent  $\kappa \leftarrow \mathcal{K}^{\mathcal{X}^{l-1}}$ ,  $\rho_1 \leftarrow \mathcal{K}^{\mathcal{X}^{l-1}}$ , and  $\rho_2 \leftarrow \mathcal{K}^{\mathcal{X}^{l-1} \times \mathcal{X}}$ . Let us parse  $O'$  into two parts and let  $O'(z, x) = (O^{(1)}(z), O^{(2)}(z, x))$ . Here,  $O^{(1)} : \mathcal{X}^{l-1} \rightarrow \mathcal{K}$  and  $O^{(2)} : \mathcal{X}^{l-1} \times \mathcal{X} \rightarrow \mathcal{K}$ .
2.  $A_{\text{os-rl}}$  picks a uniform random  $i \leftarrow \{0, \dots, l\}$ . Also,  $A_{\text{os-rl}}$  implements independent uniform random functions  $R_j \leftarrow \mathcal{K}^{\mathcal{X}^j}$  using bounded samplability, for  $j = 0, \dots, i-1$ .
3.  $A_{\text{os-rl}}$  runs the adversary  $A$  until it halts, while answering any query  $x_1 \dots x_j$  of  $A$  (for  $j = 0, 1, \dots, l$ ) as  $O(x_1 \dots x_j)$ , which is defined as follows:

$$O(x_1 \dots x_j) = \begin{cases} R_j(x_1 \dots x_j) & \text{if } j = 0, \dots, i-1, \\ O^{(1)}(\mathbf{0}^{l-i-1} x_1 \dots x_i) & \text{if } j = i, \\ H(f(\dots f(O^{(2)}(\mathbf{0}^{l-i-1} x_1 \dots x_i, x_{i+1}), x_{i+2}), \dots, x_j)) & \text{if } j = i+1, \dots, l. \end{cases}$$

In the above,  $\mathbf{0} \in \mathcal{X}$  is an arbitrarily fixed element of  $\mathcal{X}$ .

4. Eventually, when  $A$  halts with an output  $v$ ,  $A_{\text{os-rl}}$  also halts, outputting  $v$ .

We remark that  $A_{\text{os-rl}}$  makes at most two  $O'$ -queries and two  $H$ -queries to answer one query of  $A$  (for computing and uncomputing). Since  $A$  makes at most  $q$  oracle queries,  $A_{\text{os-rl}}$  makes at most  $2q$  queries to  $O'$  and  $2q$  queries to  $H$ .

Now, conditioned on the event that a specific  $i$  is chosen on line 2, if the oracle  $O'$  is given as  $O'_0(z, x) = (H(\kappa(z)), f(\kappa(z), x))$ , then the oracle  $O$  is given as follows:

$$O(x_1 \dots x_j) = \begin{cases} R_j(x_1 \dots x_j) & \text{if } j = 0, \dots, i-1, \\ H(\kappa(\mathbf{0}^{l-i-1} x_1 \dots x_i)) & \text{if } j = i, \\ H(f(\dots f(f(\kappa(\mathbf{0}^{l-i-1} x_1 \dots x_i), x_{i+1}), x_{i+2}), \dots, x_j)) & \text{if } j = i+1, \dots, l. \end{cases}$$

We see that this oracle is identically distributed as the oracle in game  $G_i$ .

On the other hand, if the oracle  $O'$  is given as  $O'_1(z, x) = (\rho_1(z), \rho_2(z, x))$ , then we have:

$$O(x_1 \dots x_j) = \begin{cases} R_j(x_1 \dots x_j) & \text{if } j = 0, \dots, i-1, \\ \rho_1(\mathbf{0}^{l-i-1} x_1 \dots x_i) & \text{if } j = i, \\ H(f(\dots f(\rho_2(\mathbf{0}^{l-i-1} x_1 \dots x_i, x_{i+1}), x_{i+2}), \dots, x_j)) & \text{if } j = i+1, \dots, l. \end{cases}$$

We see that this oracle is identically distributed as the oracle in game  $G_{i+1}$ .

Hence for each  $i$ , we have

$$\begin{aligned} & \Pr[A_{\text{os-rl}}^{O'_0, H}() = 1 \mid i] - \Pr[A_{\text{os-rl}}^{O'_1, H}() = 1 \mid i] \\ &= \Pr[G_i(A) = 1] - \Pr[G_{i+1}(A) = 1]. \end{aligned}$$

Therefore,

$$\begin{aligned} \text{Adv}_{f, \mathcal{X}^{l-1}, \mathcal{K}}^{\text{os-rl}}(A_{\text{os-rl}}) &= \left| \Pr[A_{\text{os-rl}}^{O'_0, H}() = 1] - \Pr[A_{\text{os-rl}}^{O'_1, H}() = 1] \right| \\ &= \left| \sum_{i=0}^l \left( \Pr[A_{\text{os-rl}}^{O'_0, H}() = 1 \mid i] - \Pr[A_{\text{os-rl}}^{O'_1, H}() = 1 \mid i] \right) \Pr[i] \right| \\ &= \frac{1}{l+1} \left| \sum_{i=0}^l \left( \Pr[A_{\text{os-rl}}^{O'_0, H}() = 1 \mid i] - \Pr[A_{\text{os-rl}}^{O'_1, H}() = 1 \mid i] \right) \right| \\ &= \frac{1}{l+1} \left| \sum_{i=0}^l (\Pr[G_i(A) = 1] - \Pr[G_{i+1}(A) = 1]) \right| \\ &= \frac{1}{l+1} |\Pr[G_0(A) = 1] - \Pr[G_{l+1}(A) = 1]|. \end{aligned}$$

We then get

$$|\Pr[G_0(A) = 1] - \Pr[G_{l+1}(A) = 1]| = (l+1) \text{Adv}_{f, \mathcal{X}^{l-1}, \mathcal{K}}^{\text{os-rl}}(A_{\text{os-rl}}).$$

Now, by Theorem 4.5, we have proved the theorem.  $\square$

### 5.3 Security of HMAC

Here let us briefly discuss the quantum security of HMAC. The security of HMAC is formally studied in [1]. There, the security of HMAC is reduced to the security of NMAC, with an additional assumption on the compression function  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$ . The assumption is that the ‘dual’ PRF of  $f$ , which is keyed by its data input as  $f(\cdot, K)$  for  $K \leftarrow \mathcal{X}$ , is a secure PRF against a minor related-key attack: when the key  $K$  is chosen, the adversary may query  $f(\cdot, K \oplus \text{ipad})$  and  $f(\cdot, K \oplus \text{opad})$ , and these two oracles should be indistinguishable from two independent random functions  $\rho_1, \rho_2 : \mathcal{K} \rightarrow \mathcal{K}$ . This reduction is still applicable to the quantum security, if we assume that the dual PRF of  $f$  is secure against the related-key attack. Hence, under this additional assumption, we can conclude that HMAC is a quantum-secure PRF.

*Remark 5.3.* In [12], Rötteler and Steinwandt showed that related-key attacks can be very powerful, when combined with the ability to make quantum superposed queries. Under a minor, reasonable assumption on the PRF  $f$ , if an oracle  $O(\delta, x) = f(k \oplus \delta, x)$  is given to an adversary who can make quantum superposed queries, then the secret key  $k$  can be efficiently recovered. Therefore, if a quantum adversary is allowed to derive keys by XORing an arbitrary constant  $\delta$ , then there exist essentially no quantum-secure PRFs against such an adversary.

However, in this case, we only need our dual PRF  $f$  to be *standard-secure* against this minor related-key attack, not quantum-secure: all we need is that the pair  $(f(\text{IV}, K \oplus \text{ipad}), f(\text{IV}, K \oplus \text{opad})) \in \mathcal{K}^2$  is indistinguishable from  $(k_1, k_2) \leftarrow \mathcal{K}^2$ , and for this we do not need quantum security. Hence, it is a reasonable assumption to make that  $f$  is secure in the above sense.

#### 5.4 Security of the Augmented Cascade and AMAC

We also show that the augmented cascade  $\text{ACSC}[f, \text{Out}]$  is quantum-secure. The proof is very similar to the security proof of NMAC, but unlike the case of NMAC, we need to assume that the PRF  $f$  is secure under **Out**-leakage. (The security proof of NMAC also uses similar security of  $f$  under random leakage, but this can be proved from the ordinary PRF security of  $f$ .)

First, let us give the following definition, which is similar to Definition 2.7.

**Definition 5.4 (Security of PRF under Out-leakage).** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$  be a PRF, and  $\text{Out} : \mathcal{Y} \rightarrow \mathcal{Z}$  be an unkeyed function. We say that  $f$  is secure under Out-leakage, if for any adversary  $A$ , we have the following:*

$$\text{Adv}_{f, \text{Out}}^{\text{prf-ol}}(A) := \left| \Pr[A^{f(k, \cdot)}(\text{Out}(k)) = 1] - \Pr[A^\rho(z) = 1] \right| = \text{negl}(),$$

where  $k \leftarrow \mathcal{K}, z \leftarrow \mathcal{Z}, \rho \leftarrow \mathcal{Y}^{\mathcal{X}}$  are uniformly and independently random.

*Remark 5.5.* Definition 5.4 is not exactly the same as the definition given in [2]. Their version, in our notation, would be negligibility of

$$\left| \Pr[A^{f(k, \cdot)}(\text{Out}(k)) = 1] - \Pr[A^\rho(\text{Out}(k)) = 1] \right|.$$

Definition 5.4 is, in fact, two claims combined in one: the first is that  $f(k, \cdot)$  remains pseudorandom even when  $\text{Out}(k)$  is leaked, and the second is that  $\text{Out}(k)$  itself is indistinguishable from a uniform random  $z \leftarrow \mathcal{Z}$ , for a uniform randomly chosen  $k \leftarrow \mathcal{K}$ . The definition in [2] is more general, but in order to obtain a PRF, eventually an output function  $\text{Out}$  close to regular should be selected, hence two definitions are essentially the same.

Now we may state the theorem showing that  $\text{ACSC}[f, \text{Out}]$  is quantum-secure.

**Theorem 5.6 (Quantum security of ACSC).** *Let  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$  be a PRF, and  $\text{Out} : \mathcal{K} \rightarrow \mathcal{Y}$  be an unkeyed function. Suppose that  $f$  is secure under Out-leakage. Then,  $\text{ACSC}[f, \text{Out}]$  is a secure PRF.*

*Concretely, for any adversary  $A$  of  $\text{ACSC}[f, \text{Out}]$  making at most  $q$  oracle queries, where each message has at most  $l$  message blocks, we can construct an adversary  $A_{\text{ol}}$  making at most  $4q$  queries, such that*

$$\text{Adv}_{\text{ACSC}[f, \text{Out}]}^{\text{prf}}(A) \leq 34(l+1)q^{3/2} \sqrt{\text{Adv}_{f, \text{Out}}^{\text{prf-ol}}(A_{\text{ol}})}.$$

In the proof, we first use oracle security of  $f$  under **Out**-leakage to carry out the hybrid argument, and then relate the oracle security to the PRF security under **Out**-leakage, again using Theorem 3.3. In fact, the proof is almost identical to that of Theorem 5.2, and we will omit the proof.

Similar to the security of HMAC, the security of AMAC follows directly from the security of ACSC, with an additional assumption on the compression function  $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{K}$ , namely, that the dual of  $f$ , that is,  $f(\cdot, K)$  for  $K \leftarrow \mathcal{X}$ , is a (standard-)secure PRF. This reduction is also applicable in the quantum security. Hence, with that additional assumption, we may conclude that AMAC is also quantum-secure.

**Acknowledgements.** We would like to thank the anonymous reviewers of Crypto 2017 for many helpful comments. The second author was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1601-07.

## References

1. Bellare, M.: New proofs for NMAC and HMAC: security without collision-resistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (2006). doi:[10.1007/11818175\\_36](https://doi.org/10.1007/11818175_36)
2. Bellare, M., Bernstein, D.J., Tessaro, S.: Hash-function based PRFs: AMAC and its multi-user security. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 566–595. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49890-3\\_22](https://doi.org/10.1007/978-3-662-49890-3_22)
3. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Kobitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996). doi:[10.1007/3-540-68697-5\\_1](https://doi.org/10.1007/3-540-68697-5_1)
4. Bellare, M., Canetti, R., Krawczyk, H.: Pseudorandom functions revisited: the cascade construction and its concrete security. In: FOCS 1996, pp. 514–523. IEEE Computer Society (1996)
5. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-38348-9\\_35](https://doi.org/10.1007/978-3-642-38348-9_35)
6. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-40084-1\\_21](https://doi.org/10.1007/978-3-642-40084-1_21)
7. Gaži, P., Pietrzak, K., Rybár, M.: The exact PRF-security of NMAC and HMAC. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 113–130. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-44371-2\\_7](https://doi.org/10.1007/978-3-662-44371-2_7)
8. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
9. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 207–237. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-53008-5\\_8](https://doi.org/10.1007/978-3-662-53008-5_8)
10. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: ISIT 2010, pp. 2682–2685. IEEE (2010)



11. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. In: ISITA 2012, pp. 312–316. IEEE (2012)
12. Rötteler, M., Steinwandt, R.: A note on quantum related-key attacks. *Inf. Process. Lett.* **115**(1), 40–44 (2015)
13. Santoli, T., Schaffner, C.: Using Simon’s algorithm to attack symmetric-key cryptographic primitives. *Quantum Inf. Comput.* **17**(1&2), 65–78 (2017)
14. Song, F.: A note on quantum security for post-quantum cryptography. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 246–265. Springer, Cham (2014). doi:[10.1007/978-3-319-11659-4\\_15](https://doi.org/10.1007/978-3-319-11659-4_15)
15. Song, F., Yun, A.: Quantum security of NMAC and related constructions. *Cryptology ePrint Archive*, Report 2017/509, full version of this paper (2017). <http://eprint.iacr.org/2017/509>
16. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-29011-4\\_10](https://doi.org/10.1007/978-3-642-29011-4_10)
17. Unruh, D.: Revocable quantum timed-release encryption. *J. ACM* **62**(6), 49:1–49:76 (2015)
18. Vadhan, S.P.: Pseudorandomness. *Foundations and trends® in theoretical computer science. Theoret. Comput. Sci.* **7**(1–3), 1–336 (2012)
19. Watrous, J.: Zero-knowledge against quantum attacks. *SIAM J. Comput.* **39**(1), 25–58 (2009)
20. Zhandry, M.: How to construct quantum random functions. In: FOCS 2012, pp. 679–687. IEEE Computer Society (2012)
21. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-32009-5\\_44](https://doi.org/10.1007/978-3-642-32009-5_44)

# Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields

Jean-François Biasse\*

Fang Song†

## Abstract

This paper gives polynomial time quantum algorithms for computing the ideal class group (CGP) under the Generalized Riemann Hypothesis and solving the principal ideal problem (PIP) in number fields of *arbitrary* degree. These are fundamental problems in number theory and they are connected to many unproven conjectures in both analytic and algebraic number theory. Previously the best known algorithms by Hallgren [20] only allowed to solve these problems in quantum polynomial time for number fields of *constant* degree. In a recent breakthrough, Eisenträger et al. [11] showed how to compute the unit group in arbitrary fields, thus opening the way to the resolution of CGP and PIP in the general case. For example, Biasse and Song [3] pointed out how to directly apply this result to solve PIP in classes of cyclotomic fields of arbitrary degree.

The methods we introduce in this paper run in quantum polynomial time in arbitrary classes of number fields. They can be applied to solve other problems in computational number theory as well including computing the ray class group and solving relative norm equations. They are also useful for ongoing cryptanalysis of cryptographic schemes based on ideal lattices [5, 10].

Our algorithms generalize the quantum algorithm for computing the (ordinary) unit group [11]. We first show that CGP and PIP reduce naturally to the computation of  $S$ -unit groups, which is another fundamental problem in number theory. Then we show an efficient quantum reduction from computing  $S$ -units to the continuous hidden subgroup problem introduced in [11]. This step is our main technical contribution, which involves careful analysis of the metrical properties of lattices to prove the correctness of the reduction. In addition, we show how to convert the output into an exact compact representation, which is convenient for further algebraic manipulations.

## 1 Introduction

Let  $K$  be a number field of degree  $n$  and  $\mathcal{O}$  be an order in  $K$  with discriminant  $\Delta$ . The ideal class

group  $\text{Cl}(\mathcal{O})$  is the finite abelian group consisting of the invertible fractional ideals of  $\mathcal{O}$  up to principal factors and has order  $|\Delta|^{O(1)}$ . Computing the ideal class group is an essential task in number theory that occurs in particular in the resolution of unproven heuristics such as the Cohen-Lenstra heuristics [9] on class groups of quadratic number field, Littlewood's bounds [25] on  $L(1, \chi)$ , or Bach's bound [1] on the maximum norm of the generators required to generate the class group. Besides being a fundamental problem, computing the ideal class group is also strongly connected to number theoretic problems occurring in cryptography. For example, it is at the heart of the only known unconditional classical subexponential algorithm for integer factorization [24]. Finding relations between elements in  $\text{Cl}(\mathcal{O})$  also occurs in curve-based cryptography. Indeed, both classical [4, 23] and quantum [6] subexponential methods for computing isogenies between elliptic curves depend on it.

Given an ideal  $\mathfrak{a} \subseteq \mathcal{O}$ , deciding whether or not  $\mathfrak{a}$  is principal, and if so, finding  $\alpha \in \mathcal{O}$  such that  $\mathfrak{a} = (\alpha)$  is called the Principal Ideal Problem. It has direct applications to the computation of relative class groups and unit groups, and computing the  $S$ -class group of a number field. It is also relevant to lattice-based cryptography, which has received a considerable attention since it allows quantum-safe cryptosystems and homomorphic encryption schemes. For efficiency reasons, there have been many proposals of schemes using lattices arising from ideals in the ring of integers of a number field, and in particular principal ideals generated by a small element (for example, see the homomorphic encryption scheme of Smart and Vercauteren [31] and the multilinear maps of Garg, Gentry and Halevi [18]). It has been recently shown that solving the principal ideal problem in polynomial time directly induces a polynomial time attack on schemes relying on the hardness of finding the short generator of a principal ideal [10].

Our method for finding the ideal class group of  $\mathcal{O}$  and solving the principal ideal problem in  $\mathcal{O}$  involves the computation of the  $S$ -unit group. Let  $S$  be a set of prime ideals of an order  $\mathcal{O}$  of  $K$ . The set of elements

\*Department of Mathematics and Statistics, University of South Florida, [biasse@usf.edu](mailto:biasse@usf.edu).

†Department of Combinatorics & Optimization and Institute for Quantum Computing, University of Waterloo, [fang.song@uwaterloo.ca](mailto:fang.song@uwaterloo.ca). Partially supported by Canada's NSERC, CIFAR, Government of Canada and ORF.

$\alpha \in K$  such that  $\exists (e_i)_{i \leq |S|} \in \mathbb{Z}^{|S|}$ ,  $(\alpha) = \mathfrak{p}^{e_1} \cdots \mathfrak{p}^{e_{|S|}}$  is a multiplicative group called the  $S$ -unit group of  $K$ . This notion generalizes the units of  $\mathcal{O}$  which are  $S$ -units for  $S = \emptyset$ , and computing the  $S$ -unit group is an important task in computational number theory. In particular, it is an essential ingredient of the resolution of norm equations of the form  $\mathcal{N}_{L/K}(x) = \theta$  where  $\theta \in K$ , as shown by Simon [30] and Fieker [15, 17].

**Previous work.** Computing the ideal class group and the unit group is a problem that has been extensively studied in both the classical and quantum setting. Despite these efforts, there are no known polynomial time algorithms for these tasks. On the other hand, there are quantum polynomial time algorithms for several hard computational problems in number theory based on quantum algorithms for the Hidden Subgroup Problem (HSP). Shor showed that integer factorization and the discrete logarithm problem could be solved in polynomial time [29], and Hallgren described a polynomial time algorithm for solving the Pell's equation [21]. Similar methods were used to compute the class group and the unit group in polynomial time in classes of number fields of fixed degree [20, 28]. The approach of [20] relies on the resolution of the HSP in a bounded and discretized approximation of  $\mathbb{R}^m$ , which does not seem to apply when the degree of the fields grows to infinity. In a recent breakthrough, Eisenträger, Hallgren, Kitaev and Song [11] described a polynomial time algorithm for computing the unit group in classes of number fields of arbitrary degree. One of the main tools they developed is a continuous HSP definition on  $\mathbb{R}^m$  and an efficient quantum algorithm solving it. In essence, their new HSP definition enforces stringent *continuity* properties on the function that hides the subgroup. This makes the function more amenable to quantum Fourier sampling.

**Our contribution.** In this paper, we present quantum algorithms to compute the ideal class group and solve the principal ideal problem in classes of number fields of arbitrary degree in polynomial time under the GRH. We follow a different framework than the previous work in constant-degree number fields due to Hallgren [20]. We show that both the ideal class group computation and PIP reduce to a more general problem of computing the  $S$ -unit group for suitable set of prime ideals  $S$ . For example, for the ideal class group computation  $S$  is chosen to be a succinct generating set of  $\text{Cl}(\mathcal{O})$ . Then we give an efficient quantum algorithm for computing the  $S$ -unit group by extending the work by Eisenträger, Hallgren, Kitaev and Song [11]. We show an efficient quantum reduction from the  $S$ -unit group problem to HSP on

$\mathbb{R}^m$  as defined in [11], which then can be solved efficiently by the quantum HSP algorithm in [11]. We also show how to get exact compact representations of the desired field elements with respect to a given integral basis for  $\mathcal{O}$ , while [11] only returns fixed point rational approximations of the units. Compact representations are usually easier for further algebraic processing. Our main results are summarized in the next three theorems.

**THEOREM 1.1. ( $S$ -UNIT GROUP COMPUTATION)**

*There is a quantum algorithm for computing the  $S$ -unit group of a number field  $K$  in compact representation which runs in polynomial time in the parameters  $n = \deg(K)$ ,  $\log(|\Delta|)$ ,  $|S|$  and  $\max_{\mathfrak{p} \in S} \{\log(\mathcal{N}(\mathfrak{p}))\}$ , where  $\Delta$  is the discriminant of the ring of integers of  $K$ .*

**THEOREM 1.2. (CLASS GROUP COMPUTATION)**

*Under the Generalized Riemann Hypothesis, there is a quantum algorithm for computing the class group of an order  $\mathcal{O}$  in a number field  $K$  which runs in polynomial time in the parameters  $n = \deg(K)$  and  $\log(|\Delta|)$ , where  $\Delta$  is the discriminant of  $\mathcal{O}$ .*

**THEOREM 1.3. (PIP RESOLUTION)** *There is a quantum algorithm for deciding if an ideal  $\mathfrak{a} \subseteq \mathcal{O}$  of an order  $\mathcal{O}$  in a number field  $K$  is principal, and for computing  $\alpha \in \mathcal{O}$  in compact representation such that  $\mathfrak{a} = (\alpha)$  which runs in polynomial time in the parameters  $n = \deg(K)$ ,  $\log(\mathcal{N}(\mathfrak{a}))$  and  $\log(|\Delta|)$ , where  $\Delta$  is the discriminant of  $\mathcal{O}$ .*

As an important corollary of our quantum algorithms, combining recent works in lattice cryptanalysis [5, 10], our results induce a quantum polynomial time attack on an entire family of cryptosystems relying on the hardness of finding a short generator of a principal ideal. See more in Sect. 6.

## 2 Preliminaries

In this section we review some useful background in number theory and introduce some definitions and notations. The notions of ideal class group and  $S$ -unit group are standard, and can be found in many books. We suggest Neukirch's book [27] for the fundamental aspects of this theory and Cohen's book [7] for the algorithmic aspects. We invite the reader who is already familiar to these topics to pay attention to the non-standard notion of  $E$ -ideal that we introduce in the following.

**Number fields.** A number field  $K$  is a finite extension of  $\mathbb{Q}$ . Its ring of integers  $\mathcal{O}_K$  has the structure of a  $\mathbb{Z}$ -lattice of degree  $n = [K : \mathbb{Q}]$ , and the orders  $\mathcal{O} \subseteq \mathcal{O}_K$  are the sublattices of  $\mathcal{O}_K$  which

have degree  $n$  and which are equipped with a ring structure. Throughout this paper, we assume that  $\mathcal{O}$  is an order in a number field  $K$ , and we denote by  $\omega_1, \dots, \omega_n$  a  $\mathbb{Z}$ -basis, that is  $\mathcal{O} = \mathbb{Z}\omega_1 \oplus \dots \oplus \mathbb{Z}\omega_n$ . A number field has  $n_1$  real embeddings and  $n_2$  pairs of complex embeddings which we denote  $(\sigma_j : K \rightarrow \mathbb{R})_{j \leq n_1}, ((\sigma_j, \overline{\sigma_j}) : K \rightarrow \mathbb{C})_{j \leq n_2}$  with  $n_1 + n_2 = n = \deg(K)$ . These embeddings define two essential maps, namely the norm and trace maps which are given by  $\mathcal{T}(x) := \sum_{\sigma} \sigma(x) \in \mathbb{Q}$  and  $\mathcal{N}(x) := \prod_{\sigma} \sigma(x) \in \mathbb{Q}$ . The trace map is additive while the norm map is multiplicative. Note that  $\mathcal{T}(\mathcal{O}) \subseteq \mathbb{Z}$  and  $\mathcal{N}(\mathcal{O}) \subseteq \mathbb{Z}$ . We measure the size of the ring  $\mathcal{O}$  by  $\log |\Delta|$  where  $\Delta := (\det(\sigma_j(\omega_k)))^2$  is its discriminant, and it equals the volume of the fundamental domain of  $\mathcal{O}$ . Equivalently, the discriminant can be defined from the trace map by  $\Delta := \det(\mathcal{T}(\omega_i \omega_j))_{i,j \leq n}$ .

**The ideal class group.** The fractional ideals of  $\mathcal{O}$  generalize the notion of ring ideals of  $\mathcal{O}$ . They are the subsets of  $K$  of the form  $\mathfrak{a} = \frac{1}{d}I$  where  $d \in \mathbb{Z}^+$  and  $I \subseteq \mathcal{O}$  is an (integral) ideal of  $\mathcal{O}$ . A fractional ideal  $\mathfrak{a}$  is invertible if  $\mathfrak{a}^{-1} := \{x \in K \mid x\mathfrak{a} \subseteq \mathcal{O}\}$  is also a fractional ideal. The invertible fractional ideals have a multiplicative group structure, and the principal fractional ideals are one of its subgroups. The ideal class group is defined by

$$\text{Cl}(\mathcal{O}) := \mathcal{I}/\mathcal{P},$$

where  $\mathcal{I}$  is the multiplicative group of fractional invertible ideals of  $\mathcal{O}$  and  $\mathcal{P}$  is the subgroup of elements of  $\mathcal{I}$  that are principal. This means that we identify  $\mathfrak{a}$  and  $\mathfrak{b}$  in  $\text{Cl}(\mathcal{O})$  if there is  $\alpha \in K$  such that  $\mathfrak{a} = (\alpha)\mathfrak{b}$ . Ideals are sublattices of  $\mathcal{O}$  of rank  $n$ , and we define their norm by  $\mathcal{N}(I) := |\mathcal{O}/I|$ . This notion naturally extends to fractional ideals using the multiplicative rule  $\mathcal{N}(\mathfrak{a}/\mathfrak{b}) := \mathcal{N}(\mathfrak{a})/\mathcal{N}(\mathfrak{b})$ . This notion of norm extends the norm on  $K$  in the sense that if  $\mathfrak{a} = (\alpha)$ , then  $\mathcal{N}(\mathfrak{a}) = \mathcal{N}(\alpha)$ .

**The  $S$ -unit group.** The  $S$ -units are a generalization of the units  $\mathcal{O}^*$ , which are the invertible elements of  $\mathcal{O}$ . The unit group can alternatively be defined as the  $\alpha \in \mathcal{O}$  with  $|\mathcal{N}(\alpha)| = 1$ , or the  $\alpha \in \mathcal{O}$  such that  $(\alpha) = \mathcal{O}$ . The unit group  $\mathcal{O}^*$  satisfies  $\mathcal{O}^* \simeq \mu \times \langle \varepsilon_1 \rangle \times \dots \times \langle \varepsilon_r \rangle$ , where  $r := n_1 + n_2 - 1$ ,  $\mu$  is the set of roots of unity and the  $\varepsilon_i$  are torsion-free units. Let  $S = \{\mathfrak{p}_i\}$  be a finite set of prime ideals of  $\mathcal{O}$ , the  $S$ -units are the elements  $\alpha \in K$  such that there is  $(e_i)_{i \leq |S|} \in \mathbb{Z}^{|S|}$  with  $(\alpha) = \mathfrak{p}^{e_1} \dots \mathfrak{p}^{e_{|S|}}$ . Note that the  $S$ -units are elements of  $K$ . They form a multiplicative group  $U(S)$  satisfying  $U(S) \simeq \mu \times \langle \varepsilon_1 \rangle \times \dots \times \langle \varepsilon_{r+|S|} \rangle$ , where  $r := n_1 + n_2 - 1$ ,  $\mu$  is the set of roots of unity and the  $\varepsilon_i$  are torsion-free  $S$ -units.

**$E$ -ideals.** The number field  $K$  can be naturally

embedded into  $E := \mathbb{R}^{n_1} \times \mathbb{C}^{n_2}$  by setting  $z \in \mathcal{O} \mapsto (\sigma_1(z), \dots, \sigma_{n_1+n_2}(z))$ . As in [11], we denote by  $\mathcal{Q}$  the image of  $\mathcal{O}$  via this embedding. The set  $\mathcal{Q}$  inherits from the lattice structure of  $\mathcal{O}$ , i.e. it can be identified as a lattice in  $\mathbb{R}^n$ , as well as from the multiplication between elements (which is performed component-wise). The image of the fractional ideals of  $K$  in  $E$  are lattices  $\Lambda \subseteq E$  with the property that  $x\Lambda \subseteq \Lambda$  for all  $x \in \mathcal{Q}$ . We define the  $E$ -ideals as all the lattices in  $E$  satisfying this property. When there is no ambiguity, we identify a fractional ideal of  $\mathcal{O}$  and the corresponding  $E$ -ideal.

**DEFINITION 2.1. ( $E$ -IDEALS)** Let  $E := \mathbb{R}^{n_1} \times \mathbb{C}^{n_2}$  and  $\mathcal{Q}$  the image of  $\mathcal{O}$  via the embedding  $K \rightarrow E$ . An  $E$ -ideal is a lattice  $\Lambda \subseteq E$  such that  $\forall x \in \mathcal{Q}, x\Lambda \subseteq \Lambda$ .

**Continuous HSP.** We review the definition of continuous HSP proposed by Eisenträger et al. [11], for which they have shown an efficient quantum algorithm.

**DEFINITION 2.2. (CONTINUOUS HSP OVER  $\mathbb{R}^m$ )**  
The unknown subgroup  $L \subseteq \mathbb{R}^m$  is a full-rank lattice satisfying some promise: the norm of the shortest vector is at least  $\lambda$  and the unit cell volume is at most  $d$ . The oracle has parameters  $(a, r, \varepsilon)$ . Let  $f : \mathbb{R}^m \rightarrow S$  be a function, where  $S$  is the set of unit vectors in some Hilbert space. We assume that  $f$  hides  $L$  in the following way.

1.  $f$  is periodic on  $L$ , i.e.  $f(x) = f(x + v)$  for all  $x \in \mathbb{R}^m$  and  $v \in L$ ;
2.  $f$  is Lipschitz with constant  $a$ , i.e.  $\|f(x) - f(y)\| \leq a\|x - y\|$  for all  $x, y \in \mathbb{R}^m$ ;
3. If the distance between the cosets  $(x \bmod L)$  and  $(y \bmod L)$  is greater or equal to  $r$ , i.e. if  $\min_{v \in L} \|x - y - v\| \geq r$ , then  $|\langle f(x) | f(y) \rangle| \leq \varepsilon$ .

Under these conditions, the problem is to compute a basis of  $L$  by a quantum algorithm that can make oracle calls  $|x\rangle \mapsto |x\rangle \otimes |f(x)\rangle$ .

Actually, the definition also applies more generally to other topological groups  $G = \mathbb{R}^k/\Lambda \times D$  with a proper metric on  $G$  [11, Sect.6.1]. Here  $G$  is decomposed to a continuous part, which is the quotient of  $\mathbb{R}^k$  over some lattice  $\Lambda$ , and a discrete part that is finitely generated. It is nonetheless sufficient to consider HSP on  $\mathbb{R}^m$ , because the more general case can be reduced to HSP on  $\mathbb{R}^m$  [11], and hence can be solved efficiently.

### 3 Overview of the algorithms

Our algorithms for CGP and PIP consist of reductions to the continuous hidden subgroup problem in two steps, and invoking the quantum HSP algorithm [11] at the end.

$$\begin{aligned}\text{CGP} &\leq_C S_{\text{CGP-units}} \leq_Q \text{HSP}(\mathbb{R}^{O(n)}), \\ \text{PIP} &\leq_Q S_{\text{PIP-units}} \leq_Q \text{HSP}(\mathbb{R}^{O(n)}).\end{aligned}$$

Specifically, we first reduce them to  $S$ -unit problems with proper choices of  $S$ , which are almost entirely *classical* except that we apply a quantum algorithm for factoring ideals in the case of PIP<sup>1</sup>. We describe these reductions to  $S$ -units problems in Sect. 4. Next we show a *quantum* reduction from  $S$ -units problem for any  $S$  to  $\text{HSP}(\mathbb{R}^m)$ , with  $m = O(|S|, n)$ . This is the main technical contribution of this work and it generalizes the reduction from (ordinary) unit-group problem to HSP by Eisenträger et al. [11]. The details will appear in Sect. 5, and we give an overview below.

Given  $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_k\}$ , we want to establish a function that hides the  $S$ -unit group according to Definition 2.2. To warm up, we review the reduction for the ordinary unit group (i.e.,  $S = \emptyset$ ) [11].

**Review: reduction for unit-group [11].** Observe that the unit group can be identified as a subgroup of  $G := \mathbb{R}^{n_1+n_2} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$ , and the mapping

$$\begin{aligned}\varphi: (u_1, \dots, u_{n_1+n_2}, \mu_1, \dots, \mu_{n_1}, \theta_1, \dots, \theta_{n_2}) \\ \mapsto (\dots, (-1)^{\mu_i} e^{u_i}, \dots, e^{2\pi i \theta_i} e^{u_i}, \dots).\end{aligned}$$

translates between the so-called *log coordinates* and the conjugate vector representation. To see this, note that under canonical embeddings, any  $z \in \mathcal{O}$  has the conjugate vector representation  $(\dots, \sigma_i(z), \dots) \in \mathbb{R}^{n_1} \times \mathbb{C}^{n_2}$ . If in addition  $z$  is invertible, then  $\sigma_i(z) \neq 0$ . Therefore, we can write  $\sigma_i(z) = (-1)^{\mu_i} e^{u_i}$  with  $\mu_i \in \mathbb{Z}_2$  and  $u_i \in \mathbb{R}$  if  $\sigma_i$  is real, or  $\sigma_i(z) = e^{2\pi i \theta_i} e^{u_i}$  with  $\theta_i \in \mathbb{R}/\mathbb{Z}$  and  $u_i \in \mathbb{R}$  if  $\sigma_i$  is complex.

Now one defines  $f$  in [11] as composition of two mappings:

$$f: G \xrightarrow{g} \{E\text{-ideals}\} \xrightarrow{f_q} \{\text{quantum states}\}.$$

Given  $x \in G$ ,  $g(x) := \varphi(x)\mathcal{O} \subseteq E$  produces an  $E$ -ideal which is a transformed lattice of  $\mathcal{O}$ . This is motivated by the fact that  $\alpha\mathcal{O} = \mathcal{O}$  for any unit  $\alpha \in \mathcal{O}^*$ . Actually, one can verify easily that  $g(x) = g(y)$  iff.  $\varphi(x - y) \in \mathcal{O}^*$ . Namely  $g$  is periodic on  $\mathcal{O}^*$ . For lacking of a canonical basis

<sup>1</sup>These reductions are straightforward. But classical algorithms typically compute the  $S$ -unit group by solving CGP and solving instances of PIP first. Our quantum algorithm tackles these problems in the reverse order.

to represent real-valued lattices uniquely, which is needed to apply the quantum HSP algorithm, a quantum mapping  $f_q$  follows. It encodes a lattice  $L$  into a quantum state  $|L\rangle$  that is roughly composed of quantum superposition over all lattice points, and hence provides a canonical representation for lattices. We will give more details of the quantum encoding in Sect. 5.1.

Very informally, one can show that small shift on an input to  $g$  causes small variance on the output lattice, but two inputs that are far apart modulo any unit will be mapped to lattices that have small overlap. Moreover,  $f_q$  preserves the “closeness” of lattices. Namely, quantum encodings of two lattices will have substantial inner product if and only if the lattices are very well lined up. To formalize these statements and thus proving the HSP properties, nonetheless, turn out to be highly non-trivial. It involves for example defining proper distance measures on various input and output spaces, and analyzing the continuity properties of  $f$  with respect to these metrics. This has been a great amount of efforts in [11] with further details in [12]

Other than these analytic properties, to make an efficient reduction, one needs to implement  $f = f_q \circ g$  efficiently. In fact,  $f_q$  can be implemented efficiently on a quantum computer by standard techniques. Computing  $g$ , on the other hand, is much more tricky. For instance  $e^{u_i}$  will involve doubly-exponential numbers if we manipulate them naively. Instead one splits the computation into small pieces, in the spirit of repeated squaring, and carefully controls the precision. There is one key observation that guarantees that the size of any intermediate step does not blow up. That is  $\mathcal{N}(z) = \pm 1$  for any unit  $z$  and hence  $\prod_{i=1}^{n_1} e^{u_i} \prod_{j=1}^{n_2} e^{2u_{n_1+j}} = 1$ . This indicates one redundant coordinate, and we can hence restrict  $f$  on  $\mathbb{R}^{n_1+n_2-1} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$  instead. This characterization is also essential to show a suitable bound on the volume of the unit cell of  $\mathcal{O}^*$ .

**Reducing  $S$ -units to HSP.** It is now easier to describe our generalized reduction for  $S$ -units. Let  $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_k\}$ . By definition, if  $\alpha \in \mathcal{O}$  is an  $S$ -unit, we have

$$\alpha \cdot \mathcal{O} \cdot \mathfrak{p}_1^{-v_{\mathfrak{p}_1}(\alpha)} \cdots \mathfrak{p}_k^{-v_{\mathfrak{p}_k}(\alpha)} = \mathcal{O},$$

where  $v_{\mathfrak{p}}(\alpha)$  is the coefficient of  $\mathfrak{p}$  in the power of  $(\alpha)\mathcal{O}$  (the valuation of  $\alpha$  at  $\mathfrak{p}$ ). Therefore the group of  $S$ -units  $U(S)$  is the subgroup of  $\hat{G} = \mathbb{R}^{n_1+n_2} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2} \times \mathbb{Z}^k$ . This motivates us defining the function  $\hat{g}: \hat{G} \rightarrow \{E\text{-ideals}\}$  by:

$$\hat{g}: (y, v_1, \dots, v_{|S|}) \mapsto \phi(y) \cdot \mathcal{O} \cdot \mathfrak{p}_1^{-v_1} \cdots \mathfrak{p}_{|S|}^{-v_{|S|}}.$$

We can show that (Prop. 5.1)  $\hat{g}$  is periodic on  $U(S)$ . We then apply the same quantum encoding  $f_q$  on the output of  $\hat{g}$ . Namely, our oracle function behaves like:

$$\hat{f} : \hat{G} \xrightarrow{\hat{g}} \{E\text{-ideals}\} \xrightarrow{f_q} \{\text{quantum states}\}.$$

While the classical mappings  $g$  and  $\hat{g}$  bear some similar motivation and we reuse  $f_q$ , to prove HSP properties of our function  $\hat{f}$  is far from straightforward. We need to define new metrics tailored to the specific group structure that the  $S$ -units belong and the  $E$ -ideals (lattices in  $\mathbb{R}^n$ ) that our  $\hat{g}$  may possibly generate. Then we show quantitatively that under these metrics, small variance in inputs induces slightly perturbed lattices, whereas large variance of inputs modulo any  $S$ -units will induce with high fraction of mismatch. Finally we relate the new metrics to the analysis of [11] and conclude the HSP properties. We further extend the function  $\hat{f}$  to obtain an HSP instance on  $\mathbb{R}^m$  and work out the necessary bounds  $(\lambda, d)$  as required, which allows us to invoke the quantum HSP algorithm to recover  $U(S)$ .

Again, efficient implementation of  $\hat{g}$  needs extra care. We need to split the computation differently due to the  $\prod \mathfrak{p}_j^{-v_j}$  part. It is also important to notice, similar to the unit-group case, that the  $S$ -unit group actually forms a subgroup of  $\mathbb{R}^{n_1+n_2-1} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2} \times \mathbb{Z}^k$ . That is, for a given  $S$ -unit  $\alpha$ , the information given by the  $|\sigma_j(\alpha)|$  for  $j \leq n_1 + n_2$  and  $v_{\mathfrak{p}}(\alpha)$  for  $\mathfrak{p} \in S$  is redundant. Indeed, if  $\alpha$  is an  $S$ -unit, then

$$\begin{aligned} \mathcal{N}(\alpha) \cdot \mathcal{N}\left(\prod_{j \leq |S|} \mathfrak{p}_j^{-v_{\mathfrak{p}_j}(\alpha)}\right) \\ = \prod_{j \leq n} e^{\log(\sigma_j(\alpha))} \cdot \prod_{j \leq |S|} e^{-v_{\mathfrak{p}_j}(\alpha) e_j \log(p_j)} = 1, \end{aligned}$$

where  $e_j \leq n$  and  $p_j$  are such that  $\mathcal{N}(\mathfrak{p}_j) = p_j^{e_j}$ , and where  $\log(x)$  denotes the natural logarithm of  $x$  (we use  $\log_2(x)$  for the base-2 logarithm). Therefore  $|\sigma_{n_1+n_2}(\alpha)|$  satisfies

$$\begin{aligned} \log(|\sigma_{n_1+n_2}(\alpha)|) &= -\frac{1}{2} \sum_{j \leq n_1} \log(|\sigma_j(\alpha)|) \\ &\quad - \sum_{n_1 < j < n_1+n_2} \log(|\sigma_j(\alpha)|) \\ &\quad + \frac{1}{2} \sum_{j \leq |S|} v_{\mathfrak{p}_j}(\alpha) e_j \log(p_j). \end{aligned}$$

More details will appear in Sect. 5.1. Note that the solution of HSP is given to us as approximations

of generators of the hidden subgroup. For many applications, an exact (and polynomially bounded) representation is preferable. Therefore, we process the solutions to the  $S$ -units problem classically to produce a compact representation of the generators of the  $S$ -unit group.

#### DEFINITION 3.1. (COMPACT REPRESENTATION)

Let  $l > 0$  be a constant, a compact representation of  $\alpha \in \mathcal{O}$  with respect to the integral basis  $(\omega_j)_{j \leq n}$  of  $\mathcal{O}$  is a set of exact representations of polynomial size algebraic numbers  $\gamma_j$  satisfying  $\alpha = \gamma_0 \gamma_1^l \cdots \gamma_k^l$ , where  $k$  is polynomial in the size of the input.

Recently, Biasse and Fieker [2, Sec. 5] described an efficient method based on [16, Alg. 7.53] to classically compute a compact representation of an algebraic number in polynomial time. These methods rely on the knowledge of an exact representation of the algebraic number we wish to represent (which is not the case here). A modification of [16, Alg. 7.53] using the approximation of the vector corresponding to an algebraic number yields a compact representation of that number. This extension uses well known lattice techniques and the details will be presented in the full version of this work.

#### 4 Reducing CGP and PIP to $S$ -units problem

As a motivating example, note that computing the (ordinary) unit-group problem reduces to  $S$ -units problem trivially by setting  $S$  to be the empty set. Next we show how to reduce CGP and PIP to computing  $S$ -units. There is an important observation about  $S$ -units that will be useful. Let  $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_k\}$  be a set of prime ideals and let  $U(S) = \mu_i \times \langle \varepsilon_1 \rangle \times \dots \times \langle \varepsilon_{r+k} \rangle$ . Each  $\varepsilon_i \in \mathcal{O}$  can be represented by  $(\mathbf{u}_i, v_{i,1}, \dots, v_{i,k})$  such that  $\mathbf{u}_i = (\sigma_1(\varepsilon_i), \dots, \sigma_{n_1+n_2}(\varepsilon_i))$ ,  $v_{i,j} = v_{\mathfrak{p}_j}(\varepsilon_i)$  for  $j = 1, \dots, k$ , and more importantly  $(\varepsilon_i) = \prod_{j=1}^k \mathfrak{p}_j^{v_{i,j}}$ . We define  $L(S) \subseteq \mathbb{Z}^k$  to be the lattice generated by  $\{(v_{i,1}, \dots, v_{i,k})\}_{i=1}^{r+k}$ . The following statement follows almost immediately from the definition of  $S$ -units.

LEMMA 4.1. Let  $U(S)$  and  $L(S)$  be as defined above. Let  $\mathfrak{a} := \prod_{i=1}^k \mathfrak{p}_i^{z_i}$  be an ideal with  $z_i \in \mathbb{Z}, i \in [k]$ . Then  $\mathfrak{a} = (\alpha)$  for some  $\alpha = \mu \prod_{i=1}^{r+k} \varepsilon_i^{x_i} \in U(S)$  with  $x_i \in \mathbb{Z}, i \in [r+k]$  iff.  $(z_1, \dots, z_k) \in L(S)$  with  $z_j = \sum_i x_i v_{i,j}$  for  $j \in [k]$ .

Proof. Suppose that  $\mathfrak{a} = \prod \mathfrak{p}_i^{z_i} = (\alpha)$  for some  $\alpha \in U(S)$ . Therefore  $\alpha = \prod \varepsilon_i^{x_i}$  for some  $x_i \in \mathbb{Z}, i = 1, \dots, r+k$ , and hence  $(\alpha) = \prod_i (\varepsilon_i)^{x_i}$ . Since  $(\varepsilon_i) = \prod_j \mathfrak{p}_j^{v_{i,j}}$ , we have that  $(\alpha) = \prod_i (\prod_j \mathfrak{p}_j^{v_{i,j}})^{x_i} = \prod_j \mathfrak{p}_j^{\sum_i x_i v_{i,j}}$ . By unique factorization of ideals,  $z_i =$

$\sum_i x_i v_{i,j}$  and hence  $(z_1, \dots, z_k) \in L(S)$ . Likewise, the exact same argument goes through in the reverse direction as well.  $\square$

**Class group problem.** To ensure an efficient reduction, we need a polynomial time generating set for the ideal class group. As pointed out in [2, Sec. 3.2], this directly derives from [1] (in the standard case where  $\mathcal{O} = \mathcal{O}_K$  the maximal order of  $K$ , the factor 48 can be replaced by 12).

**FACT 4.1.** *Let  $\mathcal{B} := \{\mathfrak{p} \subseteq \mathcal{O} \text{ prime} : \mathcal{N}(\mathfrak{p}) \leq 48 \log(|\Delta|)^2\}$  be the set of all prime ideals of  $\mathcal{O}$  of norm up to  $48 \log(|\Delta|)^2$ . Under the Generalized Riemann Hypothesis (GRH),  $\mathcal{B}$  generates  $\text{Cl}(\mathcal{O})$ , the size of  $\mathcal{B}$  is polynomial in  $\log(|\Delta|)$ , and can be computed in time polynomial in  $\log(|\Delta|)$ .*

Now let  $S_{\text{CGP}} = \mathcal{B} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_N\}$  as given in Fact 4.1. Consider the surjective morphism

$$\begin{array}{ccccc} \mathbb{Z}^N & \xrightarrow{\varphi} & \mathcal{I} & \xrightarrow{\pi} & \text{Cl}(\mathcal{O}) \\ (e_1, \dots, e_N) & \longrightarrow & \prod_i \mathfrak{p}_i^{e_i} & \longrightarrow & \prod_i [\mathfrak{p}_i]^{e_i}, \end{array}$$

and note that the class group  $\text{Cl}(\mathcal{O})$  is isomorphic to  $\mathbb{Z}^N / \ker(\pi \circ \varphi)$ . Therefore, computing the class group boils down to computing  $\ker(\pi \circ \varphi)$ , which consists of all  $(e_1, \dots, e_N)$  such that  $\prod \mathfrak{p}_i^{e_i}$  is a principal ideal. By Lemma 4.1,  $\ker(\pi \circ \varphi)$  is exactly  $L(S_{\text{CGP}})$ . As a result, the Smith Normal form of  $L(S_{\text{CGP}})$ , which can be computed efficiently [19, 26], will reveal the desired decomposition of  $\text{Cl}(\mathcal{O})$ . This is summarized in Algorithm 1.

---

**Algorithm 1** Reducing CGP to  $S$ -units

---

**Input:**  $\mathcal{O}$

- 1: Let  $S_{\text{CGP}} = \{\mathfrak{p} \subseteq \text{prime} \mid \mathcal{N}(\mathfrak{p}) \leq 48 \log(|\Delta|)^2\}$  with  $|S_{\text{CGP}}| = N$ .
  - 2: Compute a set of generators for the  $S_{\text{CGP}}$ -unit group  $U(S_{\text{CGP}})$ :  $\{(\mathbf{u}_i, v_{i,1}, \dots, v_{i,N})\}_{i=1}^{r+N}$ .
  - 3: Compute  $\text{diag}(d_1, \dots, d_n)$ , the Smith Normal Form of  $(v_{i,j})_{i \in [r+N], j \in [N]}$ .
  - 4: **return**  $d_1, \dots, d_n$ .
- 

**Principal ideal problem.** Given an ideal  $\mathfrak{a}$  by a  $\mathbb{Z}$ -basis, consider its prime factorization as  $\mathfrak{a} = \mathfrak{p}_1^{a_1} \cdots \mathfrak{p}_k^{a_k}$ , which can be obtained efficiently by adapting Shor's quantum factoring algorithm [29, 14]. Let  $S_{\text{PIP}} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_k\}$  be the prime divisors and clearly  $\mathfrak{a}$  is principal if and only if  $\mathfrak{a} = (\alpha)$  for an  $S_{\text{PIP}}$ -unit  $\alpha$ . By Lemma 4.1, this is equivalent to  $(a_1, \dots, a_k) \in L(S_{\text{PIP}})$ . Therefore, to decide if  $\mathfrak{a}$  is principal, it suffices to check  $(a_j) \in L(S_{\text{PIP}})$  which can be done efficiently by solving a linear system. If so, and suppose  $a_j = \sum_i x_i v_{i,j}$  with  $x_i \in \mathbb{Z}, i \in [r+k]$ ,

then  $\alpha := \prod_{i=1}^{r+k} \varepsilon_i^{x_i}$  gives a generator of  $\mathfrak{a}$ . The reduction is described in Algorithm 2.

---

**Algorithm 2** Reducing PIP to  $S$ -units

---

**Input:**  $\mathcal{O}$  and an ideal  $\mathfrak{a} \subseteq \mathcal{O}$ .

- 1: Factor  $\mathfrak{a} = \prod \mathfrak{p}_j^{a_j}$ . Let  $S_{\text{PIP}} = \{\mathfrak{p}_1, \dots, \mathfrak{p}_k\}$  be the divisors of  $\mathfrak{a}$ .
  - 2: Compute the  $S_{\text{PIP}}$ -unit group  $U(S_{\text{PIP}}) = \mu \times \langle \varepsilon_1 \rangle \times \cdots \times \langle \varepsilon_{r+k} \rangle$ . Note that  $\varepsilon_i = (\dots, v_{i,1}, \dots, v_{i,k})$  with  $\varepsilon_i = \prod_{j=1}^k \mathfrak{p}_j^{v_{i,j}}$ . Let  $M = (v_{i,j})$ .
  - 3: Solve for  $(x_1, \dots, x_{r+k})M = (a_1, \dots, a_k)$  with  $x_i \in \mathbb{Z}, i \in [r+k]$ .
  - 4: **return**  $\prod_i \varepsilon_i^{x_i}$  or “not principal” if the system has no solution.
- 

## 5 Reducing $S$ -units problem to continuous HSP

In this section, we show a quantum reduction from computing  $S$ -units to HSP for an arbitrary  $S$ . We define a function  $f$  in Sect. 5.1, which is periodic on the  $S$ -unit group. We also show that this function can be implemented efficiently on a quantum computer. Next we show in Sect. 5.2 that the function satisfies the conditions of the continuous HSP definition. In Sect. 5.3, we complete the remaining pieces of the reduction such as proving the geometric bounds of the  $S$ -unit group.

### 5.1 Defining the oracle function $(\mathbf{y}, \mathbf{v}) \mapsto |\varphi(\mathbf{y}) \mathcal{O} \prod_{\mathfrak{p} \in S} \mathfrak{p}^{-v_i}\rangle$

As we informally discussed in Sect. 3, we define  $f$  as<sup>2</sup>

$$f : G \xrightarrow{f_c} \{E\text{-ideals}\} \xrightarrow{f_q} \{\text{quantum states}\},$$

where  $G = \mathbb{R}^{n_1+n_2} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2} \times \mathbb{Z}^{|S|}$ . Specifically,  $f_c$  maps  $(\mathbf{y}, \mathbf{v}) \in G$  to a rational approximation of a basis for the  $E$ -ideal

$$(5.1) \quad f_c(y, v_1, \dots, v_{|S|}) = \phi(y) \cdot \mathcal{O} \mathfrak{p}_1^{-v_1} \cdots \mathfrak{p}_{|S|}^{-v_{|S|}}.$$

We show that  $f_c$  is periodic on  $U(S)$ .

**PROPOSITION 5.1.** *For any  $(y, (v_j))$  and  $(y', (v'_j))$ , let  $(u, (w_j)) = (y', (v'_j)) - (y, (v_j))$ . Then the function  $f_c$  satisfies that*

- $f_c(y', (v'_j)) = f(y, (v_j)) \iff \phi(u) \in U(S)$ .
- $v_{\mathfrak{p}_j}(\phi(u)) = w_j, \forall j = 1, \dots, |S|$ .

---

<sup>2</sup>Here we overload the notations of  $f$ ,  $G$ , and rewrite  $\hat{g}$  as  $f_c$  to emphasize that it is a classical function.

*Proof.* If  $\phi(u) \in U(S)$ ,  $f_c(u, (w_j)) = \mathcal{O}$  and  $f_c(y', (v'_j)) = f_c((y, (v_j)) + (u, (w_j)))$ . Reciprocally, if  $f_c(y', (v'_j)) = f_c((y, (v_j)) + (u, (w_j)))$ , then  $\phi(u) \cdot \mathcal{Q} \cdot \mathfrak{p}_1^{-v_1} \cdots \mathfrak{p}_{|S|}^{-v_{|S|}} = \mathcal{O}$ . In particular, there exist  $\alpha \in \prod_j \mathfrak{p}_j^{-v_j} \subseteq K$  and  $\beta \in \mathcal{O}$  such that  $\phi(u) = \beta/\alpha \in K$ . Therefore  $u \in K$  and has to be an  $S$ -unit.  $\square$

Note that the naive computation of  $f_c$  involves computing  $(e^{u_i})_{i \leq n_1+n_2}$ , where  $y = (u_1, \dots, n_{n_1+n_2}, \theta)$  with a phase  $\theta \in \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$ . Any rational approximation of  $e^{u_i}$  has at least  $\lceil \log_2(e^{u_i}) \rceil \in O(u_i)$  bits where  $\log_2$  denotes the base 2 logarithm. As this is exponential in the bit size of the entry, we need to proceed differently to evaluate  $f_c$ . The authors of [11] described a way to split up the computation ensuring that we only manipulate values of polynomial size. We adapt this method to our specific classical oracle that differs by a term of the form  $\prod_{\mathfrak{p}_i \in S} \mathfrak{p}_i^{-v_i}$  from the one described in [11].

**PROPOSITION 5.2.** *The methods of [11] can be adapted to evaluate  $e^{\mathcal{Q}} \prod_{\mathfrak{p} \in S} \mathfrak{p}^{-v_i}$  in a polynomial number of multiplications between  $E$ -ideals of determinant  $\sqrt{|\Delta|}$ .*

The arithmetic between  $E$ -ideals is directly inspired from the arithmetic between ideals in a number field. To evaluate our classical oracle, we need an efficient implementation of the  $E$ -ideal multiplication. Let  $A = \oplus_{j \leq n} \mathbb{Z}a_j$  and  $B = \oplus_{k \leq n} \mathbb{Z}b_k$  be  $E$ -ideals generated by the  $a_j, b_k \in E$ . Then the  $E$ -ideal  $A \cdot B$  is the lattice generated by the  $n^2$  elements  $(a_j \cdot b_k)_{j,k \leq n}$ . The multiplication of two  $E$ -ideals can be described by the two following steps:

1. Calculate all the cross terms  $a_j \cdot b_k$  for  $j, k \leq n$ .
2. Compute a basis  $(c_j)_{j \leq n}$  of  $\sum_{j,k} \mathbb{Z}a_j \cdot b_k$ .

The main challenge of  $E$ -ideal multiplication is that we need to deal with rational approximations of lattices. We need to estimate how much precision is needed to ensure accuracy, and how much precision is lost after each operation. Knowing how to bound the number of operation between  $E$ -ideals and the cost of each operation allows us to estimate the asymptotic complexity of the classical oracle.

**THEOREM 5.1.** *The  $E$ -ideal multiplication between  $E$ -ideals of determinant  $\sqrt{|\Delta|}$  requires a polynomial number of bits of precision and runs in polynomial time. The complexity of the classical oracle is in*

$$\tilde{O}\left(\|(y, v)\|^2 n^{5+\varepsilon} \left((n \log(|\Delta|) + n^2 + \|(y, v)\|^2)^{1+\varepsilon}\right) + |S| \max_j (\log(p_j)^3)\right),$$

where  $\varepsilon > 0$  is arbitrarily small.

**Quantum Encoding  $f_q$ .** Once we have obtained a basis for an  $E$ -ideal from  $f_c$ , we use the same quantum encoding proposed in [11] to encode the ideal (lattice) in a quantum state. This gives a (quantum) canonical way of representing real-valued lattices uniquely, which is needed later to apply the quantum algorithm for solving HSP. Here we give a brief review of the quantum encoding  $f_q$ .

Let  $g_s(\cdot)$  be the Gaussian function  $g_s(x) := e^{-\pi\|x\|^2/s^2}$ ,  $x \in \mathbb{R}^n$ . For any set  $S \subseteq \mathbb{R}^n$ , denote  $g_s(S) := \sum_{x \in S} g_s(x)$ . Given a lattice  $L$ , the quantum encoding maps  $L$  to the lattice Gaussian state via

$$f_q(L) = |L\rangle := \gamma \sum_{v \in L} g_s(v) |\text{str}_{\nu, n}(v)\rangle,$$

where  $\mathcal{S} = \{\text{unit vectors in a Hilbert space}\}$  and  $\gamma$  is a factor that normalized the state. Here  $|\text{str}_{\nu, n}(v)\rangle$  is the straddle encoding of a real-valued vector  $v \in \mathbb{R}^n$ , as defined in [11]. Intuitively, one discretizes the space  $\mathbb{R}^n$  by a grid  $\nu\mathbb{Z}^n$ , and encodes the information about  $v$  by a superposition over all grid nodes surrounding  $v$ . Specifically, for the one-dimensional case, the straddle encoding of a real number is

$$x \in \mathbb{R} \mapsto |\text{str}_{\nu}(x)\rangle := \cos\left(\frac{\pi}{2}t\right)|k\rangle + \sin\left(\frac{\pi}{2}t\right)|k+1\rangle,$$

where  $k := \lfloor x/\nu \rfloor$  denotes the nearest grid point no bigger than  $x$  and  $t := x/\nu - k$  denotes the (scaled) offset. Repeat this for each coordinate of  $v = (v_1, \dots, v_n)$  we get  $|\text{str}_{\nu, n}(v)\rangle := \bigotimes_{i=1}^n |\text{str}_{\nu}(v_i)\rangle$ .

**FACT 5.1.** ([11]) *Let  $L$  be an LLL-reduced basis. Assume that  $\lambda_1(L) \geq \lambda$ ,  $\det(L) \leq d$  and  $s \geq n^{n/2+1} 2^n \lambda^{-n+1} d$ . There is a quantum algorithm that takes  $L$  as input and produces a state that is  $2^{-\Omega(n)}$ -close to  $|L\rangle = \gamma \sum_{v \in L} g_s(v) |\text{str}_{\nu, n}(v)\rangle$  within time  $\text{poly}(n, \log s, \log \frac{1}{\nu})$ .*

## 5.2 Analyzing the HSP properties of $f$

In this section, we discuss the properties that the function  $f : G \rightarrow \mathcal{S}$  hiding  $U(S)$  needs to satisfy by rephrasing Definition 2.2 for the group we are interested in.

**DEFINITION 5.1. (HSP PROPERTY)** *We say that  $f : G \rightarrow \{\text{Quantum states}\}$  satisfies the HSP property for a discrete subgroup  $H \leq G$  if*

1.  $f$  is periodic on  $H$ , that is  $f(x+u) = f(x) \forall x \in G, u \in H$ ,



2.  $f$  is Lipschitz for some constant  $a : \forall x, y \in G/H, ||f(x) - f(y)|| \leq a \cdot d_{G/H}(x, y)$ ,
  3. There are  $r, \varepsilon > 0$  such that  $\forall x, y \in G/H$ , if  $d_{G/H}(x, y) \geq r$ , then  $|\langle f(x) | f(y) \rangle| \leq \varepsilon$ ,
- where  $d_{G/H}(\cdot, \cdot)$  denotes a distance on  $G/H$ .

Because the input includes valuations  $v_i$  of a power-product of prime ideals, our classical oracle significantly differs from the one used to hide the unit group in [11]. We need to define metrics on  $G$  and the set of  $E$ -ideals arising as the images of an element in  $G$ , together with a careful analysis of the topological properties of the oracle with respect to this metric.

**DEFINITION 5.2. (DISTANCE ON  $G/U(S)$ )** Let  $(z, (v_j)_{j \leq |S|})$  and  $(z', (v'_j)_{j \leq |S|})$ , we define their distance in  $G/U(S)$  by

$$\inf \left\{ \|a\| + \sum_j |w_j| e_j \log(p_j) \text{ such that } (z', (v'_j)) = (z, (v_j)) + (a, (w_j)) + u, u \in U(S) \right\},$$

where  $\|a\|$  is the Euclidean norm of the vector corresponding to  $a$  in  $\mathbb{R}^{n_1+n_2} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$  (note that we take the phase into account). The  $p_j, e_j$  are defined as  $\mathcal{N}(\mathfrak{p}_j) = p_j^{e_j}$ .

**DEFINITION 5.3. (DISTANCE BETWEEN  $E$ -IDEALS)** Let  $\mathcal{L}$  and  $\mathcal{L}'$  be two  $E$ -ideals arising as the image of elements in  $G$  by the classical encoding  $f_c$ , and  $L_\Delta := \mathcal{L}'/\mathcal{L}$ . We define

$$\text{dist}(\mathcal{L}, \mathcal{L}') := \inf \left\{ \|a\| + \sum_j \log(d_j) + n \log(d) \text{ such that } L_\Delta = e^{\text{diag}(a_j)} B_\omega \text{diag}(d_j/d) \right\},$$

where  $L_\Delta$  runs over all the matrices of a basis of  $\mathcal{L}'/\mathcal{L}$  such that there is a matrix  $B_\omega$  of an integral basis of  $\mathcal{O}$ ,  $d_j, d \in \mathbb{Z}_{>0}$ , and  $\|a\|$  is the Euclidean norm of the vector  $a \in \mathbb{R}^{n_1+n_2} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$  corresponding to  $(a_j)_{j \leq n} \in E$  satisfying  $L_\Delta = e^{\text{diag}(a_j)} B_\omega \text{diag}(d_j/d)$ .

**PROPOSITION 5.3.** Definition 5.3 defines a distances between lattices arising as the image of an element of  $G$  by the map (5.1).

Let  $G$  and  $f = f_q \circ f_c$  be as defined before, we are able to prove Theorem 5.2 and Theorem 5.3, which ensure that our oracle satisfies the HSP property.

**THEOREM 5.2.** There exists  $a > 0$  such that for any  $x, y \in G/U(S)$ ,  $||f(x) - f(y)|| \leq a \cdot \text{dist}_{G/U(S)}(x, y)$ .

**THEOREM 5.3.** There are  $r > 0$  and  $\varepsilon > 0$  such that  $d_{G/U(S)}(x, y) \geq r \Rightarrow |\langle f(x) | f(y) \rangle| \leq \varepsilon$ .

### 5.3 Completing the reduction

We have shown that the  $S$ -unit group corresponds to the periods of a function on  $\mathbb{R}^{n_1+n_2-1} \times \mathbb{Z}_2^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2} \times \mathbb{Z}^{|S|}$  satisfying the continuous HSP property. To invoke the algorithm described in [11], we need to reduce further to an instance of the continuous HSP on  $\mathbb{R}^m$  for some  $m > 0$ . This follows similar arguments as in [11, Sect.6.1]. A formal proof is deferred to the full version.

**THEOREM 5.4.** Let  $f : G = \mathbb{R}^{n_1+n_2-1} \times (\mathbb{R}/\mathbb{Z})^{n_2} \times \mathbb{Z}_2^{n_1} \times \mathbb{Z}^{|S|} \rightarrow \mathcal{S}$  be an  $(a, r, \varepsilon)$  oracle function that hides  $L$  with  $\lambda_1(L) \geq \lambda$  and  $\text{Vol}(G/L) \leq d$ . Then it reduces to an HSP instance  $g : \mathbb{R}^m \rightarrow \mathcal{S}$  that hides  $\tilde{L}$  with  $m = 2(n_1+n_2)+|S|-1$ ,  $\lambda_1(\tilde{L}) \geq \lambda$ ,  $\text{Vol}(\mathbb{R}^m/\tilde{L}) \leq d\lambda^{n_1}$  and parameters  $\varepsilon = \varepsilon$ ,

$$a = \sqrt{a^2 + |S|(\frac{\pi}{2\nu}(1+\nu))^2 + n_2(\frac{\pi}{2\nu\lambda}(1+\nu))^2},$$

$$r = \sqrt{(2r + 2|S|\nu)^2 + n_2(2\nu\lambda)^2}.$$

In addition  $g$  can be instantiated efficiently on a quantum computer with access to  $f$ .

According to Definition 2.2, we still need to derive bounds for the first minima and the fundamental volume of the lattice of  $S$ -units, which the complexity of the algorithm for solving the HSP depends on. We show such bounds by an analogue of Dirichlet's unit theorem.

**PROPOSITION 5.4.** The first minima of  $U(S) \subseteq G$  satisfies  $\lambda_1(U(S)) \geq \frac{\log(n)}{6n^4}$  where the norm on elements of  $G$  is defined by  $\|(z, v_1, \dots, v_{|S|})\| := \sqrt{\sum_j z_j^2 + \sum_j |v_j| e_j \log(p_j)}$ . Moreover, the volume of the lattice of  $S$ -units satisfies

$$\text{Vol}(G/U(S)) \leq \frac{1}{\log(2)^{|S|}} \left( 300 \log(P) \sqrt{|\Delta|} \left( \frac{e}{2} \log(|\Delta|) \right)^{n-1} \right)^{|S|+r-\frac{n}{2}},$$

where  $P = \max_j \mathcal{N}(\mathfrak{p}_j)$ .

We can now invoke the efficient quantum algorithm for HSP on  $\mathbb{R}^m$  proposed in [11, Theorem 6.1]. Pick a fine enough discrete grid  $\delta\mathbb{Z}^m$  and a sufficiently broad and smooth window function  $w$ , we create a superposition of grid points within the window, evaluate the function, and then measure the state in the Fourier basis. With sufficiently many samples, one obtains an approximate generating set of  $\mathcal{L}^*$ , from which one can compute a basis for  $\mathcal{L}$  as well within the desired precision.

## 6 Applications and Discussion

There are a few recent cryptosystems relying on the hardness of finding a short generator of a principal ideal (short-PIP) of the cyclotomic ring  $\mathbb{Z}[X]/(X^{2^n} + 1)$ . Typical examples of these schemes are the fully homomorphic encryption scheme of Smart and Vercauteren [31] and the multilinear maps of Garg, Gentry and Halevi [18]. Following an observation of CESC scientists Campbell et al. [5, Sec. 3], the short-PIP reduces to the PIP (a fact rigorously proved later by Cramer et al. [10]). The task of recovering an arbitrary generator of an ideal in  $\mathbb{Q}(\zeta_{2^n})$  (PIP under the promise that the ideal is principal) was conjectured to be feasible in quantum polynomial time by Campbell et al. [5], but the algorithm they proposed seems to have an exponential run time [3, Sec. 5]. Biasse and Song [3] later adapted the unit group algorithm of Eisenträger et al. [11] to derive a polynomial time solution to this task. However, the algorithm proposed in [3] is limited to cyclotomic fields and assumes a priori that the ideal is principal. Our algorithm for the PIP in arbitrary fields leaves the door open for further generalizations of the attacks against cryptosystems relying on the short-PIP in  $\mathbb{Q}(\zeta_{2^n})$  to other schemes using ideal lattices. In particular, there is currently a lot of attention around the possibility of reducing the NTRU problem [22] to an instance of the short-PIP in a quadratic extension of  $\mathbb{Q}(\zeta_{2^n})$ .

Our work also has direct applications in computational number theory. Indeed, the  $S$ -unit group is a central object that can be used in a lot of algorithms. It usually is computed together with the so-called  $S$ -class group, which is the quotient of the group of ideals in the ring of  $S$ -integers by the subgroup of principal ideals. The  $S$ -class group can easily be derived from the ideal class group and an oracle for the PIP by quotienting the class group by extra relations. A description of this method can be found in Simon's PhD thesis [30, Chap. 1].

Another consequence of our work is that it implies a polynomial time algorithm for computing the relative class group and the relative unit group of an arbitrary extension of number fields. Algorithms for these tasks are already known [8, Ch. 7], but their run time is exponential in the degree of the fields. As for the  $S$ -class group, they also consist of using a complete set of relations for the ideal class group and of enriching it with new relations that are obtained by solving instances of the PIP.

Our algorithms also directly imply a quantum algorithm for computing the ray class group of an arbitrary number field. The computation of the ray class group is an essential task in computational class

field theory. A classical method due to Cohen can be found in [8, 3.2] and has an exponential run time with respect to the degree (but runs in subexponential time for classes of fixed degree number fields). A quantum algorithm was described by Eisenträger and Hallgren [13] with a polynomial run time in classes of fixed degree number fields. As for the aforementioned tasks, computing the ray class group essentially relies on subroutines for computing the ideal class group and solving the PIP, for which we provide polynomial time algorithms in arbitrary number fields. It also relies on algorithms for factoring ideals and solving the discrete logarithm, both of which are easy on a quantum computer [29, 14].

Finally, our work allows us to describe polynomial time algorithms for solving relative norm equations of the form  $\mathcal{N}_{L/K}(x) = \theta$  where  $L/K$  is an arbitrary Galois extension. Norm equations are an important example of Diophantine equations which are a major topic in number theory. The resolution of the Pell's equation (for which there is a quantum algorithm [21]) can be seen as a special case where  $L = \mathbb{Q}(\sqrt{\Delta})$ ,  $K = \mathbb{Q}$  and  $\theta = 1$  (when we restrict our attention to integer solutions). Solving norm equations in general is an important task in computational number theory. A classical method was described by Simon [30] (based on the work of Fieker [15] for Galois extensions) that solves general extensions in exponential time in the degree of the fields. For the Galois case, it simply uses the knowledge of the  $S$ -unit group and the relative class group, which we can provide in polynomial time for number fields of arbitrary degree. However, the general method uses the Galois closure, whose degree can be exponential in the degree of the field, thus restricting the direct application of our work to arbitrary Galois extensions.

## References

- [1] E. Bach. Explicit bounds for primality testing and related problems. *Mathematics of Computation*, 55(191):355–380, 1990.
- [2] J.-F. Biasse and C. Fieker. Subexponential class group and unit group computation in large degree number fields. *LMS Journal of Computation and Mathematics*, 17:385–403, 1 2014.
- [3] J.-F. Biasse and F. Song. On the quantum attacks against schemes relying on the hardness of finding a short generator of an ideal in  $\mathbb{Q}(\zeta_{p^n})$ . <http://cacr.uwaterloo.ca/techreports/2015/cacr2015-12.pdf>, 2015.
- [4] R. Bröker, D. Xavier Charles, and K. Lauter. Evaluating large degree isogenies and applications to pairing based cryptography. In S. Galbraith and K. Paterson, editors, *Pairing-Based Cryptography - Pairing 2008, Second International Conference*,

- Egham, UK, September 1-3, 2008. *Proceedings*, Lecture Notes in Computer Science, pages 100–112. Springer, 2008.
- [5] P. Campbell, M. Groves, and D. Shepherd. SOLILOQUY, a cautionary tale. [http://docbox.etsi.org/Workshop/2014/201410\\_CRYPT0/S07\\_Systems\\_and\\_Attacks/S07\\_Groves\\_Annex.pdf](http://docbox.etsi.org/Workshop/2014/201410_CRYPT0/S07_Systems_and_Attacks/S07_Groves_Annex.pdf), 2014.
  - [6] A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2013.
  - [7] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, 1991.
  - [8] H. Cohen. *Advanced topics in computational algebraic number theory*, volume 193 of *Graduate Texts in Mathematics*. Springer-Verlag, 1999.
  - [9] H. Cohen and H.W. Lenstra. Heuristics on class groups of number fields. *Number Theory, Lecture notes in Math.*, 1068:33–62, 1983.
  - [10] R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering short generators of principal ideals in cyclotomic rings. *IACR Cryptology ePrint Archive*, 2015:313, 2015.
  - [11] K. Eisenträger, S. Hallgren, A. Kitaev, and F. Song. A quantum algorithm for computing the unit group of an arbitrary degree number field. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 293–302, New York, NY, USA, 2014. ACM.
  - [12] K. Eisenträger, S. Hallgren, A. Kitaev, and F. Song. A quantum algorithm for computing the unit group of an arbitrary degree number field (long version), 2015. In preparation.
  - [13] K. Eisenträger and S. Hallgren. Algorithms for ray class groups and hilbert class fields. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 471–483, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.
  - [14] Kirsten Eisenträger and Sean Hallgren. Algorithms for ray class groups and hilbert class fields. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 471–483. Society for Industrial and Applied Mathematics, 2010.
  - [15] C. Fieker. *Relative Normgleichungen*. PhD thesis, Technische Universität Berlin, 1997.
  - [16] C. Fieker. Algorithmic Number Theory. Lecture notes available at <http://www.mathematik.uni-kl.de/agag/mitglieder/professoren/prof-dr-claus-fieker>, 2014.
  - [17] C. Fieker, A. Jurk, and M. Pohst. On solving relative norm equations in algebraic number fields. *Mathematics of Computation*, 66(217):399–410, 1997.
  - [18] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Athens, Greece, May 26-30, 2013. *Proceedings*, pages 1–17, 2013.
  - [19] M. Giesbrecht, M. Jacobson, and A. Storjohann. Algorithms for large integer matrix problems. In *Proceedings of the 14th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, AAECC-14, pages 297–307, London, UK, UK, 2001. Springer-Verlag.
  - [20] S. Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474, 2005.
  - [21] S. Hallgren. Polynomial-time quantum algorithms for Pell’s equation and the principal ideal problem. *Journal of the ACM*, 54(1):1–19, 2007.
  - [22] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. Silverman, and W. Whyte. NTRUSign: Digital signatures using the NTRU lattice. In *Proceedings of the 2003 RSA Conference on The Cryptographers’ Track*, CT-RSA’03, pages 122–140, Berlin, Heidelberg, 2003. Springer-Verlag.
  - [23] D. Jao and V. Soukharev. A subexponential algorithm for evaluating large degree isogenies. In G. Hanrot, F. Morain, and E. Thomé, editors, *Algorithmic Number Theory*, volume 6197 of *Lecture Notes in Computer Science*, pages 219–233. Springer Berlin Heidelberg, 2010.
  - [24] H. Lenstra and C. Pomerance. A rigorous time bound for integer factoring. *Journal of the American Mathematical Society*, 5(3):483–516, 1992.
  - [25] J.E. Littlewood. On the class number of the corpus  $p(\sqrt{-k})$ . *Proc. London Math. Soc.*, 27:358–372, 1928.
  - [26] F. Lübeck. On the computation of elementary divisors of integer matrices. *J. Symb. Comput.*, 33(1):57–65, 2002.
  - [27] J. Neukirch. *Algebraic number theory*. Comprehensive Studies in Mathematics. Springer-Verlag, 1999. ISBN 3-540-65399-6.
  - [28] A. Schmidt and U. Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005.
  - [29] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
  - [30] D. Simons. Solving norm equations in relative number fields using s-units. *Mathematics of Computation*, 71(239):1287–1305, 2002.
  - [31] N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Nguyen and D. Pointcheval, editors, *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer Berlin Heidelberg, 2010.

# A quantum algorithm for computing the unit group of an arbitrary degree number field

Kirsten Eisenträger<sup>\*</sup>  
Department of Mathematics  
The Pennsylvania State  
University  
eisentra@math.psu.edu  
and Harvard University

Sean Hallgren<sup>†</sup>  
Dept. of Computer Science  
and Engineering  
The Pennsylvania State  
University  
hallgren@cse.psu.edu

Alexei Kitaev  
Kavli Institute for Theoretical  
Physics  
University of California, Santa  
Barbara  
kitaev@kitp.ucsb.edu  
and California Institute of  
Technology

Fang Song  
Department of Combinatorics  
& Optimization  
and Institute for Quantum  
Computing  
University of Waterloo  
fang.song@uwaterloo.ca

## ABSTRACT

Computing the group of units in a field of algebraic numbers is one of the central tasks of computational algebraic number theory. It is believed to be hard classically, which is of interest for cryptography. In the quantum setting, efficient algorithms were previously known for fields of constant degree. We give a quantum algorithm that is polynomial in the degree of the field and the logarithm of its discriminant. This is achieved by combining three new results. The first is a classical algorithm for computing a basis for certain ideal lattices with doubly exponentially large generators. The second shows that a Gaussian-weighted superposition of lattice points, with an appropriate encoding, can be used to provide a unique representation of a real-valued lattice. The third is an extension of the hidden subgroup problem to continuous groups and a quantum algorithm for solving the HSP over

<sup>\*</sup>Partially supported by National Science Foundation grant DMS-1056703 and by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-12-1-0522. Part of this work was done while the first author was visiting Harvard University and MIT.

<sup>†</sup>Partially supported by National Science Foundation awards CCF-0747274 and CCF-1218721, and by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-12-1-0522. Part of this work was done while visiting MIT.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

STOC '14, May 31 - June 03 2014, New York, NY, USA

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2710-7/14/05\$15.00.

<http://dx.doi.org/10.1145/2591796.2591860>.

the group  $\mathbb{R}^n$ .

## Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

## General Terms

Algorithms, Theory

## Keywords

Quantum Algorithms, Unit Group, Computational Algebraic Number Theory

## 1. INTRODUCTION

The problems where quantum algorithms have exponential speedups over the best known classical algorithm have mostly been of number theoretic origin. Shor found quantum algorithms for factoring and discrete log [Sho97] and Hallgren found a quantum algorithm for solving Pell's equation [Hal07]. These algorithms were further generalized to finding the unit group of a number field and related problems [Hal05, SV05]. The running time is measured in terms of the discriminant and the degree of the number field. The degree of a number field is its dimension as a vector space over  $\mathbb{Q}$ , while the discriminant is related to the volume of the fundamental domain of the ring of integers. The algorithms in [Hal05, SV05] are only efficient for constant degree number fields. In this paper we address the arbitrary degree case and give an algorithm that is efficient in both the discriminant and the degree.

A number field  $K$  can be defined as a subfield of the complex numbers  $\mathbb{C}$  which is generated over the rational numbers  $\mathbb{Q}$  by an algebraic number, i.e.  $K = \mathbb{Q}(\theta)$ , where  $\theta$  is the root of a polynomial with rational coefficients. If  $K$  is a number

field, then the subset of  $K$  consisting of all elements that are roots of monic polynomials with integer coefficients, forms a ring  $\mathcal{O}$ , called the ring of integers of  $K$ . The ring  $\mathcal{O} \subseteq K$  can be thought of as a generalization of  $\mathbb{Z} \subset \mathbb{Q}$ . In particular, we can ask whether  $\mathcal{O}$  is a principal ideal domain, whether elements of  $\mathcal{O}$  have unique factorization, and what the set of invertible elements is. The unit group  $\mathcal{O}^*$  is the set of invertible algebraic integers inside  $K$ , that is, elements  $\alpha \in \mathcal{O}$  such that  $\alpha^{-1} \in \mathcal{O}$ .

Computing the unit group of a number field is an important problem in computational number theory. By Dirichlet's Theorem the group of units  $\mathcal{O}^*$  is isomorphic to  $\mu(K) \times \mathbb{Z}^{s+t-1}$ , where  $\mu(K)$  are the roots of unity contained in  $K$  and  $K$  has  $s$  real embeddings and  $t$  pairs of complex conjugate embeddings. An elementary version of the problem is Pell's equation: given a positive non-square integer  $d$ , find  $x$  and  $y$  such that  $x^2 - dy^2 = 1$ . Solutions to this equation are parametrized by the formula  $x_k + y_k \sqrt{d} = (x_1 + y_1 \sqrt{d})^k$ ; the numbers  $\pm(x_k + y_k \sqrt{d})$  are exactly the units of the quadratic ring  $\mathbb{Z}[\sqrt{d}]$  (or a subgroup of index 2 if there is a unit that has norm  $-1$ ). The fundamental solution  $(x_1, y_1)$  is difficult to find, or even to write down because it may be exponential in  $d$  (i.e., doubly-exponential). Moreover, the computation of the real number  $R = \ln(x_1 + y_1 \sqrt{d})$  with a polynomial number of precision digits is believed to be a hard problem classically.

A polynomial time quantum algorithm for the computation of  $R$  was given in [Hal07]. The approach is to reduce the problem to a hidden subgroup problem (HSP) over the real numbers  $\mathbb{R}$ , and then to give a quantum algorithm for that hidden subgroup problem. In this context, the HSP amounts to having a periodic function on  $\mathbb{R}$  which is 1-1 within the period. The goal is to approximate the period.

For the unit group the corresponding periodic function  $g$  takes a real number  $u$  to a lattice  $g(u) \subset \mathbb{R}^2$ . More specifically, we can embed  $\mathcal{O}$  as a lattice, and then  $g(u)$  is obtained by stretching by a factor of  $e^u$  in one direction and squeezing in the other:

$$g(u) = (e^u, e^{-u}) \mathcal{O} := \{(e^u z^{(1)}, e^{-u} z^{(2)}) : (z^{(1)}, z^{(2)}) \in \mathcal{O}\},$$

where  $\mathcal{O} = \{(x + y\sqrt{d}, x - y\sqrt{d}) : x, y \in \mathbb{Z}\}$ ,

for  $d \equiv 2, 3 \pmod{4}$ . The function  $g$  is periodic with period  $R$  and 1-1 within the period. However, exponential stretching and squeezing of lattices are not computationally trivial. Furthermore, the standard quantum algorithm for the hidden subgroup problem requires a unique representation of the oracle function value (a representation up to an equivalence relation will not work). In [Hal07] these issues were addressed by using an intricate notion of "reduced ideals". This method was extended to constant degree number fields [Hal05, SV05], but it is difficult to generalize this method to rings of higher degree. At a minimum, computing the required reduced ideals seems to require solving the shortest vector problem in ideal lattices of dimension  $n$ , and enumerating lattice points also seems necessary. Cryptosystems whose security relies on the hardness of solving problems in ideal lattices have been suggested for cryptography [PR07, LPR10]. Another problem is running the hidden subgroup algorithm for the continuous group  $G = \mathbb{R}^m$ , where rounding causes errors. Such errors are tolerable when  $m$  is fixed, but worsen in higher dimensions.

We propose a different scheme, leading to a quantum re-

duction from computing the unit group of a number field of arbitrary degree  $n$  to solving an Abelian hidden subgroup problem over  $\mathbb{R}^m$ , where  $m = O(n)$ . It involves several important ingredients. First, we represent a lattice by a reduced basis (up to some precision). The exponential transformation is performed using repeated squaring of lattices. These lattices can be multiplied because they are also ideals. Having obtained some basis of the lattice  $L = f(u)$ , we construct a canonical *quantum representation* of  $L$ , namely the Gaussian-weighted superposition of lattice points with a sufficiently large dispersion. To ensure stability against rounding errors, each lattice point is represented by a superposition of nearby points in a fine grid. (For example, in one dimension, such a superposition straddles two adjacent grid points.) The initial idea for handling this was using double Gaussian states as in [GKP01], which required a different representation of points. In addition to showing how to classically compute approximate bases for the stretched lattices, we prove that the inner product of Gaussian lattice states has a hidden subgroup property.

One byproduct of this work is a generalization of the HSP to uncountable topological groups such as  $\mathbb{R}$ . Most exponential speedups by quantum algorithms either use or try to use the HSP [FIM<sup>+</sup>03, HMR<sup>+</sup>10]. In the HSP a function  $f : G \rightarrow S$  is given on a group  $G$  to some set  $S$ . For an unknown subgroup  $H \subseteq G$ , the function is constant on cosets of  $H$  and distinct on different cosets. The goal is to find a set of generators for  $H$  in time polynomial in the appropriate input size, e.g.  $\log |G|$ . When  $G$  is finite Abelian or  $\mathbb{Z}^m$  there is an efficient quantum algorithm to solve the problem.

Using the usual definition of the HSP for the group  $G = \mathbb{R}$  does not work as can be seen by the following illustration. When the group is discrete the function can be evaluated on any group element. For example, it is possible to verify that a given element  $h$  is in  $H$ , by testing if  $g(0) = g(h)$ . Over the reals, if the period is some transcendental number  $x$ , then no algorithm could ever even query  $g(x)$ , and then see that it matches  $g(0)$ . It is possible to address this by giving an ad-hoc technical definition if we replace  $\mathbb{R}$  by a discrete set with rounding, as in the case of constant degree number fields [Hal07, Hal05, SV05]. However, it is not known how to solve the HSP with such a definition. Here we give a cleaner definition using continuous functions which aids us in finding an algorithm to solve the general problem.

#### DEFINITION 1.1 (THE CONTINUOUS HSP OVER $\mathbb{R}^m$ ).

The unknown subgroup  $L \subseteq \mathbb{R}^m$  is a full-rank lattice satisfying some promise: the norm of the shortest vector is at least  $\lambda$  and the unit cell volume is at most  $d$ . The oracle has parameters  $(a, r, \varepsilon)$ . Let  $f : \mathbb{R}^m \rightarrow S$  be a function, where  $S$  is the set of unit vectors in some Hilbert space. We assume that  $f$  hides  $L$  in the following way.

1.  $f$  is periodic on  $L$ : for all  $v \in L$ ,  $x \in \mathbb{R}^m$ ,  $f(x) = f(x + v)$ ;
2.  $\| |f(x)\rangle - |f(y)\rangle \| \leq a \cdot \text{dist}(x, y)$  for all  $x, y \in \mathbb{R}^m$  (Lipschitz);
3. If  $\min_{v \in L} \|x - y - v\| \geq r$ , then  $|\langle f(x) | f(y) \rangle| \leq \varepsilon$ .

Given an efficiently computable function with this property, compute a basis for  $L$ .

We show that computing the unit group of an arbitrary degree number field can be (quantum) reduced to this definition of the HSP, and we also give a quantum algorithm for solving it. We prove the following main theorem

**THEOREM 1.2.** *There is an efficient quantum algorithm to compute the unit group of a number field  $K$  that is polynomial in the degree of  $K$  and polynomial in log of the discriminant of  $K$ .*

This follows from Theorem 4.4, Theorem 5.7, Theorem 6.1, and the fact that lattice Gaussians can be computed efficiently given an approximate basis.

Computing the unit group is one of the main computational tasks in algebraic number theory [Coh93]. Two of the others are solving the principal ideal problem and computing the class group. Based on the previous quantum algorithms for solving these three problems in the constant degree case, the unit group seems to be the most difficult part. The other two problems can be solved using the unit group algorithm and general hidden subgroup techniques. We leave the other two problems open for arbitrary degree. The main issue will be proving that the HSP functions constructed to solve them are Lipschitz.

In the context of cryptography, the problem of computing the unit group and solving the principal ideal problem are considered to be hard classically, even over degree two number fields. It was used as a basis in the Buchmann-Williams key exchange problem in an effort to find a system that is harder to break than factoring-based systems. On the other hand, the typical ideal lattice problem, such as finding short vectors over degree two number fields, is easy because the degree is constant.

In the last few years, since the discovery of homomorphic encryption and the ensuing efforts to make the systems more efficient and more secure, assumptions related to number fields have been used. These systems are set up based on high degree number fields. In [GH11], a version of the principal ideal problem where a special generator is the secret was used as the hardness assumption. The Ring-LWE problem which forms the basis in [LPR10, BV11] assumes that finding short vectors in ideal lattices of high degree number fields is hard.

To summarize, the constant degree assumptions are broken by quantum algorithms. The relatively recent high degree number field assumptions about computing short vectors are still open in terms of security against quantum computers. However, in this paper we show that it is now possible to efficiently compute the unit group in these number fields, which could move towards understanding whether the new homomorphic cryptosystems really are secure against quantum computers.

## 2. NUMBER-THEORETIC BACKGROUND

In the following  $K$  will denote a number field of degree  $n$  over  $\mathbb{Q}$ , and  $\mathcal{O}$  will denote its ring of integers. When we want to consider  $\mathcal{O}$  as a lattice in  $E = \mathbb{R}^s \times \mathbb{C}^t$  with  $s+2t = n$  (see below), we will write  $\mathcal{Q}$ . We use bold letters to designate elements of  $E$  and vectors in general.

If  $\{b_1, \dots, b_n\}$  is a basis for a lattice  $\Lambda \subseteq \mathbb{R}^n$ , let  $B$  be the matrix  $B = (b_1, \dots, b_n)$  which is composed of column vectors  $b_k$ . Then  $d(\Lambda) := |\det(B)|$  is the unit cell volume of the lattice  $\Lambda$  generated by the basis.

Elements of  $K$  can be conveniently represented by using the embeddings of  $K$  into the field of complex numbers. In general, there are  $n$  such embeddings, which break into  $s$  real ones and  $t$  complex-conjugate pairs:

$$\begin{aligned} \tau_1, \dots, \tau_s : K &\rightarrow \mathbb{R}, \\ \tau_{s+1}, \dots, \tau_{s+t}, \overline{\tau_{s+1}}, \dots, \overline{\tau_{s+t}} : K &\rightarrow \mathbb{C} \quad (s+2t = n). \end{aligned}$$

Each element  $z \in K$  is mapped to the corresponding conjugate vector  $\tau(z) = (z^{(1)}, \dots, z^{(s)}, z^{(s+1)}, \dots, z^{(s+t)}, \overline{z^{(s+1)}}, \dots, \overline{z^{(s+t)}})^T \in \mathbb{R}^s \times \mathbb{C}^{2t}$ , where the last  $t$  coordinates are redundant. Thus,  $K$  is embedded into  $E = \mathbb{R}^s \times \mathbb{C}^t$ . Conjugate vectors are added and multiplied coordinate-wise. Many useful functions on  $K$  extend naturally to  $E$ . For example, the algebraic trace and norm are defined for arbitrary conjugate vectors:

$$\text{tr}(z) = \sum_{j=1}^s z^{(j)} + \sum_{j=s+1}^{s+t} (z^{(j)} + \overline{z^{(j)}}),$$

$$\mathcal{N}(z) = \prod_{j=1}^s z^{(j)} \prod_{j=s+1}^{s+t} |z^{(j)}|^2.$$

Both these functions take real values.

As far as the additive structure is concerned, the ring  $E$  is simply an  $n$ -dimensional real space. We can define a Euclidean inner product on  $E$  by letting

$$\begin{aligned} \langle x, y \rangle &= \text{tr}(x\overline{y}) = \sum_{j=1}^s x^{(j)} y^{(j)} + \\ &2 \sum_{j=s+1}^{s+t} \left( (\text{Re } x^{(j)}) (\text{Re } y^{(j)}) + (\text{Im } x^{(j)}) (\text{Im } y^{(j)}) \right). \end{aligned}$$

The length of a vector with respect to this inner product is denoted by  $\|z\|$ .

Now let  $\{\omega_1, \dots, \omega_n\}$  be some basis (over  $\mathbb{Z}$ ) for the ring  $\mathcal{O}$  of integral elements in  $K$ . One way to characterize  $\mathcal{O}$  is by its “multiplication table”, i.e., the decomposition of  $\omega_j \omega_k$  into  $\omega_l$  with integer coefficients. In the conjugate vector representation,  $\mathcal{O}$  becomes a lattice  $\mathcal{Q} \subseteq E$  with basis  $\{z_1, \dots, z_n\}$ . From the computational perspective, it is important to have some upper bound on the length of the basis vectors or, equivalently, on the coefficients in the multiplication table. To this end, we use the notion of *discriminant*, which is defined as the determinant of the matrix  $G$  with entries  $G_{jk} = \text{tr}(\omega_j \omega_k)$ . The discriminant  $D = D(\mathcal{O})$  depends only on the ring but not the basis. The extension degree,  $n$ , and the discriminant,  $D$ , constitute a natural set of parameters characterizing the “complexity” of the ring. Our algorithm for finding the group of units is polynomial in  $n + \log |D|$ .

For various algorithmic tasks, e.g. the computation of the lattice  $e^u \mathcal{Q}$ , the basis vectors of  $\mathcal{Q}$  must be known with sufficient precision. We will use the fact that the embedding of elements of  $K$  can be found to any polynomial number of precision bits in polynomial time [Thi95].

## 3. OVERVIEW OF THE ALGORITHM

The group of units  $\mathcal{O}^*$  consists of elements  $z \in \mathcal{O}$  such that  $\mathcal{N}(z) = \pm 1$ , and they are represented by conjugate vectors

of the form  $\mathbf{z} = e^{\mathbf{u}}\mathbf{v}$ . Here  $\mathbf{u} = (u^{(1)}, \dots, u^{(s+t)}) \in \mathbb{R}^{s+t}$  satisfies the condition  $\sum_j u^{(j)} = 0$ , and the components of  $\mathbf{v} = (v^{(1)}, \dots, v^{(s+t)}) \in E = \mathbb{R}^s \times \mathbb{C}^t$  are real or complex numbers of absolute value 1. Thus, the group of units  $\mathcal{O}^*$  is contained in

$$G = \mathbb{R}^{s+t-1} \times ((\mathbb{Z}_2)^s \times (\mathbb{R}/\mathbb{Z})^t).$$

More specifically,  $\mathcal{O}^*$  is the hidden subgroup in  $G$  which corresponds to the following oracle:

$$g : G \rightarrow \text{lattices in } E : (\mathbf{u}, \mathbf{v}) \mapsto e^{\mathbf{u}}\mathbf{v}\mathcal{Q}. \quad (3.1)$$

We give an efficient classical realization of this function, where the output (i.e. a lattice  $L \subset E$ ) is represented by some basis with a certain precision. Unfortunately, such a representation is not unique, and therefore  $g$  cannot be used as an oracle for a quantum HSP algorithm. To deal with this issue, we compose the function  $g$  with another function:

$$\tilde{f} : \text{lattices in } E \rightarrow \text{quantum states} : L \mapsto |\tilde{f}(L)\rangle, \quad (3.2)$$

where  $|\tilde{f}(L)\rangle$  is a uniquely defined quantum superposition that encodes the lattice  $L$ . Thus, we obtain a usable quantum oracle

$$f = \tilde{f} \circ g : G \rightarrow \text{quantum states} : (\mathbf{u}, \mathbf{v}) \mapsto |\tilde{f}(e^{\mathbf{u}}\mathbf{v}\mathcal{Q})\rangle.$$

**Note:** By abuse of notation, we will later denote  $\tilde{f}$  as  $f$ . For example, we will refer to the quantum state that encodes the lattice  $L$  as  $|f(L)\rangle$ .

Finally, we reduce the HSP problem for  $G$  to that for  $\mathbb{R}^m$  and apply a general algorithm for finding the hidden subgroup in  $\mathbb{R}^m$ .

Thus, our algorithm for finding the group of units splits into three self-contained parts:

- A classical algorithm for computing the function  $g : (\mathbf{u}, \mathbf{v}) \mapsto \mathbf{v}e^{\mathbf{u}}\mathcal{Q}$ . Note that we cannot compute  $e^{\mathbf{u}}$  because it is an exponentially long number. Instead, we begin with representing  $\mathbf{u}$  as  $2^l\mathbf{u}_0$ , where  $\mathbf{u}_0$  is sufficiently small, and compute the lattice  $e^{\mathbf{u}_0}\mathcal{Q}$  directly. Then we apply the following procedure  $l$  times: given a basis  $\{\mathbf{z}_1, \dots, \mathbf{z}_n\}$  of the lattice  $\Lambda = e^{\mathbf{w}}\mathcal{Q}$  (for some  $\mathbf{w}$  that does not need to be known), we compute some basis of the lattice  $\Lambda^2 = e^{2\mathbf{w}}\mathcal{Q}$ . The repeated squaring yields a basis of the lattice  $e^{\mathbf{u}}\mathcal{Q}$ ; then we multiply it by  $\mathbf{v}$ . The ideal squaring procedure is not trivial. We need to compute all products  $\mathbf{z}_j\mathbf{z}_k$  and find some basis of the lattice they generate. This requires the detection of linear dependencies with integer coefficients as well as some way to prevent the vector lengths from growing. The algorithm for computing a basis for  $e^{\mathbf{u}}\mathcal{Q}$  is new as far as we know.
- A quantum procedure for the creation of the state  $|\tilde{f}(L)\rangle$  representing a lattice  $L \subset \mathbb{R}^n$ . Let  $L$  be given by some basis  $B$ . We first make a Gaussian-weighted superposition of coefficient vectors  $\mathbf{x} \in \mathbb{Z}^n$  that represent the lattice points relative to the basis. Then for each  $\mathbf{x} \in \mathbb{Z}^n$  we create the corresponding lattice point  $\mathbf{z} = B\mathbf{x} \in L$  as a quantum state. In doing so, we use a straddle encoding  $\text{str}_{n,\nu}$  (to be defined in Sect. 5) to account for rounding errors. The original value of  $\mathbf{x}$  may now be erased (in a reversible way, which requires the reconstruction of  $\mathbf{x}$  from  $\tilde{\mathbf{z}} \approx \mathbf{z}$ ). The resulting state,

$|\tilde{f}(L)\rangle = c \sum_{\mathbf{z} \in L} e^{-\pi\|\mathbf{z}\|^2/s^2} |\text{str}_{n,\nu}(\mathbf{z})\rangle$ , does not depend on the basis (except for small inaccuracies in the preparation of the Gaussian superposition). Assuming a lower bound on the length of a shortest vector,  $\lambda_1(L) \geq \lambda$ , and an upper bound on the unit cell volume,  $d(L) \leq d$ , we find a Lipschitz constant of the function  $\tilde{f}$  and estimate the inner product  $\langle \tilde{f}(L) | \tilde{f}(L') \rangle$  when the lattices  $L$  and  $L'$  are far apart.

- An efficient quantum algorithm for finding a hidden subgroup in an elementary Abelian group (as discussed in the introduction). Such groups are quotients of  $\mathbb{R}^k \times \mathbb{Z}^l$ , therefore it is sufficient to consider this case. The problem is further reduced to the HSP for  $G = \mathbb{R}^m$ . The algorithm has the usual structure. We create a superposition of points in  $\mathbb{R}^m$  with a sufficiently broad wavefunction  $w$ , apply the oracle, and measure in the Fourier basis. In the first approximation, the Fourier sampling generates a point  $u$  of the reciprocal lattice  $L^*$  with the probability distribution

$$q_u = \frac{1}{d(L)^2} \int_{(\mathbb{R}^m/L)^2} \langle f(x') | f(x) \rangle e^{2\pi i \langle x-x', u \rangle} dx dx'.$$

Repeating the procedure sufficiently many times, we obtain a set of vectors that generate  $L^*$ . In practice, the Fourier samples deviate from the lattice points by, roughly, the inverse width of the wavefunction  $w$ . Then the lattice is reconstructed from an approximate generating set (see Section 4.3).

All of our results are stated for the ring of integers  $\mathcal{O}$  of a given number field  $K$ , but they can easily be extended to general orders of  $K$ .

## 4. COMPUTING A BASIS FOR $e^t\mathcal{O}$

In this section we show how to compute an approximate basis for the lattice  $e^t\mathcal{Q}$ . Because  $e^t$  is in general doubly exponential in size and we have to use floating point computations, this is a non-trivial operation. The basic steps are to alternate ideal multiplication with size reduction to compute a short basis for the product of the two ideals that were multiplied. The algebraic numbers that appear in this computation would take exponentially many bits to represent exactly. Instead we show that a polynomial number of bits of precision is sufficient. The idea is to use the fact that we are always using ideal lattices with lower bounds and upper bounds on the vector lengths appearing throughout the computation, so that the precision loss can be bounded at each step. With this we can pick a precision high enough, some polynomial number of bits, so that we still have high precision at the end. The precision we need is that for any vector of length at most  $s\sqrt{n}$ , the computed vector is within  $1/(2N)$  of the actual vector. Here  $s$  and  $N$  are parameters chosen such that the lattice Gaussian superposition in Section 5 will be a good approximation to the lattice.

Given  $\mathbf{t} = (t_1, \dots, t_n) \in \mathbb{R}^n$  such that  $\sum t_i = 0$  we will show how to compute a basis of the lattice  $f(\mathbf{t}) = (e^{t_1}, \dots, e^{t_n}) \cdot \mathcal{Q}$ .

The function  $f : \mathbb{R}^n \rightarrow \{\text{real-valued lattices}\}$  is constant and distinct on cosets of the Log embedding of (the free part of) the units. We will later handle the fact that we only have approximations of these lattices, in particular, how to create useful superpositions using the approximations.

The main subroutine needed for computing  $f$  computes a basis of the product of two lattices. Lattices  $A$  and  $B$  can be multiplied in this case since they are always of the form  $(a_1, \dots, a_n) \cdot \mathcal{Q}$ , where  $a_i \in \mathbb{R}$ , and  $\mathcal{Q}^2 = \mathcal{O}$ . In particular, given the bases of two lattices  $A = \langle \mathbf{w}_1, \dots, \mathbf{w}_n \rangle$  and  $B = \langle \mathbf{v}_1, \dots, \mathbf{v}_n \rangle$ , all pairwise products  $\mathbf{w}_i \mathbf{v}_j$  of basis vectors are computed giving a generating set of the lattice  $AB$ , to which we apply Theorem 4.3 to obtain a size-reduced basis.

To ensure that the entire computation can be done in polynomial time we must give an upper bound for the determinant of each lattice and a lower bound for the shortest vector, which bounds the basis sizes.

## 4.1 Splitting up the computation

The computation must be split up carefully to avoid ending up with doubly exponential size coefficients. First it is split into two parts. The first part handles the integer part of the  $t_i$ 's which is complicated by the fact that  $e^{t_i}$  will be doubly-exponential in general. The solution will be to compute a sequence of ideals  $A_{-1}, A_0, \dots, A_m$  of bounded determinant such that  $A_{-1} \times \prod_{i=0}^m A_i^{2^i} = f(\mathbf{t})$ .

For  $1 \leq i \leq n-1$  let  $t_i = r_i + s_i$ , where  $r_i \in \mathbb{Z}$  and  $0 \leq s_i < 1$ . Let  $r_n = -\sum_{i=1}^{n-1} r_i$  and  $s_n = t_n - r_n$ . Using the fact that  $(e^{t_1}, \dots, e^{t_n}) \cdot \mathcal{Q} = (e^{r_1}, \dots, e^{r_n}) \cdot \mathcal{Q} \cdot (e^{s_1}, \dots, e^{s_n}) \cdot \mathcal{Q}$  we will compute these two pieces separately.

Define  $A_j = (e^{r_{1j}}, \dots, e^{r_{(n-1)j}}, (e^{-1})^{\sum_{i=0}^{n-1} r_{ij}}) \cdot \mathcal{Q}$ , where  $r_{ij}$  is  $\text{sign}(r_i)$  times bit  $j$  of  $|r_i|$ . From the determinant formula it follows that the determinant of  $A_j$  is the determinant of  $\mathcal{O}$  times  $(e^{-1})^{\sum_i r_{ij}} \prod_{i=0}^{n-1} e^{r_{ij}} = e^{\sum_i r_{ij} - \sum_i r_{ij}} = 1$ . This also bounds the powers of  $A_j$ . The log of the determinant of  $\mathcal{O}$  and  $n$  define the input size to the problem.

The second part handles the fractional part of the  $t_i$ 's by directly computing the ideal  $A_{-1} = (e^{s_1}, \dots, e^{s_n}) \cdot \mathcal{Q}$  using the first polynomially many terms in the formula  $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$  to get the desired approximation. From the determinant formula it follows that  $\det A_{-1}$  is  $\prod_i e^{s_i}$  times the determinant of  $\mathcal{O}$ . The product  $\prod_i e^{s_i}$  is between  $e^{-2n}$  and  $e^{2n}$ .

To see that  $f(\mathbf{t}) = A_{-1} \cdot \prod_j A_j^{2^j}$ , we compute

$$\begin{aligned} & A_{-1} \cdot \prod_j A_j^{2^j} \\ &= A_{-1} \cdot \prod_j \left( (e^{r_{1j}}, \dots, e^{r_{(n-1)j}}, (e^{-1})^{\sum_i r_{ij}}) \cdot \mathcal{Q} \right)^{2^j} \\ &= A_{-1} \cdot (e^{\sum_j r_{1j} 2^j}, \dots, e^{\sum_j r_{(n-1)j} 2^j}, (e^{-1})^{\sum_j \sum_i r_{ij} 2^j}) \cdot \mathcal{Q} \\ &= (e^{s_1}, \dots, e^{s_n}) \cdot (e^{r_1}, \dots, e^{r_{n-1}}, (e^{-1})^{\sum_i r_i}) \cdot \mathcal{Q} \\ &= (e^{t_1}, \dots, e^{t_{n-1}}, (e^{-1})^{\sum_i t_i}) \cdot \mathcal{Q} = f(\mathbf{t}) \end{aligned}$$

The algorithm now works as follows. First compute a  $\mathbb{Z}$ -basis  $\omega_1, \dots, \omega_n$  of  $\mathcal{O}$ . Next compute the conjugate vector representation  $\mathbf{z}_i = \underline{\omega}_i$ . Compute  $A_{-1}$  as described above. Next compute each  $A_j$  by first computing  $(e^{r_{1j}}, \dots, e^{r_{nj}})$  and then  $(e^{r_{1j}}, \dots, e^{r_{nj}}) \cdot \mathbf{z}_i$  for each  $i$ .

Next use repeated squaring of ideals to compute a basis for  $A_j = ((e^{t_{1j}}, \dots, e^{t_{nj}}) \cdot \mathcal{Q})^{2^j}$ . Finally, multiply the  $A_j^{2^j}$ 's and  $A_{-1}$ .

## 4.2 Computations with approximations

Now we introduce some notation and prove some facts that will allow us to analyze the algorithm for computing an

approximate basis of the ideal lattice  $e^{\mathbf{t}} \mathcal{O}$ , which is given in Section 4.4. A real number  $x$  is typically approximated by a rational number  $\tilde{x} = \frac{p}{q}$  such that  $|\tilde{x} - x| \leq \varepsilon$ . The parameter  $\varepsilon$  is called the *absolute error* and the ratio  $\gamma = \varepsilon/|x|$  is called the *relative error*. A vector  $\mathbf{x} = (x^{(1)}, \dots, x^{(n)})$  is approximated by another  $\tilde{\mathbf{x}} = (\tilde{x}^{(1)}, \dots, \tilde{x}^{(n)})$  such that  $\|\tilde{\mathbf{x}} - \mathbf{x}\| \leq \varepsilon$ ; the relative error is defined as  $\gamma = \varepsilon/\|\mathbf{x}\|$ .

We start with a lemma bounding the error in the ideal multiplication step.

**LEMMA 4.1.** *Suppose a basis  $\mathbf{b}_1, \dots, \mathbf{b}_n$  for a lattice  $e^{\mathbf{t}_1} \mathcal{O}$  and a basis  $\mathbf{c}_1, \dots, \mathbf{c}_n$  for  $e^{\mathbf{t}_2} \mathcal{O}$  are given with relative precision  $\gamma$ , and each  $e^{\mathbf{t}_i}$  has norm 1, then the resulting products  $\mathbf{b}_i \mathbf{c}_j$  have relative precision  $4\gamma\sqrt{n}\|\mathbf{b}_i\|^n$ .*

In the next section we show how to compute a basis from a generating set.

## 4.3 Computing a basis of a lattice from an approximate generating set

To compute in polynomial time a basis  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$  that is bounded in size and  $\delta$ -close to a basis for  $e^{\mathbf{t}} \mathcal{O}$  we need to compute a bounded-length basis from a generating set. This is also used in the hidden subgroup problem algorithm when computing the basis for the unit lattice from a generating set for its dual. The input and output vectors for this are approximate. We need an algorithm that can find integer dependencies among the rounded vectors and also to bound the errors that result from the transformation to the reduced basis. An algorithm for computing a lattice basis from a set of generators was given by Buchmann and Pohst [BP87] and Buchmann and Kessler [BK93]. They do not bound all of the errors, though. In order to bound the errors that result from the transformation to the reduced basis we need better bounds on the coefficient sizes in the transformation than what they give. For that reason we will present their algorithm and improve their analysis. Both [BP87] and [BK93] analyze the same algorithm for computing a basis, so we give an outline of their algorithm and bounds before giving our analysis and proving the error bounds.

The setup is as follows. Suppose that vectors  $\mathbf{a}_1, \dots, \mathbf{a}_k \in \mathbb{R}^n$  generate an  $r$ -dimensional lattice  $L$ . We are given approximations  $\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_k \in \mathbb{Z}^n$  such that  $\|\mathbf{a}_i - \hat{\mathbf{a}}_i/2^q\| \leq \sqrt{n}/2^{q+1}$ . We want to compute a set of vectors  $\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_r$  which approximate a basis  $\mathbf{c}_1, \dots, \mathbf{c}_r$  of  $L$  with some precision  $q'$ , which will necessarily be smaller than  $q$ . The algorithms in [BP87] and [BK93] compute such a basis via a reduction to LLL. We will use their algorithm to solve this problem, but we need a better analysis to make sure that the output accuracy  $q'$  is not too much smaller than  $q$ . This is done in Theorem 4.3 below.

### 4.3.1 The Buchmann-Pohst algorithm

The reduction to LLL takes the input vectors  $\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_k$  and creates a new lattice with basis defined from the concatenated vectors

$$\tilde{\mathbf{a}}_j = (\mathbf{e}_j, \hat{\mathbf{a}}_j) \quad (1 \leq j \leq k),$$

where  $\mathbf{e}_j$  denotes the  $j$ -th unit vector in  $\mathbb{Z}^k$ . These vectors  $\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_k \in \mathbb{Z}^{n+k}$  are clearly linearly independent and form a basis of the lattice  $\tilde{L} = \bigoplus_{j=1}^k \mathbb{Z} \tilde{\mathbf{a}}_j$ . Note that with this setup, the bottom of the matrix has vectors  $\lfloor 2^q \mathbf{a}_i \rfloor$  as the basis vectors, where  $\lfloor \cdot \rfloor$  rounds each coordinate of the vector. The LLL-algorithm is then applied to the lattice basis



$\tilde{\mathbf{a}}_1, \dots, \tilde{\mathbf{a}}_k$  to obtain an LLL-reduced basis  $\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_k$ . For the first  $k - r$  of these vectors we will denote the top and bottom components as

$$\tilde{\mathbf{b}}_j = (\mathbf{m}_j, \hat{\mathbf{z}}_j)^T \quad (1 \leq j \leq k - r),$$

and for the last  $r$  vectors as

$$\tilde{\mathbf{b}}_{k-r+j} = (\mathbf{m}'_j, \hat{\mathbf{b}}_j)^T \quad (1 \leq j \leq r).$$

Note that the vectors  $\mathbf{m}_1, \dots, \mathbf{m}_{k-r} \in \mathbb{Z}^k$  are the coefficient vectors transforming the vectors  $\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_k$  to the vectors  $\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_{k-r}$ , respectively, and the vector  $\mathbf{m}'_j$  takes  $\hat{\mathbf{a}}_1, \dots, \hat{\mathbf{a}}_k$  to  $\hat{\mathbf{b}}_j$  (for  $1 \leq j \leq r$ ).

In [BP87] and [BK93] it is shown that the resulting basis has the following two properties. First, the top left  $k - r$  columns contain a linearly independent set of relations  $\mathbf{m}_1, \dots, \mathbf{m}_{k-r} \in \mathbb{Z}^k$  for the exact vectors  $\mathbf{a}_1, \dots, \mathbf{a}_k$ . A relation vector  $\mathbf{m}_j = (m_{1j}, \dots, m_{kj})^t$  satisfies  $\sum_{i=1}^k m_{ij} \mathbf{a}_i = 0$ . In the exact matrix the vectors  $\mathbf{z}_1, \dots, \mathbf{z}_{k-r}$  would be zero, so the approximate vectors  $\hat{\mathbf{z}}_i$  will be small. The second property of the resulting basis is that the bottom right of the matrix contains approximations  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_r$  to a basis for the lattice generated by  $\mathbf{a}_1, \dots, \mathbf{a}_k$ .

In [BK93] it is shown that given good enough approximations of  $\mathbf{a}_1, \dots, \mathbf{a}_k$ , the vectors

$$\mathbf{b}_j = \sum_{i=1}^k m'_{i,j} \mathbf{a}_i \quad (1 \leq j \leq r) \quad (4.1)$$

form a basis for  $L$  and satisfy

$$\|\mathbf{b}_j\| \leq (\sqrt{kn} + 2) 2^{\frac{k-1}{2}} \lambda_j(L_r) \quad (1 \leq j \leq r). \quad (4.2)$$

This holds for every sublattice  $L_r$  which is spanned by a subset of  $r$  linearly independent vectors of  $\mathbf{a}_1, \dots, \mathbf{a}_k$ .

LEMMA 4.2. Choose  $q$  as in [BK93, Theorem 4.1]. The square length of the coefficient vectors  $\mathbf{m}'_j$  ( $1 \leq j \leq r$ ) that transform the generators  $\mathbf{a}_i$  of  $L$  into a basis vector  $\mathbf{b}_j$  of  $L$  as in Equation 4.1 is bounded by  $(\frac{\alpha^{r-1}}{d(L)} \cdot \sqrt{r} \Delta_{1,j})^2 + \Delta_2^2$ . Here  $\Delta_{1,j}$  is the right-hand-side of Equation 4.2,

$$\Delta_{1,j} = (\sqrt{kn} + 2) \cdot 2^{\frac{k-1}{2}} \cdot \lambda_j(L_r),$$

and  $L_r$  is any sublattice of  $L$  which is spanned by a subset of  $r$  linearly independent vectors of  $\mathbf{a}_1, \dots, \mathbf{a}_k$ . The quantity  $\Delta_2$  is

$$\Delta_2 = \sqrt{k-r} \cdot 2^{\frac{k-1}{2}} \left( \frac{k\sqrt{n}}{2} + \sqrt{k} \right) \cdot \frac{\alpha^r}{d(L)}.$$

Lemma 4.2 can be used to prove the following theorem which shows that a basis of bounded length for  $L$  exists, for which we can efficiently compute a good approximation.

THEOREM 4.3. Let  $\mathbf{a}_1, \dots, \mathbf{a}_k \in \mathbb{R}^n$  generate a lattice  $L$  of rank  $r$ , let  $\mu$  be a lower bound on the shortest vector, let  $\alpha = \max \|\mathbf{a}_i\|$ , and let  $q$  be such that  $2^q \geq (k2^{(k+1)/2} \cdot \max \|\mathbf{a}_i\|)^r / (\mu \det(L)^2)$ .

Given approximations of  $\mathbf{a}_1, \dots, \mathbf{a}_k \in \mathbb{R}^n$  with  $q$  bits of precision, a basis  $\hat{\mathbf{c}}_1, \dots, \hat{\mathbf{c}}_r$  for  $L$  can be computed in time polynomial in  $q$ , where the exact vectors  $\mathbf{c}_j$  satisfy

$$\|\mathbf{c}_j\| \leq (\sqrt{kn} + 2) 2^{\frac{k-1}{2}} \cdot \lambda_j(L).$$

The absolute error on each output vector  $\hat{\mathbf{c}}_i$  is bounded by  $rk\gamma_1\gamma_3\sqrt{n}/2^{q+1}$ .

$$\text{Here } \gamma_1 = \sqrt{\left(\frac{\alpha^{r-1}}{\det(L)} \sqrt{r} \Delta_{1,n}\right)^2 + \Delta_2^2},$$

and

$$\gamma_3 = \sqrt{\left(\frac{(\alpha')^r}{d(L)} \cdot \sqrt{r} \cdot \|\mathbf{b}_j\|\right)^2 + \left(\sqrt{k-r} \sqrt{k} \cdot \frac{(\alpha')^r}{d(L)}\right)^2},$$

with  $\alpha' = (\sqrt{kn} + 2) 2^{\frac{k-1}{2}} \alpha$ .

#### 4.4 The algorithm for computing $e^t \mathcal{O}$

Given  $t$ , compute a basis for  $e^t \mathcal{O}$ :

1. Choose a polynomial  $q$ .
2. For each bit index  $j$  do the following:
3. Compute the diagonal matrix  $T_j$ , where  $(T_j)_{i,i} = e^{r \cdot i \cdot j}$  for  $i < n$ , and  $(T_j)_{n,n} = (e^{-1})^{\sum_{i=1}^{n-1} r \cdot i \cdot j}$ .
4. Compute  $A_j := T_j \cdot \mathcal{O}$ , and compute a short basis for it using Theorem 4.3.
5. Square  $A_j$   $j$  times, using  $j$  applications of ideal multiplication below.
6. Multiply the resulting ideals together. To multiply two ideals  $B$  and  $C$  proceed as follows: Let the ideal  $B$  have basis  $\mathbf{b}_1, \dots, \mathbf{b}_n$  and ideal  $C$  have basis  $\mathbf{c}_1, \dots, \mathbf{c}_n$ .
  - (a) Multiply pairwise columns to get  $n^2$  vectors  $\mathbf{c}_1 \mathbf{b}_1, \mathbf{c}_1 \mathbf{b}_2, \dots, \mathbf{c}_1 \mathbf{b}_n, \mathbf{c}_2 \mathbf{b}_1, \dots, \mathbf{c}_n \mathbf{b}_n$ .
  - (b) Use Theorem 4.3 to compute a short basis for  $BC$ .
  - (c) Truncate the precision to  $q$  bits.

THEOREM 4.4. There is an algorithm that on input  $t \in \mathbb{Q}^n$  and  $\delta$  computes a basis  $\hat{\mathbf{b}}_1, \dots, \hat{\mathbf{b}}_n$  that is  $\delta$ -close to a basis for  $e^t \mathcal{O}$ , has bounded size, and runs in time polynomial in  $\log \Delta$ ,  $n$ ,  $\log \|t\|$  and  $\log 1/\delta$ .

PROOF. We analyze the complexity of the algorithm given above. By Thiel [Thi95] we can take the initial precision as high as we need in Step 3 and Step 4. The main step in the algorithm consists of multiplying ideals  $B$  and  $C$ , so it is enough to show that each multiplication step  $BC$  can be done efficiently, and that we can bound the loss of precision. By Lemma 4.1 we can compute  $n^2$  generators for  $BC$  from  $n$  generators for  $B$  and  $n$  from  $C$  and bound the loss of precision. By Theorem 4.3, we can compute a basis approximating  $\mathbf{b}_1, \dots, \mathbf{b}_n$  for  $BC$  in polynomial time, with

$$\|\mathbf{b}_j\| \leq (\sqrt{n^3} + 2) 2^{\frac{n^2-1}{2}} \cdot \lambda_j(BC).$$

The loss of precision for the squaring step is bounded in Theorem 4.3. Therefore we may choose the initial number of precision bits  $q$  to be high enough to satisfy  $\delta$  at the end.  $\square$

## 5. THE QUANTUM REPRESENTATION OF LATTICES

The representation of a lattice by a basis is not unique, which makes it unsuitable for use in an algorithm that deals with quantum superpositions of lattices. To avoid this issue, we will represent each lattice by a unique quantum state,

namely, a superposition of the lattice points with certain weights. Let us first discuss some desired properties of such a representation. We want a small deformation of the lattice to result in a small change in the quantum state, whereas substantially different lattices should be mapped to almost orthogonal vectors. These requirements can be formalized as follows.

**DEFINITION 5.1.** Let  $\text{dist}(x, y)$  denote the distance between two points in a metric space  $X$ , and let  $\mathcal{H}$  be some Hilbert space. A map  $f : X \rightarrow \mathcal{H}$  is called an  $(a, r, \varepsilon)$  quantum encoding if the following conditions are met:

1.  $\langle f(x) | f(x) \rangle = 1$  for all  $x \in X$ ;
2.  $||f(x) - f(y)|| \leq a \cdot \text{dist}(x, y)$  for all  $x, y \in X$ ;
3. If  $\text{dist}(x, y) \geq r$ , then  $|\langle f(x) | f(y) \rangle| \leq \varepsilon$ .

Given such an encoding, the vector  $|f(x)\rangle$  is called the signature state for  $x$ .

The number  $a$  in condition 2 is called a *Lipschitz constant* of the function  $f$ . When  $X = \mathbb{R}^n$  (or, more generally, when  $X$  is a Riemannian manifold),  $f$  can be approximated by a smooth function to an arbitrary precision in the sup-norm at cost of an arbitrary small parameter change in the above definition. For smooth functions,  $a$  is simply an upper bound on the first derivative of  $f$ .

**LEMMA 5.2.** Let  $f_1 : X_1 \rightarrow \mathcal{H}_1$  and  $f_2 : X_2 \rightarrow \mathcal{H}_2$  take values in unit vectors and satisfy the Lipschitz condition with constants  $a_1$  and  $a_2$ , respectively.

- a) If  $X_1 = X_2 = X$ , then the function  $f_1 \otimes f_2 : X \rightarrow \mathcal{H}_1 \otimes \mathcal{H}_2$  is  $(a_1 + a_2)$ -Lipschitz.
- b) If  $X_1$  and  $X_2$  are Euclidean spaces, then the function  $g : X_1 \times X_2 \rightarrow \mathcal{H}_1 \otimes \mathcal{H}_2$  defined as  $g(x_1, x_2) = f_1(x_1) \otimes f_2(x_2)$  is  $a$ -Lipschitz, where  $a = \sqrt{a_1^2 + a_2^2}$ .

**EXAMPLE 5.3 (STRADDLE ENCODING).** A representation of real numbers by quantum superposition of integers can be defined as follows:

$$|\text{str}_\nu(x)\rangle = \cos\left(\frac{\pi}{2}t\right)|k\rangle + \sin\left(\frac{\pi}{2}t\right)|k+1\rangle,$$

where  $k = \lfloor x/\nu \rfloor$ ,  $t = x/\nu - k$ .

The map  $\text{str}_\nu : \mathbb{R} \rightarrow \mathbb{C}^{\mathbb{Z}}$  is a  $(\frac{\pi}{2\nu}, 2\nu, 0)$  quantum encoding. Applying this map to each coordinate on an  $n$ -dimensional real vector, we obtain a  $(\frac{\pi}{2\nu}\sqrt{n}, 2\nu\sqrt{n}, 0)$  encoding  $\text{str}_{\nu,n} : \mathbb{R}^n \rightarrow (\mathbb{C}^{\mathbb{Z}})^{\otimes n}$  (by statement (b) of Lemma 5.2).

We usually set  $\nu = 2^{-q}$ . The transformation  $|x\rangle \mapsto |x\rangle \otimes |\text{str}_\nu(x)\rangle$  can be implemented efficiently if we assume that  $x$  is represented as  $2^{-l}\tilde{x}$ , where  $l \geq q$  and  $\tilde{x}$  is an integer, which is actually stored in the quantum memory. (In practice,  $l$  should be substantially greater than  $q$  so that the rounding error in  $x$  does not matter.) To construct  $|x\rangle \otimes |\text{str}_\nu(x)\rangle$  from  $|x\rangle$ , we compute  $k$  and  $t$ , create the state  $\cos(\frac{\pi}{2}t)|0\rangle + \sin(\frac{\pi}{2}t)|1\rangle$ , add  $k$ , and reverse the computation of  $k$  and  $t$ .

A full-rank lattice  $L \subseteq \mathbb{R}^n$  may be represented by a superposition of its points with Gaussian amplitudes. Each

lattice point is in turn represented using the straddle encoding. Thus,

$$|f(L)\rangle = \gamma^{-1/2} \sum_{x \in L} e^{-\pi\|x\|^2/s^2} |\text{str}_{n,\nu}(x)\rangle. \quad (5.1)$$

Here  $\gamma = \sum_{x \in L} e^{-2\pi\|x\|^2/s^2}$ .

We must prove the HSP properties for this state when the lattice  $L$  is of the form  $e^t\mathcal{O}$ . To prove that the states over different ideal lattices have small inner product when  $t_1 - t_2$  is not near a lattice point we need the following lemma showing that the two corresponding ideal lattices do not have too many points close to each other.

**LEMMA 5.4.** Let  $e^{t_1}$  and  $e^{t_2}$  be vectors of algebraic norm 1. If some nonzero point of  $e^{t_1}\mathcal{O}$  is equal to some point of  $e^{t_2}\mathcal{O}$  and has length at most  $R$ , then the distance between any unequal pair of points, one from  $e^{t_1}\mathcal{O}$  and one from  $e^{t_2}\mathcal{O}$ , is at least  $\sqrt{n}/R^n$ .

**PROOF.** Suppose  $e^{t_1}\mathbf{a} = e^{t_2}\mathbf{b}$  ( $\mathbf{a}, \mathbf{b} \in \mathcal{O}$ ) is a point in the intersection of  $e^{t_1}\mathcal{O}$  and  $e^{t_2}\mathcal{O}$  and has length at most  $R$ . Since the length of  $e^{t_1}\mathbf{a}$  is at most  $R$ , each coordinate of  $e^{t_1}\mathbf{a}$  is bounded by  $R$ , and hence the norm of  $e^{t_1}\mathbf{a}$ , which is the product over all coordinates, is bounded by  $R^n$ . Since  $e^{t_1}$  has norm 1, the norm of  $\mathbf{a}$  is then bounded by  $R^n$  as well. To see how close points in  $e^{t_1}\mathcal{O}$  and  $e^{t_2}\mathcal{O}$  can be (without being equal) we consider the minimum distance of points in the lattice  $e^{t_1}\mathcal{O} + e^{t_2}\mathcal{O}$ .

To compute this lattice we first compute  $e^{t_1-t_2}\mathcal{O} + \mathcal{O}$ : Since  $e^{t_1}\mathbf{a} = e^{t_2}\mathbf{b}$  we have  $\mathbf{b}/\mathbf{a} = e^{t_1-t_2}$ . So  $e^{t_1-t_2}\mathcal{O} + \mathcal{O} = (\mathbf{b}/\mathbf{a})\mathcal{O} + \mathcal{O}$ . Let  $N(\mathbf{a})$  denote the norm of  $\mathbf{a}$ . (I.e.,  $N(\mathbf{a}) = N(a_1) = \prod a_i$ , where  $\mathbf{a} = (a_1, \dots, a_n)$ .)

**Claim:**  $(\mathbf{b}/\mathbf{a})\mathcal{O} + \mathcal{O} \subseteq \frac{1}{N(\mathbf{a})}\mathcal{O}$ .

**Proof of claim:** Clearly  $\mathcal{O} \subseteq \frac{1}{N(\mathbf{a})}\mathcal{O}$ . Let  $\mathbf{a} = (a_1, \dots, a_n) \in \mathcal{O}$ . Since each  $a_i$  satisfies the same minimal polynomial as  $a_1$  they are all algebraic integers, and hence so is the product  $\prod_{i=2}^n a_i$ . Since  $a_2 \cdot a_3 \cdot \dots \cdot a_n = N(\mathbf{a})/a_1$ , the product is also in  $K$  and hence it is in  $\mathcal{O}$ . Then  $b_1 \cdot a_2 \cdot \dots \cdot a_n$  is in  $\mathcal{O}$  as well. Thus

$$(b_1/a_1)\mathcal{O} = \frac{b_1 \cdot a_2 \cdot \dots \cdot a_n}{N(\mathbf{a})}\mathcal{O} \subseteq \frac{1}{N(\mathbf{a})}\mathcal{O}.$$

Hence  $(\mathbf{b}/\mathbf{a})\mathcal{O} + \mathcal{O} \subseteq \frac{1}{N(\mathbf{a})}\mathcal{O}$ . This proves the claim and shows that  $e^{t_1-t_2}\mathcal{O} + \mathcal{O} \subseteq \frac{1}{N(\mathbf{a})}\mathcal{O}$ .

The shortest vector in  $\mathcal{O}$  is  $(1, \dots, 1)$  which has length  $\sqrt{n}$ . Since we showed that  $N(\mathbf{a}) \leq R^n$ , this implies that the shortest vector in  $\frac{1}{N(\mathbf{a})}\mathcal{O}$  has length at least  $\sqrt{n}/R^n$ .

Now we consider  $e^{t_1}\mathcal{O} + e^{t_2}\mathcal{O}$ . By the above argument

$$e^{t_1}\mathcal{O} + e^{t_2}\mathcal{O} = e^{t_2}(e^{t_1-t_2}\mathcal{O} + \mathcal{O}) \subseteq \frac{1}{N(\mathbf{a})}e^{t_2}\mathcal{O}.$$

Since the shortest vector in  $e^{t_2}\mathcal{O}$  has length at least  $\sqrt{n}$ , the shortest vector in  $e^{t_1}\mathcal{O} + e^{t_2}\mathcal{O}$  has length at least  $\sqrt{n}/R^n$ . Hence points in  $e^{t_1}\mathcal{O}$  and  $e^{t_2}\mathcal{O}$ , which are not equal, are at least  $\sqrt{n}/R^n$  apart.  $\square$

The following lemma helps bound the overlap between two lattices.

**LEMMA 5.5.** Suppose we are given lattices  $L$  and  $L'$ , and sublattices  $I \subseteq L$  and  $I' \subseteq L'$ . Assume there is a 1-1 correspondence  $h : I \rightarrow I'$  s.t., for any  $(x, x') \in \tilde{L} \times \tilde{L}'$ ,

1. if  $(x, x') \in C := \{(u, v) \in I \times I' : v = h(u)\}$ , then  $\|x - x'\| \leq \varepsilon$ ,
2. otherwise  $\|x - x'\| \geq 2\nu\sqrt{n}$ .

If  $I \subsetneq L$  and  $I' \subsetneq L'$ , then for any  $n \geq 5$ ,  $\langle f(L) | f(L') \rangle \leq 3/4$  if in addition  $s \geq 4\pi n^3 \max\{\lambda_n(L), \lambda_n(L')\}$ .

LEMMA 5.6. Let  $\nu \leq \frac{\lambda}{2\sqrt{n}}$  and  $s \geq cn(\sqrt{n}/\lambda)^{n-1}d$  for a certain constant  $c$ , and let us restrict the encoding  $L \mapsto |f(L)\rangle$  to lattices with  $d(L) \leq d$  and  $\lambda_1(L) \geq \lambda$ . On such lattices,  $f$  has a Lipschitz constant

$$a = \frac{\sqrt{\pi n} s}{4\nu} + 1.$$

Next we can choose the parameters to show that the hidden subgroup property holds. Since the inner product is at most a constant  $< 1$ , a tensor product of  $n$  copies will reduce the inner product to be exponentially small. For simplicity, we state the theorem only for the free part of the unit group  $\text{Log } \mathcal{O}^* \leq \mathbb{R}^{s+t-1}$ , and for elements of norm 1.

THEOREM 5.7. Let  $s = 2^{2n}\sqrt{nD}$ ,  $\nu = 1/(4n(s\sqrt{n})^{2n})$ . Let  $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^n$ . Let  $\gamma = (\gamma_1, \dots, \gamma_n)$  be  $\mathbf{t}_2 - \mathbf{t}_1 - \mathbf{u}$ , where  $\mathbf{u}$  is the unit closest to  $\mathbf{t}_2 - \mathbf{t}_1$  in  $\text{Log } \mathcal{O}^*$ . Then the function  $f$  is  $a$ -Lipschitz with constant  $a = \frac{\sqrt{\pi n} s}{4\nu} + 1$ .

The inner product  $\langle e^{\mathbf{t}_1} \mathcal{O} | e^{\mathbf{t}_2} \mathcal{O} \rangle$  is at most  $3/4$  if for some  $i$ , we either have  $\ln(1 - (s\sqrt{n})^{n-1} 2\nu\sqrt{n}) \geq \gamma_i$  or  $\gamma_i \geq \ln(1 + (s\sqrt{n})^{n-1} 2\nu\sqrt{n})$ .

The idea is as follows: fix an ideal  $e^{\mathbf{t}_1} \mathcal{O}$ . We want to bound its inner product with ideals  $e^{\mathbf{t}_2} \mathcal{O}$ . First we show that when the two lattices have points inside the ball of radius  $s\sqrt{n}$  where they overlap exactly, then their inner product must be small (unless the two lattices coincide, i.e. when  $\mathbf{t}_2 - \mathbf{t}_1$  is a unit). Then we show that this inner product does not get bigger when we perturb one of the lattices slightly. Finally, we show that when two lattices are not close to having an exact point of intersection, then their inner product is small as well.

Finally, it can be shown that a good approximation of these states can be computed when evaluated on rational numbers. First an approximate basis is computed using the algorithm in the last section, and then the superposition is created over the points.

## 6. THE HIDDEN SUBGROUP PROBLEM ON A CONTINUOUS GROUP

The HSP algorithm for  $\mathbb{R}^m$  can be thought of in the usual structure, however the analysis is difficult, and we combine phase estimation and our new continuous definition of the HSP to get it to work. The analysis works in continuous space and is discretized in a general way to derive the algorithm. The algorithm creates a superposition of points in  $\mathbb{R}^m$  with a sufficiently broad wavefunction  $w$ . We cannot measure in the Fourier basis of the continuous group, but we show that phase estimation can be used to approximate this to measure a point  $u$  of the reciprocal lattice  $L^*$  with the probability distribution

$$q_u = \frac{1}{d(L)^2} \int_{(\mathbb{R}^m/L)^2} \langle f(x') | f(x) \rangle e^{2\pi i \langle x - x', u \rangle} dx dx'. \quad (6.1)$$

This distribution is not close to uniform but we are able to show that the probability of staying in any sublattice is bounded. For rounding, the samples deviate from the lattice points by, roughly, the inverse width of the wavefunction  $w$ . We show that the reconstruction of a lattice from an approximate generating set can be done using an improved analysis of [BK93] in Section 4.3. It can be shown that the condition number of a reduced basis is bounded so that the dual lattice, which is the hidden subgroup, can be computed. These provide the main ingredients in the full proof.

THEOREM 6.1. There is a polynomial time quantum algorithm for solving the HSP over  $\mathbb{R}^m$ .

In Section 6.2 we outline the steps of the algorithm and indicate how to derive the probability expression. First we show that the known cases of the Abelian HSP can be reduced to the new continuous case in Definition 1.1 over the HSP instance  $\mathbb{R}^m$ .

### 6.1 Application: Reduction to $G = \mathbb{R}^m$

Our HSP algorithm is applicable to Abelian groups of the form  $\mathbb{R}^k \times \mathbb{Z}^l \times (\mathbb{R}/\mathbb{Z})^s \times H$ , where  $H$  is finite. We call such groups “elementary”. The reduction to  $G = \mathbb{R}^k \times \mathbb{Z}^l$  is straightforward. In the case of interest, the hidden subgroup  $L$  is a full-rank lattice in  $G \subseteq \mathbb{R}^k \times \mathbb{R}^l$  such that  $\lambda_1(L \cap \mathbb{R}^k) \geq \lambda$  and  $d(L) \leq d$  for some fixed numbers  $\lambda$  and  $d$ . We now describe the further reduction to the group  $\tilde{G} = \mathbb{R}^{k+l}$ .

The main idea can be illustrated in the one-dimensional case, where the parameter  $\lambda$  has no meaning. We embed  $G = \mathbb{Z}$  into  $\tilde{G} = \mathbb{R}$  in the standard way, set  $\nu = 2^{-q}$  for some  $q \geq 2$ , and define the  $\mathbb{R}$ -oracle  $g$  in terms of the  $\mathbb{Z}$ -oracle  $f$  as follows:

$$\begin{aligned} |g(x)\rangle &= c_0 |\text{str}_\nu(t)\rangle \otimes |f(s)\rangle + \\ &\quad c_1 |\text{str}_\nu(t-1)\rangle \otimes |f(s+1)\rangle, \end{aligned} \quad (6.2)$$

where  $s = \lfloor x \rfloor$ ,  $t = x - s$ ,  $c_0 = \cos(\frac{\pi}{2}t)$ ,  $c_1 = \sin(\frac{\pi}{2}t)$ .

It is clear that  $g$  is a continuous function. If  $f$  is a periodic function, then  $g$  is also periodic with the same period.

To construct the state  $|g(x)\rangle$  using the original oracle  $f$ , we compute  $s$  and  $t$ , use them as parameters in the following sequence of operations, and “uncompute”  $s$  and  $t$ :

$$\begin{aligned} |0\rangle &\mapsto \sum_z c_z |z\rangle \mapsto \sum_z c_z |z\rangle \otimes |f(s+z)\rangle \\ &\mapsto \sum_z c_z |\text{str}_\nu(t-z)\rangle \otimes |f(s+z)\rangle, \end{aligned}$$

where  $z \in \{0, 1\}$ . The last step,  $|z\rangle \mapsto |\text{str}_\nu(t-z)\rangle$ , requires that we discriminate between the states  $|\text{str}_\nu(t)\rangle$  and  $|\text{str}_\nu(t-1)\rangle$ . This is easy because the supports of those states on the  $\nu$ -grid do not overlap.

Let us now consider the general case,  $G = \mathbb{R}^k \times \mathbb{Z}^l$ . The group  $G$  is embedded in  $\tilde{G} = \mathbb{R}^{k+l}$  by scaling the  $\mathbb{Z}$  factors by  $\lambda$ . This is to guarantee that  $\lambda_1(\tilde{L}) \geq \lambda$ , where  $\tilde{L}$  is the image of  $L$  under the embedding. The other condition on the new hidden subgroup reads:  $d(\tilde{L}) \leq \tilde{d}$ , where  $\tilde{d} = d\lambda^l$ . The generalization of Eq. (6.2) is straightforward:

$$\begin{aligned} |g(\mathbf{x}, x_1, \dots, x_l)\rangle &= \sum_{z_1, \dots, z_l \in \{0, 1\}} \left( \bigotimes_{j=1}^l |\psi(x_j, z_j)\rangle \right) \\ &\quad \otimes |f(\mathbf{x}, s(x_1, z_1), \dots, s(x_l, z_l))\rangle, \end{aligned} \quad (6.3)$$

where  $s(x, z) = \lfloor x/\lambda \rfloor + z$ ,  $|\psi(x, z)\rangle = \cos(\frac{\pi}{2}t) |\text{str}_\nu(t)\rangle$ , with  $t = x/\lambda - s(x, z)$ .

Note that the terms in the above sum are mutually orthogonal vectors. It can be shown that  $g$  has parameters  $\tilde{a}^2 = a^2 + l(\frac{\pi}{2\nu\lambda}(1 + \nu))^2$ ,  $\tilde{r}^2 = r^2 + l(2\nu\lambda)^2$  and  $\epsilon$ .

## 6.2 An HSP algorithm for the group $\mathbb{R}^m$

Let  $f$  be an  $(a, r, \varepsilon)$  oracle function for some full-rank lattice  $L \subseteq \mathbb{R}^m$  such that  $\lambda_1(L) \geq \lambda$  and  $d(L) \leq d$  (see Definition 1.1). The core part of our algorithm is a sampling subroutine that generates an approximation to a random point of the reciprocal lattice  $L^*$ . It works under certain assumptions about the oracle parameters.

Let  $\omega : \mathbb{R} \rightarrow \mathbb{C}$  be some Lipschitz function with unit  $L^2$ -norm supported on the interval  $[0, 1]$ . For example,

$$\omega(x) = \begin{cases} \sqrt{2} \sin(\pi x) & \text{for } x \in [0, 1], \\ 0 & \text{otherwise.} \end{cases} \quad (6.4)$$

Let us also choose a sufficiently large number  $\Delta = 2^{q_1}$  and a sufficiently small number  $\delta = 2^{-q_2}$ . Define

$$w(x_1, \dots, x_m) = \frac{1}{\Delta^{m/2}} \prod_{j=1}^m \omega\left(\frac{x_j}{\Delta}\right), \quad (6.5)$$

$w_\delta = w|_{\delta\mathbb{Z}^m}$  (restriction of  $w$  to the lattice  $\delta\mathbb{Z}^m$ ).

In our calculations, we will use the following variables:

Real domain:  $x = \delta\tilde{x} \in \delta\mathbb{Z}^m$  or  $\tilde{x} \in \mathbb{Z}^m$ ;  
Fourier domain:  $y = \delta^{-1}\tilde{y} \in \mathbb{R}^m/\delta^{-1}\mathbb{Z}^m$  or  $\tilde{y} \in \mathbb{R}^m/\mathbb{Z}^m$ .

We first create the superposition of points  $x$  with the wavefunction  $w_\delta$ . In the quantum computer,  $x$  is actually represented by  $\tilde{x}$ , therefore the initial state may be written as follows:

$$|w_\delta\rangle = \delta^{m/2} \sum_{\tilde{x} \in \mathbb{Z}^m} w(x) |\tilde{x}\rangle \quad \text{with } x = \delta\tilde{x}.$$

(Our choice of the function  $\omega$  guarantees the correct normalization on the  $\delta$ -grid; otherwise we would need to multiply the above expression by some factor that tends to 1 as  $\delta$  tends to 0.) Then we apply the oracle to get the state

$$|\psi_\delta\rangle = \delta^{m/2} \sum_{\tilde{x} \in \mathbb{Z}^m} w(x) |\tilde{x}\rangle \otimes |f(x)\rangle \quad \text{with } x = \delta\tilde{x}. \quad (6.6)$$

The quantum register containing  $f(x)$  may be ignored, and we would like to measure the other register in the Fourier basis,

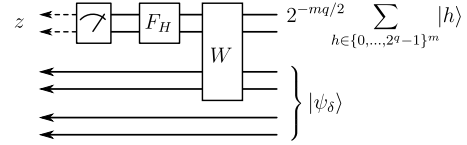
$$|\xi_{\tilde{y}}\rangle = F_{\mathbb{Z}^m}^{-1} |\tilde{y}\rangle = \sum_{\tilde{x} \in \mathbb{Z}^m} e^{-2\pi i \langle \tilde{x}, \tilde{y} \rangle} |\tilde{x}\rangle, \quad \text{where } \tilde{y} \in (\mathbb{R}/\mathbb{Z})^m.$$

An obvious procedure would be to perform the Fourier transform and measure in the standard basis. However, a quantum computer can only do the Fourier transform over a finite group, and the resulting approximation errors are difficult to analyze. Therefore we use a different method.

Note that  $|\xi_{\tilde{y}}\rangle$  is an eigenvector of the mutually commuting translation operators  $T_{e_1}, \dots, T_{e_m}$ , where  $e_j$  ( $j = 1, \dots, m$ ) are the generators of the group  $\mathbb{Z}^m$ . The translation by  $h \in G$  on an Abelian group  $G$  is defined as follows:

$$T_h : L^2(G) \rightarrow L^2(G), \quad (T_h f)(x) = f(x - h). \quad (6.7)$$

Thus, the Fourier measurement is equivalent to measuring the eigenvalues of the unitary operators  $T_{e_j}$ . A general procedure for the eigenvalue measurement (a.k.a. phase estimation) is described in [Kit95]. To illustrate the difference from the direct use of discrete Fourier transform, let us consider a variant of the phase estimation. In the following circuit, the Fourier transform  $F_H$  on the group  $H = (\mathbb{Z}_{2^q})^m$  acts on a set of ancillary qubits.



Here  $W|h, \tilde{x}\rangle = |h, \tilde{x} + h\rangle$ . For each value of  $h$ , the operator  $W$  acts on  $\tilde{x}$  as  $T_1^{h_1} \dots T_m^{h_m}$ . Note that the “+” in the definition of  $W$  means the addition of integer vectors, which are *not* reduced modulo  $2^q$ . Thus,  $W$  preserves the decomposition of  $|\psi_\delta\rangle$  into the vectors  $|\xi_{\tilde{y}}\rangle$ . The measurement outcome  $z$  may be regarded as a random variable conditioned on  $\tilde{y}$ , and the latter can be inferred from the former with some precision and confidence. The final result of the sampling subroutine,  $Y = -\delta^{-1}2^{-q}z_j$  provides an approximation for  $y$ . The error bound for this procedure is pretty standard.

**LEMMA 6.2.** *For each  $j$ , the probability that the inferred value  $\tilde{Y}_j = -2^{-q}z_j$  deviates from  $\tilde{y}_j$  by  $\tilde{y}_j \gg \tilde{\nu}$  is at most  $2^{-q}/\tilde{\nu}$ . Thus,  $Y$  approximates  $y$  with precision  $\nu$  in each coordinate, up to an error probability  $\mu_{\text{meas}} = m2^{-q}/(\delta\nu)$ .*

To further analyze the sampling subroutine, we approximate the probability distribution  $p_\delta(y)$  of the variable  $y = \delta^{-1}\tilde{y}$  by the distribution  $p$  that occurs in the  $\delta \rightarrow 0$  limit. These two distributions are derived from the following quantum states (cf. Eq. (6.6)):

$$|\psi_\delta\rangle = \delta^{m/2} \sum_{\tilde{x} \in \mathbb{Z}^m} |\tilde{x}\rangle \otimes |\psi(x)\rangle, \quad |\psi\rangle = \int_{\mathbb{R}^m} |x\rangle \otimes |\psi(x)\rangle dx,$$

where  $|\psi(x)\rangle = w(x)|f(x)\rangle$ . (The function  $w : \mathbb{R}^m \rightarrow \mathbb{C}$  is given by Eq. (6.5); the hidden subgroup oracle  $f$  and, thus, the function  $\psi$ , are vector-valued, i.e.  $f, \psi : \mathbb{R}^m \rightarrow \mathcal{H}$ .) More exactly,  $p_\delta$  and  $p$  are related to the Fourier transform of  $\psi_\delta$  and  $\psi$ , respectively:

$$\begin{aligned} p_\delta(y) &= \langle \hat{\psi}_\delta(y) | \hat{\psi}_\delta(y) \rangle & \text{with } \hat{\psi}_\delta &= F_{\delta\mathbb{Z}^m} \psi; \\ p(y) &= \langle \hat{\psi}(y) | \hat{\psi}(y) \rangle & \text{with } \hat{\psi} &= F_{\mathbb{R}^m} \psi. \end{aligned} \quad (6.8)$$

It can be shown that  $p_\delta$  is close to  $p$ .

Let us now focus on the distribution  $p(y) = \langle \hat{\psi}(y) | \hat{\psi}(y) \rangle$ . We have

$$\psi = w f, \quad \hat{\psi} = \hat{w} * (F_{\mathbb{R}^m} f), \quad \text{where } \hat{w} = F_{\mathbb{R}^m} w.$$

Since  $f$  is a hidden subgroup oracle, we may regard it as a function on  $\mathbb{R}^m/L$  and define  $\hat{f} = F_{\mathbb{R}^m/L} f$ . That is,

$$\begin{aligned} \hat{f}_u &= \int_{\mathbb{R}^m/L} e^{2\pi i \langle x, u \rangle} f(x) dx \quad \text{for } u \in L^*, \\ f(x) &= \frac{1}{d(L)} \sum_{u \in L^*} e^{-2\pi i \langle x, u \rangle} \hat{f}_u. \end{aligned} \quad (6.9)$$

The Fourier transform over  $\mathbb{R}^m$  is

$$\begin{aligned}(F_{\mathbb{R}^m} f)(y) &= \int_{\mathbb{R}^m} e^{2\pi i \langle x, y \rangle} \left( \frac{1}{d(L)} \sum_{u \in L^*} e^{-2\pi i \langle x, u \rangle} \hat{f}_u \right) dx \\ &= \frac{1}{d(L)} \sum_{u \in L^*} \hat{f}_u \delta(y - u).\end{aligned}$$

It follows that

$$\hat{\psi}(y) = (\hat{w} * (F_{\mathbb{R}^m} f))(y) = \frac{1}{d(L)} \sum_{u \in L^*} \hat{f}_u \hat{w}(y - u), \quad (6.10)$$

$$p(y) = \frac{1}{d(L)^2} \sum_{u, u' \in L^*} \langle \hat{f}_{u'} | \hat{f}_u \rangle \hat{w}(y - u) \overline{\hat{w}(y - u')}. \quad (6.11)$$

The last equation is complicated, but to have a good approximation it is enough to keep the terms with  $u = u'$ . Let us consider the quantum state  $F_{\mathbb{R}^m} |\psi\rangle$  whose wavefunction is given by Eq. (6.10). It consists of identically shaped peaks at the points  $u \in L^*$ . Each peak has a weight

$$q_u = \frac{\langle \hat{f}_u | \hat{f}_u \rangle}{d(L)^2} = \frac{1}{d(L)^2} \int e^{2\pi i \langle x - x', u \rangle} \langle f(x') | f(x) \rangle dx dx',$$

where the integral is over  $(\mathbb{R}^m/L)^2$ . The numbers  $q_u$  can be interpreted as probabilities because they are nonnegative and add up to 1. Indeed, let us normalize  $f$  to make a function of unit norm,  $g(x) = d(L)^{-1/2} f(x)$ . Then

$$\sum_{u \in L^*} q_u = \frac{1}{d(L)} \sum_{u \in L^*} \langle \hat{g}_u | \hat{g}_u \rangle = \langle \hat{g} | \hat{g} \rangle = \langle g | g \rangle = 1.$$

## 7. REFERENCES

- [Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
- [BK93] Johannes Buchmann and Volker Kessler. Computing a reduced lattice basis from a generating system, 1993. Preprint, August 4, 1993.
- [BP87] Johannes Buchmann and Michael Pohst. Computing a lattice basis from a system of generating vectors. In *Eurocal'87*, volume 378 of *LNCS*, pages 54–63. Springer-Verlag, June 1987.
- [BV11] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in cryptology—CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, 2011.
- [Coh93] Henri Cohen. *A course in computational algebraic number theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1993.
- [FIM<sup>+</sup>03] Katalin Friedl, Gabor Ivanyos, Frederic Magniez, Miklos Santha, and Pranab Sen. Hidden translation and orbit coset in quantum computing. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, San Diego, CA, 9–11 June 2003.
- [GH11] C. Gentry and S. Halevi. Implementing gentry's fully-homomorphic encryption scheme. *Eurocrypt 2011*, pages 132–150, 2011.
- [GKP01] Daniel Gottesman, Alexei Kitaev, and John Preskill. Encoding a qubit in an oscillator. *Phys. Rev. A*, 64:012310, Jun 2001.
- [GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.
- [Hal05] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474, 2005.
- [Hal07] Sean Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. *Journal of the ACM*, 54(1):1–19, 2007.
- [HMR<sup>+</sup>10] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57:34:1–34:33, November 2010.
- [Kit95] Alexei Kitaev. Quantum measurements and the abelian stabilizer problem, 1995. quant-ph/9511026.
- [KW08] Alexei Kitaev and William A. Webb. Wavefunction preparation and resampling using a quantum computer, January 2008. arXiv:0801.03422.
- [LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in cryptology—EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, 2010.
- [Mic08] Daniele Micciancio. Efficient reductions among lattice problems. In *Proceedings of SODA 2008*, pages 84–93, New York, 2008. ACM.
- [PR07] Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 478–487, New York, NY, USA, 2007. ACM Press.
- [San91] Jonathan W. Sands. Generalization of a theorem of Siegel. *Acta Arith.*, 58(1):47–57, 1991.
- [Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [SV05] Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005.
- [Thi95] Christoph Thiel. *On the complexity of some problems in algorithmic algebraic number theory*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1995.