Another Round of Breaking and Making Quantum Money:
How to Not Build It from Lattices, and More

Jiahui Liu* Hart Montgomery† Mark Zhandry ‡

Abstract

Public verification of quantum money has been one of the central objects in quantum cryptography ever since Wiesner's pioneering idea of using quantum mechanics to construct banknotes against counterfeiting. So far, we do not know any publicly-verifiable quantum money scheme that is provably secure from standard assumptions.

In this work, we provide both negative and positive results for publicly verifiable quantum money.

- In the first part, we give a general theorem, showing that a certain natural class of quantum money schemes from lattices cannot be secure. We use this theorem to break the recent quantum money scheme of Khesin, Lu, and Shor.
- In the second part, we propose a framework for building quantum money and quantum lightning we call *invariant money* which abstracts some of the ideas of quantum money from knots by Farhi et al.(ITCS'12). In addition to formalizing this framework, we provide concrete hard computational problems loosely inspired by classical knowledge-of-exponent assumptions, whose hardness would imply the security of *quantum lightning*, a strengthening of quantum money where not even the bank can duplicate banknotes.
- We discuss potential instantiations of our framework, including an oracle construction using cryptographic group actions and instantiations from rerandomizable functional encryption, isogenies over elliptic curves, and knots.

1 Introduction

1.1 Motivation

Quantum information promises to revolutionize cryptography. In particular, the no cloning theorem of quantum mechanics opens the door to *quantum cryptography*: cryptographic applications that are simply impossible classically. The progenitor of this field, due to Wiesner [Wie83], is quantum money: quantum digital currency that cannot be counterfeited due to the laws of physics. Since Wiesner's proposal, many applications of quantum information to cryptography have been proposed, including quantum key distribution (QKD) [BB87], randomness expansion [Col09, CY14, BCM+18], quantum copy protection [Aar09, AL21, ALL+21, CLLZ21], quantum one-time programs [BGS13], and much more.

*University of Texas at Austin. Email: jiahui@utexas.edu
†Linux Foundation & Fujitsu Research. Email: hart.montgomery@gmail.com
‡NTT Research. Email: mzhandry@gmail.com

Throughout the development of quantum cryptography, quantum money has remained a central object, at least implicitly. Indeed, the techniques used for quantum money are closely related to those used in other applications. For example, the first message in the BB84 quantum QKD protocol [BB87] is exactly a banknote in Wiesner's scheme. The techniques used by [BCM+18] to prove quantumness using classical communication have been used to construct quantum money with classical communication [RS19]. The subspace states used by [AC12] to construct quantum money were recently used to build quantum copy protection [ALL+21].

The Public Verification Barrier. Wiesner's scheme is only privately verifiable, meaning that the mint is needed to verify. This results in numerous weaknesses. Improper verification opens the scheme to active attacks [Lut10]. Moreover, private verification is not scalable, as the mint would be required to participate in every single transaction. Wiesner's scheme also requires essentially perfect quantum storage, since otherwise banknotes in Wiesner's scheme will quickly decohere and be lost.

All these problems are readily solved with *publicly verifiable* quantum money[1], where anyone can verify, despite the mint being the sole entity that can mint notes. Public verification immediately eliminates active attacks, and solves the scaling problem since the transacting users can verify the money for themselves. Aaronson and Christiano [AC12] also explain that public verifiability allows for also correcting any decoherance, so users can keep their banknotes alive indefinitely.

Unfortunately, constructing convincing publicly verifiable quantum money has become a notoriously hard open question. Firstly, some natural modifications to Wiesner's quantum money scheme will not give security under public verification [FGH+10]. Aaronson [Aar09], and later Aaronson and Christiano [AC12] gave publicly verifiable quantum money relative to quantum and classical oracles, respectively. Such oracle constructions have the advantage of provable security, but it is often unclear how to instantiate them in the real world[2]: in both [Aar09] and [AC12], "candidate" instantiations were proposed, but were later broken [LAF+10, CPDDF+19]. Another candidate by Zhandry [Zha19] was broken by Roberts [Rob21]. Other candidates have been proposed [FGH+12, Kan18, KSS21], but they all rely on new, untested assumptions that have received little cryptanalysis effort. The one exception, suggested by [BDS16] and proved by [Zha19], uses indistinguishability obfuscation (iO) to instantiate Aaronson and Christiano's scheme [AC12]. Unfortunately, the post-quantum security of iO remains poorly understood, with all known constructions of post-quantum iO [GGH15, BGMZ18, BDGM20, WW21] being best labeled as candidates, lacking justification under widely studied assumptions.

Thus, it remains a major open question to construct publicly verifiable quantum money from standard cryptographic tools. Two such post-quantum tools we will investigate in this work are the two most influential and well-studied: lattices and isogenies over elliptic curves.

This public verification barrier is inherited by many proposed applications of quantum cryptography. For example, quantum copy protection for any function whose outputs can be verified immediately implies a publicly verifiable quantum money scheme. As such, all such constructions in the standard model [ALL+21, CLLZ21] require at a minimum a computational assumption that implies quantum money.[3]

[1] Sometimes it is also referred to as public-key quantum money. We may use the two terms interchangeably.

[2] Quantum oracles are quantum circuits accessible only as a black-box unitary. They are generally considered as strong relativizing tools when used in proofs. Classical oracles are black-box classical circuits, a much weaker tool.

[3] This holds true even for certain weaker versions such as copy *detection*, also known as infinite term secure software leasing.

Quantum Money Decentralized: Quantum Lightning An even more ambitious goal is a publicly verifiable quantum money where the bank/mint itself should *not* be capable of duplicating money states. To guarantee unclonability, the scheme should have a "collision-resistant" flavor: no one can (efficiently) generate two valid money states with the same serial number. This notion of quantum money appeared as early in [LAF+10]; the name "quantum lightning" was given in [Zha19].

Quantum lightning has broader and more exciting applications: as discussed in [Zha19, Col19, CS20, AGKZ20], it can be leveraged as verifiable min-entropy, useful building blocks to enhance blockchain/smart contract protocols and moreover, it could lead to decentralized cryptocurrency without a blockchain.

Quantum money has a provably secure construction from iO, a strong cryptographic hammer but still a widely used assumption. On the other hand, quantum lightning from even *relatively standard-looking* assumptions remains open. Some existing constructions [Kan18, KSS21] use strong oracles such as quantum oracles, with conjectured instantiations that did not go through too much cryptanalysis. [FGH+12] is another candidate built from conjectures in knot theory. But a correctness proof and security reduction are not provided in their paper.

Collapsing vs. Non-Collapsing With a close relationship to quantum money, collapsing functions [Unr16] are a central concept in quantum cryptography. A collapsing function f says that one should not be able to distinguish a superposition of pre-images $\frac{|x_1\rangle+|x_2\rangle\cdots|x_k\rangle}{\sqrt{k}}$, from a measured pre-image $|x_i\rangle, i \in [k]$ for some image $y = f(x_i)$, for all $i \in [k]$.

While collapsing functions give rise to secure post-quantum cryptography like commitment schemes, its precise opposite is necessary for quantum money: if no verification can distinguish a money state in a superposition of many supports from its measured state, a simple forgery comes ahead. Hence, investigating the collapsing/non-collapsing properties of hash functions from lattices and isogenies will provide a win-win insight into quantum money and post-quantum security of existing cryptographic primitives.

2 Our Results

In this work, we give both negative and positive results for publicly verifiable quantum money.

Breaking Quantum Money. Very recent work by Khesin, Lu, and Shor [KLS22] claims to construct publicly verifiable quantum money from the hardness of worst-case lattice problems, a standard assumption. Our first contribution is to identify a fatal flaw in their security proof, and moreover show how to exploit this flaw to forge unlimited money. After communicating this flaw and attack, the authors of [KLS22] have retracted their paper.[4]

More importantly, we show that a general class of *natural* money schemes based on lattices *cannot* be both secure and publicly verifiable. We consider protocols where the public key is a short wide matrix $\mathbf{A}^T$, and a banknote with serial number $\mathbf{u}$ is a superposition of "short" vectors $\mathbf{y}$ such that $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{u} \bmod q$. Our attack works whenever $\mathbf{A}^T$ is uniformly random. We also generalize this to handle the case where $\mathbf{A}^T$ is uniform conditioned on having a few public short vectors in

[4] We thank the authors of [KLS22] for patiently answering our numerous questions about their work, which was instrumental in helping us identify the flaw.

its kernel. This generalization includes the Khesin-Lu-Shor scheme as a special case. Our result provides a significant barrier to constructing quantum money from lattices.

Along the way, we prove that the SIS hash function is *collapsing* [Unr16] for all moduli, resolving an important open question in the security of post-quantum hash functions.[5]

Invariant Money/Lightning. To complement our negative result, we propose a new framework for building quantum money, based on invariants. Our framework abstracts some of the ideas behind the candidate quantum money from knots in [FGH+12] and behind [LAF+10]. Our main contributions here are two-fold:

- We propose a (classical) oracle construction that implements our framework assuming the existence of a quantum-secure cryptographic group action and a relatively modest assumption about *generic* cryptographic group actions. We then give proposals for instantiating our invariant framework on more concrete assumptions. The first is based on isogenies over elliptic curves[6]; the second is based on rerandomizable functional encryption with certain properties; finally, we also discuss the quantum money from knots construction in [FGH+12] with some modifications.

- In order to gain confidence in our proposals, we for the first time formalize abstract properties of the invariant money under which security can be proved. Concretely, we prove that a certain mixing condition is sufficient to characterize the states accepted by the verifier, and in particular prove correctness[7]. We also propose "knowledge of path" security properties for abstract invariant structures which would be sufficient to justify security. These knowledge of path assumptions are analogs of the "knowledge of exponent" assumption on groups proposed by Damgård [Dam92]. Under these assumptions, we are even able to show that the invariants give quantum *lightning* [LAF+10, Zha19], the aforementioned strengthening of quantum money that is known to have additional applications.

  Note that the knowledge of exponent assumption in groups is quantumly broken on groups due to the discrete logarithm being easy. However, for many of our assumptions, which are at least conjectured to be quantum-secure, the analogous knowledge of path assumption appears plausible, though certainly more cryptanalysis is needed to gain confidence. The main advantage of our proposed knowledge of path assumption is that it provides a concrete cryptographic property that cryptographers can study and analyze with a well-studied classical analog.

3 Technical Overview

3.1 How to Not Build Quantum Money from Lattices

We first describe a natural attempt to construct quantum money from lattices, which was folklore but first outlined by Zhandry [Zha19]. The public key will contain a random tall matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}, m \gg$

[5] Previously, [LZ19] showed that SIS was collapsing for a super-polynomial modulus.

[6] The recent attacks [CD22a, MM22a, Rob22] on SIDH do not apply to the isogeny building blocks we need. We will elaborate in the C and E.1

[7] [FGH+12] did not analyze correctness of their knot-based proposal, nor analyze the states accepted by their verifier and formalize the property needed for a security proof. [LAF+10] had informal correctness analysis on their proposal, but also did not analyze the security property needed.

n. To mint a banknote, first generate a superposition $|\psi\rangle = \sum_{\mathbf{y}} \alpha_{\mathbf{y}} |\mathbf{y}\rangle$ of short vectors $\mathbf{y} \in \mathbb{Z}^m$, such that $|\mathbf{y}| \ll q$. A natural $|\psi\rangle$ is the discrete-Gaussian-weighted state, where $\alpha_{\mathbf{y}} \propto \sqrt{e^{-\pi|\mathbf{y}|^2/\sigma^2}}$ for a width parameter σ. Then compute in superposition and measure the output of the map $\mathbf{y} \mapsto \mathbf{A}^T \cdot \mathbf{y} \bmod q$, obtaining $\mathbf{u} \in \mathbb{Z}_q^n$. The state collapses to:

$$|\psi_{\mathbf{u}}\rangle \propto \sum_{\mathbf{y}:\mathbf{A}^T\cdot\mathbf{y}=\mathbf{u}} \alpha_{\mathbf{y}} |\mathbf{y}\rangle \ .$$

This will be the money state, and $\mathbf{u}$ will be the serial number. This state can presumably not be copied: if one could construct two copies of $|\psi_{\mathbf{u}}\rangle$, then one could measure both, obtaining two short vectors $\mathbf{y}, \mathbf{y}'$ with the same coset $\mathbf{u}$. As $|\psi_{\mathbf{u}}\rangle$ is a superposition of many vectors (since $m \gg n$), with high probability $\mathbf{y} \neq \mathbf{y}'$. Subtracting gives a short vector $\mathbf{y} - \mathbf{y}'$ such that $\mathbf{A}^T \cdot (\mathbf{y} - \mathbf{y}') = 0$, solving the Short Integer Solution (SIS) problem. SIS is presumably hard, and this hardness can be justified based on the hardness of worst-case lattice problems such as the approximate Shortest Vector Problem (SVP).

The challenge is: how to verify $|\psi_{\mathbf{u}}\rangle$? Certainly, one can verify that the support of a state is only short vectors $\mathbf{y}$ such that $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{u}$. But this alone is not enough: one can fool such a verification by any *classical* $\mathbf{y}$ in the support of $|\psi_{\mathbf{u}}\rangle$. To forge then, an adversary simply measures $|\psi_{\mathbf{u}}\rangle$ to obtain $\mathbf{y}$, and then copies $\mathbf{y}$ as many times as it likes.

To get the scheme to work, then, one needs a verifier that can distinguish classical $\mathbf{y}$ from superpositions. This is a typical challenge in designing publicly verifiable money schemes. A typical approach is to perform the quantum Fourier transform (QFT): the QFT of $\mathbf{y}$ will result in a uniform string, whereas the QFT of $|\psi_{\mathbf{u}}\rangle$ will presumably have structure. Indeed, if $|\psi_{\mathbf{u}}\rangle$ is the Gaussian superposition, following ideas of Regev [Reg05], the QFT of $|\psi_{\mathbf{u}}\rangle$ will be statistically close to a superposition of samples $\mathbf{A} \cdot \mathbf{r} + \mathbf{e}$, where $\mathbf{r}$ is uniform in $\mathbb{Z}_q^n$, and $\mathbf{e} \in \mathbb{Z}_q^m$ is another discrete Gaussian of width q/σ. The goal then is to distinguish such samples from uniform.

Unfortunately, such distinguishing is likely hard, as this task is the famous (decisional) Learning with Errors (LWE) problem. LWE is presumably hard, which can be justified based on the hardness of the same worst-case lattice problems as with SIS, namely SVP. So either LWE is hard, or the quantum money scheme is insecure in the first place.

Nevertheless, this leaves open a number of possible strategies for designing quantum money from lattices, including:

1. What if non-Gaussian $|\psi\rangle$ is chosen?
2. What if distinguishing is not done via the QFT but some other quantum process?
3. What if we somehow make LWE easy?

The first significant barrier beyond the hardness of LWE is due to Liu and Zhandry [LZ19]. They show that, if the modulus q is super-polynomial, then the map $\mathbf{y} \mapsto \mathbf{A}^T \cdot \mathbf{y}$ for a random $\mathbf{A}$ is *collapsing* [Unr16]: that is, for *any* starting state $|\psi_{\mathbf{u}}\rangle$ of short vectors, distinguishing $|\psi_{\mathbf{u}}\rangle$ from $\mathbf{y}$ is infeasible for *any* efficient verification process. Collapsing is the preferred notion of post-quantum security for hash functions, as it is known that collision resistance is often not sufficient for applications when quantum adversaries are considered.

The result of [LZ19] follows from the hardness of LWE (which is quantumly equivalent to SIS [Reg05]), albeit with a noise rate super-polynomially smaller than q/σ which is a stronger

assumption than the hardness with rate q/σ. Moreover, their result requires q to be super-polynomially larger than σ. In practice, one usually wants q to be polynomial, and the result of [LZ19] leaves open the possibility of building quantum money in such a setting.

What about making LWE easy (while SIS remains hard)? The usual approach in the lattice literature to making decisional LWE easy is to output a short vector $\mathbf{s}$ in the kernel of $\mathbf{A}^T$. If $|\mathbf{s}| \ll (q/\sigma)$, this allows for distinguishing LWE samples from uniform, since $\mathbf{s} \cdot (\mathbf{A} \cdot \mathbf{r} + \mathbf{e}) = \mathbf{s} \cdot \mathbf{e}$, which will be small relative to q, while $\mathbf{s} \cdot \mathbf{x}$ for uniform $\mathbf{x}$ will be uniform in $\mathbb{Z}_q$. Unfortunately, adding such short vectors breaks the security proof, since $\mathbf{s}$ is a SIS solution, solving SIS is trivially easy by outputting $\mathbf{s}$. To revive the security, one can try reducing to the 1-SIS problem, which is to find a short SIS solution that is linearly independent of $\mathbf{s}$. 1-SIS can be proved hard based on the same worst-case lattice problems as SIS [BF11]. However, in the scheme above, it is not clear if measuring two forgeries and taking the difference should result in a vector linearly independent of $\mathbf{s}$.

The Recent Work of [KLS22]. Very recently, Khesin, Lu, and Shor [KLS22] attempt to provide a quantum money scheme based on lattices. Their scheme has some similarities to the blueprint discussed above, taking advantage of each of the strategies 1, 2 and 3. But there are other differences as well: the state $|\psi\rangle$ is created as a superposition over a lattice rather than the integers, and the measurement of $\mathbf{u}$ is replaced with a move complex general positive operator-value measurement (POVM). [KLS22] claims to prove security under the hardness of finding a second short vector in a random lattice when already given a short vector. This problem is closely related to 1-SIS, and follows also from the hardness of worst-case lattice problems.

Our Results. First, we show an alternative view of [KLS22] which shows that it does, indeed, fall in the above framework. That is, there is a way to view their scheme as starting from $|\psi\rangle$ that is a non-Gaussian superposition of short integer vectors $\mathbf{y}$. The minting process in our alternate view then measures $\mathbf{A}^T \cdot \mathbf{y}$, where $\mathbf{A}$ is part of the public key, and is chosen to be uniform except that it is orthogonal to 3 short vectors $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2$. These vectors play a role in verification, as they make the QFT non-uniform. Using this alternative view, we also demonstrate a flaw in the security proof of [KLS22], showing that forged money states actually do not yield new short vectors in the lattice. See Section 6 for details.

We then go on (Section 7) to show an explicit attack against their money scheme. More generally, we show an attack on a wide class of instantiations of the above framework. Our attack works in two steps:

- First, we extend the collapsing result of [LZ19] to also handle the case of polynomial modulus, and in particular, we only need LWE to be hard for noise rate that is slightly smaller than q/σ. This resolves an important open by showing that SIS is collapsing for all moduli.

  Our proof requires a novel reduction that exploits a more delicate analysis of the quantum states produced in the proof of [LZ19]. We also extend the result in a meaningful way to the case where several short kernel vectors $\mathbf{s}_0, \mathbf{s}_1, \ldots$ are provided. We show that instead of just using $\mathbf{y}$ as a forgery (which can be distinguished using the short vectors $\mathbf{s}_i$), a particular superposition over vectors of the form $\mathbf{y} + \sum_i c_i \mathbf{s}_i$ can fool any efficient verification. Fooling verification requires the hardness a certain "k-LWE" problem, which we show follows from worst-case lattice problems in many settings (see Section B). This requires us to extend the known results on k-LWE hardness, which may be of independent interest.

- Then we show how to construct such a superposition efficiently given only $\mathbf{y}$ and the $\mathbf{s}_i$, in many natural settings. Our settings include as a special case the setting of [KLS22]. Along the way, we explain how to construct Gaussian superpositions over lattices, when given a short basis. The algorithm is a coherent version of the classical discrete Gaussian sampling algorithm [GPV08]. In general, it is not possible to take a classical distribution and run it on a superposition of random coins to get a superposition with weights determined by the distribution. This is because the random coins themselves will be left behind and entangled with the resulting state. We show how to implement the classical algorithm coherently in a way that does not leave the random coins behind or any other entangled bits. Such an algorithm was previously folklore (e.g. it was claimed to exist without justification by [KLS22]), but we take care to actually write out the algorithm.

After communicating this flaw and attack to the authors of [KLS22], they have retracted their paper.[8]

3.2 Quantum Money from Walkable Invariants

In the second part of the paper, we describe a general framework for instantiating publicly verifiable quantum money from invariants satisfying certain conditions. This framework abstracts the ideas behind the construction of quantum money from knots [FGH+12] and its precedent [LAF+10].

At a high level, we start from a set X, which is partitioned into many disjoint sets $O \subseteq X$. There is a collection of efficiently computable (and efficiently invertible) permutations on X, such that for every permutation in the collection and every O in the partition, the permutation maps elements of O to O. Such a set of permutations allows one to take an element $x \in O$, and perform a walk through O. We additionally assume an invariant $I : X \to Y$ on X, such that I is constant on each element O of the partition. In other words, I is invariant under action by the collection of permutations.

In the case of [FGH+12], X is essentially the set of knot diagrams[9], the permutations are Reidemeister moves, and the invariant is the Alexander polynomial.

An honest quantum money state will essentially be a uniform superposition over O [10]. Such a state is constructed by first constructing the uniform superposition over X, and then measuring the invariant I. Applying a permutation from the collection will not affect such a state. Thus, verification attempts to test whether the state is preserved under action by permutations in the collection by performing an analog of a swap test, and only accepts if the test passes.

In [FGH+12], it is explained why certain attack strategies are likely to be incapable of duplicating banknotes. However, no security proof is given under widely believed hard computational assumptions. To make matters worse, [FGH+12] do not analyze what types of states are accepted by the verifier. It could be, for example, that duplicating a banknote perfectly is computationally infeasible, but there are fake banknotes that pass verification that can be duplicated; this is exactly what happens in the lattice-based schemes analyzed above in Section 3.1. Given the complexities of their scheme, there have been limited efforts to understand the security of the scheme. This is

[8] We once again want to emphasize that the authors of [KLS22] were exceptionally helpful and we thank them for their time spent helping us understand their work.

[9] Due to certain concerns about security, [FGH+12] actually sets X to contain extra information beyond a knot diagram.

[10] Technically, it is a uniform superposition over the pre-images of some y in the image of I. If multiple O have the same y, then the superposition will be over all such O.

problematic, since there have been many candidates for public key quantum money that were later found to be insecure.

Generally, a fundamental issue with public key quantum money schemes is that, while quantum money schemes rely on the no-cloning principle, the no-cloning theorem is information-theoretic, whereas publicly verifiable quantum money is always information-theoretically clonable. So unclonability crucially relies on the adversary being computationally efficient. Such computational unclonability is far less understood than traditional computational tasks. Indeed, while there have been a number of candidate post-quantum hard computational tasks, there are very few quantum money schemes still standing. The challenge is in understanding if and how quantum information combines with computational bounds to give computational unclonability.

To overcome this challenge, the security analysis should be broken into two parts: one part that relies on *information-theoretic* no-cloning, and another part that relies on a computational hardness assumption. Of course, the security of the scheme itself could be such an assumption, so we want to make the assumption have nothing to do with cloning. One way to accomplish this is to have the assumption have classical inputs and outputs (which we will call "classically meaningful"), so that it could in principle be falsified by a classical algorithm, which are obviously not subject to quantum unclonability. Separating out the quantum information from the computational aspects would hopefully give a clearer understanding of why the scheme should be unclonable, hopefully allow for higher confidence in security. Moreover, as essentially all widely studied assumptions are classically meaningful, any attempt to prove security under a widely studied assumptions would have to follow this blueprint, and indeed the proof of quantum money from obfuscation [Zha19] is of this form.

Our Results. In this work, we make progress towards justifying invariant-based quantum money.

- First, we prove that if a random walk induced by the collection of permutations mixes, then we can completely characterize the states accepted by verification. The states are exactly the uniform superpositions over O [11]. Unfortunately, it is unclear if the knot construction actually mixes, and any formal proof of mixing seems likely to advance knot theory[12].

- Second, we provide concrete security properties under which we can prove security. These properties, while still not well-studied, at least have no obvious connection to cloning, and are meaningful even classically. Under these assumptions, we can even prove that the schemes are in fact quantum lightning, the aforementioned strengthening of quantum money where not even the mint can create two banknotes of the same serial number.

Our Hardness Assumptions We rely on two hardness assumptions in our invariant money scheme for a provably secure: the *path-finding* assumption and *knowledge of path finding* assumption.

Informally speaking, the path-finding assumption states that, given some adversarially sampled x from a set of elements X and given a set of "permutations" Σ, it is hard for any efficient adversary, given a random $z \in X$, where there exists some $\sigma \in \Sigma$ such that $\sigma(x) = z$, to find such a σ. One can observe that it is similar to a "discrete logarithm" style of problem. Even though we cannot use

[11] Or more generally, if multiple O have the same y, then accepting states are exactly those that place equal weight on elements of each O, but the weights may be different across different O

[12] Nevertheless we provide a discussion on the knot money instantiation in F.

discrete logarithm due to its quantum insecurity, we have similar hard problems in certain isogenies over ellitic curves, abstracted as "group action discrete logarithm" problems [ADMP20].

Our Knowledge of Path Assumptions. The main novel assumption we use is a "knowledge of path" assumption. This roughly says that if an algorithm outputs two elements x, z in the same O, then it must "know" a path between them: a list of permutations from the collection that, when composed, would take x to z. While such a knowledge of path assumption is undoubtedly a strong assumption, it seems plausible in a number of relevant contexts (e.g. elliptic curve isogenies that have no known non-trivial attacks or "generic" group actions).

Formalizing the knowledge of path assumption is non-trivial. The obvious *classical* way to define knowledge of path is to say that for any adversary, there is an extractor that can compute the path between x and z. Importantly, the extractor must be given the same random coins as the adversary, so that it can compute x and z for itself and moreover know what random choices the adversary made that lead to x, z. Essentially, by also giving the random coins, we would be effectively making the adversary deterministic, which is crucial for the extractor's output to be related to the adversary's output.

Unfortunately, quantumly the above argument does not make much sense, as quantum algorithms can have randomness without having explicit random coins. In fact, there are quantum procedures that are *inherently probabilistic*, in the sense that the process is efficient, but there is no way to run the process twice and get the same outcome both times. This is actually crucial to our setting: we are targeting the stronger quantum lightning, which means that even the mint cannot create two banknotes with the same serial number. This means that the minting process is inherently probabilistic. The adversary could, for example, run the minting process, but with its own minting key. Such an adversary would then be inherently probabilistic and we absolutely would need a definition that can handle such adversaries.

Our solution is to exploit the fact that quantum algorithms can always be implemented *reversibly*. We then observe that with a classical reversible adversary, an equivalent way to define knowledge assumptions would be to just feed the entire *final* state of the adversary (including output) into the extractor. By reversibility, this is equivalent to giving the input, coins included, to the extractor. But this alternate extraction notion actually *does* make sense quantumly. Thus our knowledge of path assumption is defined as giving the extractor the entire final (quantum) state of the reversible adversary, and asking that the extractor can find a path between x and z. This assumption allows us to bypass the issue of inherently probabilistic algorithms, and is sufficient for us to prove security.

Instantiations of Invariant Quantum Money and Lightning After we provide the characterization of security needed for invariant money, we discuss four candidate instantiations[13]:

- We show a construction from structured oracles and generic cryptographic group actions. Notably, while we do not know how to instantiate these oracles, we can prove that this construction is secure assuming the existence of a cryptographic group action and the assumption that the knowledge of path assumption holds over a generic cryptographic group action.[14]

[13]Throughout the sections on invariant quantum money framework and construction 8, C, D, E and F, we will sometimes interchangeably use "money" or "lightning". But in fact the proposed candidates are all candidates for quantum lightning.

[14]This seems like a very plausible assumption to us: classically, the knowledge of exponent would almost trivially hold over generic groups.

- We explain how re-randomizable functional encryption, a type of functional encryption with special properties that seem reasonable, can be used to build another candidate quantum lightning. We don't currently have a provably secure construction from standard cryptographic assumptions for this special re-randomizable functional encryption, but we provide a candidate construction based on some relatively well-studied primitives.
- Elliptic curve isogenies are our final new candidate instantiation. We outline how, given some assumptions about sampling certain superpositions of elliptic curves, it may be possible to build quantum lightning from isogeny-based assumptions.
- Finally, we analyze the construction of quantum money from knots in [FGH+12] in our framework.

For all these three constructions, we show that their corresponding *path-finding* problem between two elements x, z in the same O is relatively straightforward to study (reducible to reasonably well-founded assumptions). Nevertheless, we need the knowledge of path assumptions to show that we can extract these paths from a (unitary) adversary. We believe that one may show a knowledge-of-path property when replacing some plain model components in the above candidates with (quantum accessible) *classical* oracles, thus giving the possibility for a first quantum lightning scheme relative to only classical oracles and widely studied assumptions.

4 Related Work and Discussion

Other Related Work. In addition to the related work we have discussed in the introduction, we would like to mention some other relevant work.

The schemes [AGKZ20] and [Shm22] can be viewed as quantum lightning with an interactive minting procedure: a multi-round interactive protocol between the bank and the user is needed to create a valid money state. [AGKZ20] is based on classical oracles and [Shm22] is based on (subexponential) iO. One may argue that "compressing" their interaction might lead to a standard quantum lightning protocol. However, after investigating the constructions, we believe that showing the security for such "compressed" protocols leads us back to the land of highly non-standard assumptions.

Semi-quantum money is a notion put forward in [RS19]. More precisely, the authors construct an interactive private-key quantum money scheme verifiable through classical interactions, built from LWE, with ideas from classically-verifiable proof of quantumness [BCM+18]. Later, the idea was extended to a notion in-between public and private key quantum money, which is called *two-tier quantum lightning* in [KNY21]. Due to the essential structure of the proof-of-quantumness protocol on which these two schemes are based, they cannot be made publicly verifiable.

Regarding non-collapsing functions: it has been shown that some "natural" classes of hash functions are collapsing [Unr16, LZ19, Zha22, CX22]. [ARU14] constructed non-collapsing functions using quantum oracles.

For the cryptanalysis on existing quantum money proposals: [BDG22] shows a quantum reduction from the conjecture in [KSS21] to a linear algebra problem, which is the only cryptanalysis work we know of on a quantum money scheme that is not broken yet.

Discussion and Open Problems. In this paper, we aimed to investigate the feasibility of quantum money and, in particular, quantum lightning. While we hope that readers believe our work helps shed light on the subject, we still believe that this is a wide open area for study with important applications once we reach a world where quantum computers proliferate.

In particular, fully settling the question of whether or not it is possible to build quantum lightning from lattice-based assumptions would be a very exciting result. Our paper rules out (arguably) the most natural class of schemes, but that does not mean that a less natural lattice-based quantum lightning scheme could exist.

5 Preliminaries

In this section we explain some background material needed for our work.

For quantum notations, we denote $|\cdot\rangle$ as the notation for a pure state and $|\cdot\rangle\langle\cdot|$ for its density matrix. ρ denotes a general mixed state.

We will go over some fundamental lattice facts and then move to quantum money definitions. Due to the restriction of space, we leave some additional lattice basics, hardness theorems and necessary quantum background (in particular related to lattices) to Appendix A.

5.1 Lattice Basics

We say a distribution $\mathcal{D}$ is (B, δ)-bounded if the probability that $\mathcal{D}$ outputs a value larger than B is less than δ. We extend this to distributions that output vectors in an entry-by-entry way. Given a set of vectors $\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n\}$, we define the norm of $\mathbf{B}$, denoted $||\mathbf{B}||$, as the length of the longest vector in $\mathbf{B}$, so $||\mathbf{B}|| = \max_i ||\mathbf{b}_i||$. For any lattice Λ, we define the minimum distance (or first successive minimum) $\lambda_1(\Lambda)$ as the length of the shortest nonzero lattice vector in Λ.

We next define a the *Gram-Schmidt basis* and the *Gram-Schmidt norm* based on the definitions of [GPV08].

Definition 1. Gram-Schmidt Basis. For any (ordered) set $\mathbf{S} = \{\mathbf{s}_1, ..., \mathbf{s}_n\} \subset \mathbb{R}^n$ of linearly independent vectors, let $\tilde{S} = \{\tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_n\}$ denote its Gram-Schmidt orthogonalization, defined iteratively in the following way: $\tilde{\mathbf{s}}_1 = \mathbf{s}_1$, and for each $i \in [2, n]$, $\tilde{\mathbf{s}}_i$ is the component of $\mathbf{s}_i$ orthogonal to $span(\mathbf{s}_1, ..., \mathbf{s}_{i-1})$.

We next define discrete Gaussians formally. Since we later use their lemmas, our definition is loosely based on that of [BLP+13].

Definition 2. For any $\sigma > 0$, the n-dimensional Gaussian function $\rho_\sigma : \mathbb{R}^n \to [0, 1]$ is defined as

$$\rho_r(\mathbf{x}) = e^{-\pi \frac{\mathbf{x}^2}{\sigma^2}}$$

We define the *discrete Gaussian function* with parameter σ at point $\mathbf{p} \in \mathbb{R}^n$, which we usually denote $\mathcal{D}_{\mathbf{\Psi}_\sigma}$ or just $\mathbf{\Psi}_\sigma$ when the context is clear, as the function over all of the integers $\mathbf{y} \in \mathbb{Z}^n$ such that the probability mass of any $\mathbf{y}$ is proportional to

$$e^{-\pi \frac{(\mathbf{p}-\mathbf{y})^2}{\sigma^2}}.$$

We can also define more complicated discrete Gaussians over lattices. In this case, let $\mathbf{\Sigma}$ be a matrix in $\mathbb{R}^{n\times n}$. The discrete Gaussian over a lattice Λ with center $\mathbf{p}$ and "skew" parameter Σ is the function over all *lattice points* in Λ such that the probability mass of any $\mathbf{y}$ is proportional to

$$e^{-\pi(\mathbf{p}-\mathbf{y})^T\left(\mathbf{\Sigma}\mathbf{\Sigma}^\mathbf{T}\right)^{-1}(\mathbf{p}-\mathbf{y})},$$

very similar to as before. We usually denote this type of discrete Gaussian as $\mathbf{\Psi}_{\Lambda,\mathbf{\Sigma},\mathbf{p}}$ or $\mathcal{D}_{\mathbf{\Psi}_{\Lambda,\mathbf{\Sigma},\mathbf{p}}}$, where we sometimes substitute σ for $\mathbf{\Sigma}$ when $\mathbf{\Sigma} = \sigma \cdot \mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix. We also sometimes omit parameters when they are obvious (e.g. 0) in context.

5.2 (Lattice-relevant) Quantum Facts

All quantum notations used in this work are relatively basic and standard; we therefore omit extensive preliminaries on them. For further quantum basics, we refer the readers to [NC02].

5.2.1 Gaussian Superposition Preparation

We now show that it is possible to efficiently sample a discrete Gaussian quantumly over a lattice basis.

Generating Gaussian Superpositions. Let $\mathcal{L}$ be a lattice. Given a vector $\mathbf{c}$ (not necessarily in $\mathcal{L}$) and covariance matrix Σ, define

$$|\mathbf{\Psi}_{\mathcal{L},\Sigma,\mathbf{c}}\rangle \propto \sum_{\mathbf{x}\in\mathcal{L}} \sqrt{e^{-\pi(\mathbf{x}-\mathbf{c})^T\cdot\Sigma^{-1}\cdot(\mathbf{x}-\mathbf{c})}}|\mathbf{x}\rangle$$

This is the discrete-Gaussian-weighted superposition over lattice vectors.

Theorem 1. *There is a QPT algorithm which, given a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ for a lattice $\mathcal{L}$, a center $\mathbf{c}$, and covariance matrix Σ such that $\mathbf{b}_i \cdot \Sigma^{-1} \cdot \mathbf{b}_i \leq 1/\omega(\log\lambda)$, constructs a state negligibly close to $|\mathbf{\Psi}_{\mathcal{L},\Sigma,\mathbf{c}}\rangle$.*

We prove Theorem 1 by gradually building up from special cases.

Lemma 2. *There is a QPT algorithm which, given $c \in \mathbb{Z}$ and $\sigma \geq \omega(\sqrt{\log\lambda})$, constructs a state negligibly close to $|\mathbf{\Psi}_{\mathbb{Z},\sigma^2,c}\rangle$.*

Proof. This is a straightforward adaptation of the classical algorithm for sampling from the discrete Gaussian over integers [GPV08]. Let $t \geq \omega(\sqrt{\log\lambda})$. The algorithm proceeds in the following steps:

1. Let $\mathcal{I} = \mathbb{Z} \cap [c - t\sigma, c + t\sigma]$. Let x_{min} be the minimal element of $\mathcal{I}$, and $w = |\mathcal{I}|$.
2. Initialize a register to $|0\rangle$. Then using the QFT on w elements, construct the state $\propto \sum_{i=0}^{w-1} |i\rangle$.
3. By adding x_{min} in superposition, construct the state $\propto \sum_{i\in\mathcal{I}} |i\rangle$
4. Now apply in superposition the map $|i\rangle \mapsto |i\rangle \otimes \left(\sqrt{e^{-\pi(i-c)^2/\sigma^2}}|0\rangle + \sqrt{1 - e^{-\pi(i-c)^2/\sigma^2}}|1\rangle\right)$.
5. Measure the second register, obtaining a bit b; the first register collapses to a state $|\mathbf{\Psi}\rangle$. If $b = 0$, output $|\mathbf{\Psi}\rangle$. Otherwise, discard $|\mathbf{\Psi}\rangle$ and restart from Step 2.

Note that the state outputted by the above algorithm is $|\Psi\rangle \propto \sum_{i\in\mathcal{I}} \sqrt{e^{-\pi(i-c)^2/\sigma^2}}|i\rangle$. This is identical to $|\Psi_{\mathbb{Z},\sigma^2,c}\rangle$, except that the support is truncated to the interval $\mathcal{I}$ (and therefore the state is also re-scaled, but the proportions for $i \in \mathcal{I}$ are identical). [GPV08] show that the analogous distributions over i are negligibly close ([GPV08], Lemma 4.3). An almost identical argument shows that the states $|\Psi_{\mathbb{Z},\sigma^2,c}\rangle$ and $|\Psi\rangle$ are negligibly close. □

Lemma 3. *There is a QPT algorithm which, given a basis* $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ *for a lattice* $\mathcal{L}$*, a center* $\mathbf{c} \in \mathbb{Z}^n$*, and* $\sigma \geq \|\mathbf{B}\|\omega(\sqrt{\log \lambda})$*, constructs a state negligibly close to* $|\Psi_{\mathcal{L},\sigma^2,\mathbf{c}}\rangle$*.*

Proof. This is also a straightforward adaptation of the classical algorithm for sampling from discrete Gaussians over lattices. However, care is needed to ensure that there is no spurious information left behind; such spurious information would not affect classical sampling, but could be entangled with the resulting quantum state, thereby perturbing it. We show that the algorithm can be implemented without such perturbation.

The classical sampling algorithm ([GPV08], Section 4.2) works as follows:

1. Initialize $\mathbf{v} \leftarrow 0$. Then for $i = 1, \ldots, m$ (m being the dimension of $\mathcal{L}$), do:
   (a) Let $c'_i = \frac{(\mathbf{c}-\mathbf{v})\cdot\tilde{\mathbf{b}}_i}{|\tilde{\mathbf{b}}_{1i}|^2}$ and $\sigma'_i = \sigma/|\tilde{\mathbf{b}}_{1i}|$.
   (b) Sample $z_i \leftarrow D_{\mathbb{Z},\sigma'_i,c'_i}$
   (c) Update $\mathbf{v} \leftarrow \mathbf{v} + z_i\mathbf{b}_i$.

Then [GPV08] proves that the output distribution is statistically close to $D_{\mathcal{L},\sigma,\mathbf{c}}$. We now explain how to run the sampling algorithm coherently to produce $|\Psi_{\mathcal{L},\sigma^2,\mathbf{c}}\rangle$:

1. Initialize a register $\mathcal{V}$ to $|0\rangle$. Then for $i = 1, \ldots, m$, do:
   (a) Apply the map $|\mathbf{v}\rangle \mapsto |\mathbf{v}\rangle \otimes |c'_i\rangle$ in superposition to $\mathcal{V}$, where $c'_i = \frac{(\mathbf{c}-\mathbf{v})\cdot\tilde{\mathbf{b}}_i}{|\tilde{\mathbf{b}}_{1i}|^2}$. Also compute $\sigma'_i = \sigma/|\tilde{\mathbf{b}}_{1i}|$.
   (b) Apply the map $|\mathbf{v}\rangle \otimes |c'_i\rangle \mapsto |\mathbf{v}\rangle \otimes |c'_i\rangle \otimes |\Psi_{\mathbb{Z},(\sigma'_i)^2,c'_i}\rangle$ in superposition. Here, we assume for simplicity the ideal $|\Psi_{\mathbb{Z},(\sigma'_i)^2,c'_i}\rangle$, but we would actually use the algorithm from Lemma 2, incurring a negligible error.
   (c) Uncompute c'_i by performing the operation in Step 1a in reverse.
   (d) Apply the map $|\mathbf{v}, z_i\rangle \mapsto |\mathbf{v} + z_i\mathbf{b}_i, z_i\rangle$
   (e) Uncompute z_i, which can be computed from $\mathbf{v} + z_i\mathbf{b}_i$ via linear algebra, since z_i is just the coefficient of $\mathbf{b}_i$ when representing $\mathbf{v}$ in the basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_i\}$.

By an essentially identical analysis to [GPV08], Theorem 4.1, the resulting state is statistically close to $|\Psi_{\mathcal{L},\sigma^2,\mathbf{c}}\rangle$. □

We are now ready to prove Theorem 1, which follows as a simple corollary from Lemma 3.

Proof. Write Σ^{-1} as $\Sigma^{-1} = \mathbf{U}^T \cdot \mathbf{U}$. Let $\mathbf{B}' = \{\mathbf{b}'_1 = \mathbf{U}\cdot\mathbf{b}_1, \ldots, \mathbf{b}'_n = \mathbf{U}\cdot\mathbf{b}_n\}$, and let $\mathcal{L}'$ be the lattice generated by $\mathbf{B}'$. Let $\mathbf{c}' = \mathbf{U}\cdot\mathbf{c}$.

Now invoke Lemma 3 on $\mathbf{B}'$ to construct the state (statisticaly close to) $|\Psi_{\mathcal{L}',1,\mathbf{c}'}\rangle$. To see that $\mathbf{B}'$ satisfied the conditions of Lemma 3, observe that $\|\mathbf{B}'\| \leq \max_i |\mathbf{b}'_i|$, and we have that $|\mathbf{b}'_i|^2 = \mathbf{b}_i^T \cdot \mathbf{U}^T \cdot \mathbf{U}\mathbf{b}_i = \mathbf{b}_i \cdot \Sigma^{-1}\mathbf{b}_i \leq 1/\omega(\log\lambda)$.

Now given $|\mathbf{\Psi}_{\mathcal{L}',1,\mathbf{c}'}\rangle$, we simply apply in superposition the map $\mathbf{v} \mapsto \mathbf{U}^{-1}\mathbf{v}$, which gives us exactly. $|\mathbf{\Psi}_{\mathcal{L},\Sigma,\mathbf{c}}\rangle$. □

5.3 General LWE Definition

In this section we define basic LWE with an eye towards eventually defining k-LWE. We note that, while equivalent to the standard definitions, our definitions here are presented a little bit differently than usual in lattice cryptography. This is so that we can keep the notation more consistent with the typical quantum money and quantum algorithms presentation styles. We first provide a properly parameterized definition of the LWE problem [Reg05].

Definition 3. Learning with Errors (LWE) Problem: Let n, m, and q be integers, let $\mathcal{D}_{\mathbf{A}}$ and $\mathcal{D}_{\mathbf{r}}$ be distributions over $\mathbb{Z}_q^n$, and let $\mathcal{D}_{\Psi}$ be a distribution over $\mathbb{Z}_q^m$. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be a matrix where each row is sampled from $\mathcal{D}_{\mathbf{A}}$, let $\mathbf{r} \in \mathbb{Z}_q^n$ be a vector sampled from $\mathcal{D}_{\mathbf{r}}$, and let $\mathbf{e} \in \mathbb{Z}_q^m$ be a vector sampled from $\mathcal{D}_{\Psi}$. Finally, let $\mathbf{t} \in \mathbb{Z}_q^m$ be a uniformly random vector.

The $(n, m, q, \mathcal{D}_{\mathbf{A}}, \mathcal{D}_{\mathbf{r}}, \mathcal{D}_{\Psi})$-LWE problem is defined to be distinguishing between the following distributions:

$$(\mathbf{A}, \mathbf{A}\cdot\mathbf{r}+\mathbf{e}) \text{ and } (\mathbf{A}, \mathbf{t}).$$

5.4 The k-LWE Problem

With the LWE definition in place, we are ready to move to the actual k-LWE problem. The k-LWE problem was first formally defined in [LPSS14] and used to build traitor-tracing schemes. It extends the k-SIS assumption [BF11] which was used to build linearly homomorphic signatures. Our definition below is essentially a parameterized version of the one in [LPSS14].

Definition 4. **k-LWE Problem:** Let k, n, m, and p be integers, let $\mathcal{D}_{\mathbf{R}}$ be a distribution over $\mathbb{Z}_q^n$, and let $\mathcal{D}_{\Psi}$ and $\mathcal{D}_{\mathbf{S}}$ be distributions over $\mathbb{Z}_q^m$. Let $\mathbf{S} \in \mathbb{Z}_q^{k\times m}$ be a matrix where each *row* is selected from $\mathcal{D}_{\mathbf{S}}$.

Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be a matrix sampled uniformly from the set of matrices in $\mathbb{Z}_q^{m\times n}$ such that $\mathbf{S}\cdot\mathbf{A} = 0 \mod q$, let $\mathbf{r} \in \mathbb{Z}_q^n$ be a vector sampled from $\mathcal{D}_{\mathbf{r}}$, and let $\mathbf{e} \in \mathbb{Z}_q^m$ be a vector sampled from $\mathcal{D}_{\Psi}$. Let $\mathbf{C} \in \mathbb{Z}_q^{m\times(m-k)}$ be a basis for the set of vectors $\mathbf{v} \in \mathbb{Z}_q^m$ such that $\mathbf{S}\cdot\mathbf{v} = 0$, and let $\mathbf{r}' \in \mathbb{Z}_q^{m-k}$ be a uniformly random vector.

The $(k, n, m, q, \mathcal{D}_{\mathbf{S}}, \mathcal{D}_{\mathbf{r}}, \mathcal{D}_{\Psi})$-$k$-LWE problem is defined to be distinguishing between the following distributions:

$$(\mathbf{S}, \mathbf{A}, \mathbf{C}, \mathbf{A}\cdot\mathbf{r}+\mathbf{e}) \text{ and } (\mathbf{S}, \mathbf{A}, \mathbf{C}, \mathbf{C}\cdot\mathbf{r}'+\mathbf{e})$$

We note that k-LWE is traditionally defined in a slightly different way: usually the matrix $\mathbf{A}$ is sampled before (or jointly with) $\mathbf{S}$ rather than after it. We sample $\mathbf{S}$ first in our definition because we will need to handle very unusual (at least for cryptographic applications) distributions $\mathcal{D}_{\mathbf{S}}$.

5.5 Quantum Money and Quantum Lightning

Here, we define public key quantum money and quantum lightning. Following Aaronson and Christiano [AC12], we will only consider so-called "mini-schemes", where there is only a single banknote.

Both quantum money and quantum lightning share the same syntax and correctness requirements. There are two quantum polynomial-time algorithms $\mathsf{Gen}, \mathsf{Ver}$ such that:

- $\mathsf{Gen}(1^\lambda)$ samples a classical serial number σ and a quantum state $|\psi\rangle$.
- $\mathsf{Ver}(\sigma, |\psi\rangle)$ outputs a bit 0 or 1.

Correctness. We require that there exists a negligible function negl such that $\Pr[\mathsf{Ver}(\mathsf{Gen}(1^\lambda))] \geq 1 - \mathsf{negl}(\lambda)$.

Security. Where public key quantum money and quantum lightning differ is in security. The differences are analogous to the differences between one-way functions and collision resistance.

Definition 5 (Quantum Money Unforgeability)**.** $(\mathsf{Gen}, \mathsf{Ver})$ is secure public key quantum money if, for all quantum polynomial-time A, there exists a negligible negl such that A wins the following game with probability at most negl:

- The challenger runs $(\sigma, |\psi\rangle) \leftarrow \mathsf{Gen}(1^\lambda)$, and gives $\sigma, |\psi\rangle$ to A.
- A produces a potentially entangled joint state $\rho_{1,2}$ over two quantum registers. Let ρ_1, ρ_2 be the states of the two registers. A sends $\rho_{1,2}$ to the challenger.
- The challenger runs $b_1 \leftarrow \mathsf{Ver}(\sigma, \rho_1)$ and $b_2 \leftarrow \mathsf{Ver}(\sigma, \rho_2)$. A wins if $b_1 = b_2 = 1$.

Definition 6 (Quantum Lightning Unforgeability)**.** $(\mathsf{Gen}, \mathsf{Ver})$ is secure quantum lightning if, for all quantum polynomial-time A, there exists a negligible negl such that A wins the following game with probability at most negl:

- A, on input 1^λ, produces and sends to the challenger σ and $\rho_{1,2}$, where $\rho_{1,2}$ is a potentially entangled joint state over two quantum registers.
- The challenger runs $b_1 \leftarrow \mathsf{Ver}(\sigma, \rho_1)$ and $b_2 \leftarrow \mathsf{Ver}(\sigma, \rho_2)$. A wins if $b_1 = b_2 = 1$.

The difference between quantum lightning and quantum money is therefore that in quantum lightning, unclonability holds, even for adversarially constructed states.

Note that, as with classical collision resistance, quantum lightning does not exist against non-uniform adversaries. Like in the case of collision resistance, we can update the syntax and security definition to utilize a common reference string (crs), which which case non-uniform security can hold. For this paper, to keep the discussion simple, we will largely ignore the issue of non-uniform security.

6 The Flaw in [KLS22] Lattice-Based Quantum Money

6.1 Overview of [KLS22]

We do not describe the whole scheme here, but re-create enough of the scheme to describe our attack.

Setup. To set up the scheme, first a few parameters are specified:

- An exponentially-large prime P
- A Gaussian width $\sigma \gg \sqrt{d}P^{1/d}$
- An integer t ([KLS22] suggest $t = 3$).
- An odd positive integer $\Delta \gtrsim \sigma$.
- An integer $k \leq t\sigma\Delta$.

Next [KLS22] create a vector $\mathbf{v} = (v_1, \ldots, v_d) \in \mathbb{Z}_P^d$ such that

$$v_1 = 1 \quad , \quad v_2 = \frac{P+\Delta}{2} = \Delta/2 \bmod P$$

All the remaining entries of v are uniform in $\mathbb{Z}_P$. [KLS22] then defines the linear subspaces $\mathcal{L} = \{\mathbf{x} \in \mathbb{Z}_P^d : \mathbf{x}.\mathbf{v} = 0 \bmod P\}$ and $\mathcal{L}^\perp = \{m\mathbf{v} : m \in \mathbb{Z}_P\} \subseteq \mathbb{Z}_P^d$. Note that the authors require that there is a $\mathbf{w} \in \mathcal{L}^\perp$ such that $\mathbf{w} \cdot \mathbf{v} = 1 \bmod P$. This is equivalent to requiring that $\mathbf{v}.\mathbf{v} \neq 0 \bmod P$, which holds with overwhelming probability since $\mathbf{v}.\mathbf{v}$ is statistically close to uniform over $\mathbb{Z}_P$.

Minting Process. After a few steps, the minting process has the following state ([KLS22] page 3, Eq (8)) over registers $\mathcal{X}$,$\mathcal{U}$:

$$|\phi\rangle \approx \frac{1}{C} \sum_{\mathbf{u} \in \mathcal{L}^\perp} \sum_{\mathbf{x} \in \mathcal{L}} e^{-(x-u)^2/4\sigma^2} |\mathbf{x}\rangle |\mathbf{u}\rangle \tag{1}$$

Where C is a normalization constant.

Next, the mint maps $\mathbf{u} \in \mathcal{L}^\perp$ to $m \in \mathbb{Z}_P$ in the $\mathcal{U}$ register, where $\mathbf{u} = m\mathbf{w}$, obtaining the state close to:

$$\frac{1}{C} \sum_{m \in \mathbb{Z}_P} \sum_{\mathbf{x} \in \mathcal{L}} e^{-(x-m\mathbf{w})^2/4\sigma^2} |\mathbf{x}\rangle |m\rangle$$

Next, the mint applies a certain POVM to the $\mathcal{U}$ register. Let $\Gamma = 1/2 + t\sigma\Delta \bmod P$. The measurement is specified matrices M_T where:

$$M_T = \frac{1}{4k+2} \left(\sum_{j=-k}^{k} |T+j\rangle\langle T+j| + |T+\Gamma+j\rangle\langle T+\Gamma+j| \right)$$

[KLS22] does not explain how to realize the POVM. But the following process suffices:

- Initialize registers $\mathcal{J}$ and $\mathcal{B}$ to $\frac{1}{\sqrt{2k+1}} \sum_{j=-k}^{k} |j\rangle$ and $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$, respectively. It also initializes a register $\mathcal{T}$ to $|0\rangle$
- Apply the following operation to the $\mathcal{U}, \mathcal{J}, \mathcal{B}, \mathcal{T}$ registers:

$$|m, j, b, 0\rangle \mapsto |m, j, b, m - j - b\Gamma\rangle$$

- Now measure the $\mathcal{T}$ register to obtain a serial number T. Discard the $\mathcal{T}$ register.
- Now, given $T = m - j - b\Gamma$, and $\mathbf{x}$, this uniquely determines m, j, b: since $\mathbf{x} - m\mathbf{w}$ must be a short vector, there is only a single value of m for any $\mathbf{x}$. As explained in [KLS22], $m\mathbf{w}$ and hence m can be computed from $\mathbf{x}$ alone in the given parameter settings. Moreover, since $\Gamma \approx P/2 \gg k$, there is at most a single (j, b) for any m, T with $T = m - j - b\Gamma$. Moreover, (j, b) can be computed efficiently by simply trying both $b = 0$ and $b = 1$, and seeing which one gives $j = T + b\Gamma \in [-k, k]$. So we use this to uncompute $\mathcal{U}, \mathcal{J}, \mathcal{B}$ registers, which we then discard.

After obtaining T, the banknote is whatever state $|\phi_T\rangle$ is left in the $\mathcal{X}$ register, and the serial number is T.

Verification. We don't replicate the verification procedure here. Ultimately, we will show that verification is irrelevant: there is a procedure which produces two faked quantum money states, but which nevertheless fools the verifier.

Remark 1. In [KLS22], the banknote actually consists of a tuple of states as above. The overall verification will verify each element of the tuple separately, and then accept if a certain threshold of the elements accept. This structure is needed because their verification algorithm will reject honest banknotes some of the time. As we will see, this fact will not affect our attack, since our forged money will pass the verification of individual elements with the same probability as an honest banknote (up to potentially negligible error). For this reason, we just focus on a single element as described above.

6.2 An Alternate View of [KLS22]

Here, we present an alternative view of [KLS22] that we believe is easier to reason about.

First, we notice that the requirement that $\mathbf{v} \cdot \mathbf{v} \neq 0 \bmod P$ implies that the linear spaces $\mathcal{L}$ and $\mathcal{L}^\perp$ together span all of $\mathbb{Z}_P^d$. This means there is a one-to-one correspondence between $\mathbb{Z}_P^d$ and $\mathcal{L} \times \mathcal{L}^\perp$. Using that $\mathcal{L}^\perp$ is just the linear space of multiples of $\mathbf{v}$ and that $\mathbf{w} \cdot \mathbf{v} = 1$ by the definition of $\mathbf{w}$, we also have the correspondence:

$$\begin{gathered} \mathbf{y} \in \mathbb{Z}_P^d \leftrightarrow (\mathbf{x}, m) \in \mathcal{L} \times \mathbb{Z}_P \\ \mathbf{y} = \mathbf{x} - m\mathbf{w} \leftrightarrow \begin{array}{rcl} m & = & -\mathbf{v} \cdot \mathbf{y} \\ \mathbf{x} & = & \mathbf{y} + m\mathbf{w} = \mathbf{y} - (\mathbf{v} \cdot \mathbf{y})\mathbf{w} \end{array} \end{gathered} \tag{2}$$

Using this correspondence, the state in Equation 1 can be equivalently written as a state over register $\mathcal{Y}$ containing a Gaussian-weighted superposition over all integers:

$$|\phi_0\rangle = \frac{1}{C} \sum_{\mathbf{y} \in \mathbb{Z}_P^d} e^{-|\mathbf{y}|^2/4\sigma^2} |\mathbf{y}\rangle$$

Going between $|\phi\rangle$ and $|\phi_0\rangle$ is just a simple matter of linear algebra to go between $\mathbf{y}$ and $\mathbf{x}, m$.

Remark 2. Note that our alternative view gives a much simpler way to construct the state in Equation 1 than what was described in [KLS22]. Indeed, they construct the state by preparing first a large Guassian superposition of *lattice points* in $\mathcal{L}$. Then it performs a *lattice* quantum

Fourier transform, which gives a superposition over lattice points very close to dual lattice vectors. Finally, the lattice points in the superposition are close enough to the dual vectors that bounded distance decoding algorithms can be run to recover the dual lattice vectors, which gives the state in Equation 1. Each of these steps requires non-trivial algorithms that must all be performed in superposition. In contrast, our alternative view shows that one could instead simply create a Guassian-weighted superposition of *integers* and then perform some basic linear algebra.

The Measurement. Now consider the POVM described above. The action of the POVM on $|\phi\rangle$ translates straightforwardly to acting on $|\phi_0\rangle$, where instead of acting on the $\mathcal{U}$ register containing m directly, we instead compute m from $\mathbf{y}$ as $m = -\mathbf{v}\cdot\mathbf{y}$, and apply the POVM to this m. The $|\phi_T\rangle$ of the original minting process can therefore be obtained from $|\phi_0\rangle$ as follows:

- Initialize registers $\mathcal{J}$ and $\mathcal{B}$ to $\frac{1}{\sqrt{2k+1}}\sum_{j=-k}^{k}|j\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, respectively. It also initializes a register $\mathcal{T}$ to $|0\rangle$
- Apply the following operation to the $\mathcal{Y},\mathcal{J},\mathcal{B},\mathcal{T}$ registers:

$$|\mathbf{y},j,b,0\rangle \mapsto |\mathbf{y},j,b,-\mathbf{v}\cdot\mathbf{y}-j-b\Gamma\rangle$$

- Now measure the $\mathcal{T}$ register to obtain serial number T. Discard the $\mathcal{T}$ register. Call the state remaining in the $\mathcal{Y},\mathcal{J},\mathcal{B}$ registers $|\phi'_T\rangle$.
- Given $\mathbf{y}, T = -\mathbf{v}\cdot\mathbf{y}-j-b\Gamma$, we can compute $m,\mathbf{x}$ from $\mathbf{y}$ via the correspondence in Equation 2. From $m,\mathbf{x}$ we can un-compute $\mathbf{y}$, again using Equation 2. Then using $\mathbf{x}, T$, we can uniquely determine (m,j,b). So we use this to uncompute $\mathcal{U},\mathcal{J},\mathcal{B}$ registers, which we then discard.

Our equivalent formulation. Instead of arriving at $|\phi_T\rangle$ using our alternate formulation, we simply stop at the second-to-last step, outputting $|\phi'_T\rangle$. Since the final step of the minting process is reversible, it is equivalent to give out $|\phi_T\rangle$ and $|\phi'_T\rangle$.

6.3 Summary of Alternate Minting Process

We can now concisely describe out alternate minting process. This process will be equivalent to the original one, in that, we can map banknotes from our process to banknotes from [KLS22] and vice versa. This means that our alternate minting process is secure if and only if [KLS22] is secure.

Setup. Let P,σ,Δ,t,k as before. Let $\mathbf{v}$ as before, and define $\mathbf{v}' = (-\Gamma,-1,\mathbf{v}) \in \mathbb{Z}_P^{d'}$ where $d' = d+2$.

Minting. Let $\mathcal{Y}' = \mathcal{B}\times\mathcal{J}\times\mathcal{Y}$. Initialize the following state over register $\mathcal{Y}'$:

$$|\phi'\rangle \propto \sum_{\mathbf{y}\in\mathbb{Z}_P^d, j\in[-k,k], b\in\{0,1\}} e^{-|\mathbf{y}|^2/4\sigma^2}|b,j,\mathbf{y}\rangle$$

Note that this is a superposition over short vectors $\mathbf{y}' \in \mathbb{Z}_P^{d'}$. It is also a product state, with each coordinate being produced separately.

Now we apply the following map in superposition:

$$|\mathbf{y}'\rangle \mapsto |\mathbf{y}', \mathbf{v}' \cdot \mathbf{y}'\rangle$$

Now measure the $\mathcal{T}$ register (the second register), obtaining T. The result of the $\mathcal{Y}'$ register is exactly the state $|\phi'_T\rangle$, a uniform superposition over short vectors $\mathbf{y}'$ such that $\mathbf{v}' \cdot \mathbf{y}' = T$. Here, by short, we mean that each entry of $\mathbf{y}'$ is in $[-W, W]$, except with negligible probability, where $W = \max(2, k, \sigma \times \omega(\sqrt{\log(\lambda)}))$.

6.4 The Flaw

Here, we describe the flaw in [KLS22]. The flaw is most easily seen using our alternate view of their scheme.

Note that the vector $\mathbf{v}'$ has 3 known orthogonal short vectors, namely

$$\mathbf{s}_0 = \begin{pmatrix} 0 \\ 0 \\ \Delta \\ -2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{s}_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{s}_2 = \begin{pmatrix} -2 \\ 2\Gamma = 2t\sigma\Delta + 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

In [KLS22], the authors define three vectors $s^{(0)}, s^{(1)}, s^{(2)}$; $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2$ play the roles of $s^{(0)}, s^{(1)}, s^{(2)}$ in the alternate view, respectively.

In the security proof (originally starting from Page 4 of [KLS22], here translated to our alternate view), the authors imagine an adversary an adversary that outputs two possibly entangled states that pass verification. The authors argue that each state $|\phi\rangle$ must have significant weight on $\mathbf{y}'$ where the first digit b is 0, and also significant weight where b is 1. Recall that a valid money state must have the first digit being 0 or 1. If true, this would allow them to finish the proof: by measuring both states, the authors can then conclude that with non-negligible probability the resulting vectors $\mathbf{y}'_0, \mathbf{y}'_1$ will have different first bits. Then the difference between the vectors $\mathbf{y}'_0 - \mathbf{y}'_1$ is a short vector orthogonal to $\mathbf{v}'$. It moreover cannot be in the span of $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2$ because if if did, the coefficient of $\mathbf{s}_2$ must be $1/2$, and since the second coordinate of $\mathbf{s}_2$ is odd, the resulting vector would have a large second coordinate. Thus, the authors obtain a short vector linearly independent of the provided vectors, which is presumably hard.

The flaw in their argument is the claim that each state the adversary constructs must have significant weight on $b = 0$ and also $b = 1$. Indeed, the authors toward contradiction consider an adversary which measures the provided state to obtain $\mathbf{y}'$, and then constructs a state of the form:

$$\sum_j \alpha_j |\mathbf{y}' + j\mathbf{s}_1\rangle$$

for small j. They argue that their verification procedure would reject such a state. Indeed, their test essentially looks at the 4th coordinate in the Fourier domain (which is the 2nd coordinate in their view). Since the 4th coordinate in the primal is not affected by adding multiples of $\mathbf{s}_1$, the 4th coordinate is exactly the 4th coordinate of $\mathbf{y}'$. As such, in the Fourier domain the 4th coordinate

is uniform. On the other hand, the authors show that the 4th coordinate of their money state is non-uniform, which is leveraged in verification.

However, the authors failed to consider more general states of the form

$$\sum_{j_0,j_1} \alpha_{j_0,j_1} |\mathbf{y}' + j_0\mathbf{s}_0 + j_1\mathbf{s}_1\rangle$$

for small j_0, j_1. Since $\mathbf{s}_0$ is non-zero in the 4th coordinate, by adding multiples of $\mathbf{s}_0$, the Fourier transform of the 4th coordinate is no longer guaranteed to be uniform.

The authors reject such states as being a possibility, since the 3rd coordinate of $\mathbf{s}_0$ is Δ, which is larger than the allowed width of the 3rd coordinate of $\mathbf{y}'$ (which is limited to $\sigma \ll \Delta$). Thus, adding $\mathbf{s}_0$ to a vector in the allowed support of $\mathbf{y}'$ would move do outside the allowed support. However, the vector $\mathbf{s}_0 - \Delta \times \mathbf{s}_1$ has a small non-zero 4rd coordinate, and the 3rd coordinate is 0. Now, the 2nd coordinate of $\mathbf{s}_0 - \Delta \times \mathbf{s}_1$ is of size Δ, but the 2nd coordinate of $\mathbf{y}'$ is allowed to be as large as $k \gg \Delta$. So while one cannot add $\mathbf{s}_0$ itself to a state and remain in the required domain, one can add multiples of $\mathbf{s}_0 - \Delta \times \mathbf{s}_1$. In other words, even though adding $\mathbf{s}_0$ results in an invalid $\mathbf{y}'$, we can add an appropriate multiple of $\mathbf{s}_1$ to move back to a valid $\mathbf{y}'$. The introduction of $\mathbf{s}_0$ completely invalidates their security proof.

7 Our General Attack on a Class of Quantum Money

Now, we show that a natural class of schemes, including the equivalent view on [KLS22] demonstrated in Section 6.3, cannot possibly give secure quantum money schemes, regardless of how the verifier works.

7.1 The General Scheme

Here, we describe a general scheme which captures the alternate view above. Here, we use somewhat more standard notation from the lattice literature. Here we give a table describing how the symbols from section 6 map to this section:

This Section	Section 6.3
q	P
n	1
m	$d' = d + 2$
$\mathbf{A}$	$\mathbf{v}'$ as a column vector
$\|\psi\rangle$	$\|\phi'\rangle$
$\mathbf{u}$	T
W	$k + \sqrt{m} \times \sigma \times \omega(\sqrt{\log(\lambda)}))$

Setup. Let q be a super-polynomial, which may or may not be prime. Sample from some distribution several short vectors $\mathbf{s}_1, \ldots, \mathbf{s}_\ell \in \mathbb{Z}_q^m$ for a constant ℓ, and assemble them as a matrix $\mathbf{S} \in \mathbb{Z}_q^{m \times \ell}$. Then generate a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ such that $\mathbf{A}^T \cdot \mathbf{S} = 0 \bmod q$.

Minting. Create some superposition $|\psi\rangle$ of vectors in $\mathbf{y} \in \mathbb{Z}_q^m$ such that an all but negligible fraction of the support of $|\psi\rangle$ are on vectors with norm W. Let $\alpha_{\mathbf{y}}$ be the amplitude of $\mathbf{y}$ in $|\psi\rangle$.

Then apply the following map to $|\psi\rangle$:

$$|\mathbf{y}\rangle \to |\mathbf{y}, \mathbf{A}^T \cdot \mathbf{y} \bmod q\rangle$$

Finally, measure the second register to obtain $\mathbf{u} \in \mathbb{Z}_q^n$. This is the serial number, and the note is $|\psi_{\mathbf{u}}\rangle$, whatever remains of the first register, which is a superposition over short vectors $\mathbf{y}$ such that $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{u}$.

Verification. We do not specify verification. Indeed, in the following we will show that the money scheme is insecure, for *any* efficient verification scheme.

7.2 Attacking the General Scheme

We now show how to attack the general scheme. Let $\mathbf{C}$ be a matrix whose columns span the space orthogonal to the columns of $\mathcal{S}$. Let $|\psi'_{\mathbf{u}}\rangle$ be the state sampled from $|\psi_{\mathbf{u}}\rangle$ by measuring $\mathbf{y} \mapsto \mathbf{C}^T \cdot \mathbf{y}$, and letting $|\psi'_{\mathbf{u}}\rangle$ be whatever is left over.

Our attack will consist of two parts:

- Showing that $|\psi'_{\mathbf{u}}\rangle$ is indistinguishable from $|\psi_{\mathbf{u}}\rangle$, for *any* efficient verification procedure. We show (Section 7.3) that this follows from a certain "k-LWE" assumption, which depends on the parameters of the scheme (SS, k, n, etc). In Section B, we justify the assumption in certain general cases, based on the assumed hardness of worst-case lattice problems. Note that these lattice problems are essentially (up to small differences in parameters) the same assumptions we would expect are needed to show security for the money scheme in the first place. As such, if k-LWE does not hold for these special cases, most likely the quantum money scheme is insecure anyway. Our cases include the case of [KLS22].
- Showing that $|\psi'_{\mathbf{u}}\rangle$ can be cloned. Our attack first measures $|\psi'_{\mathbf{u}}\rangle$ to obtain a single vector $\mathbf{y}$ in it's support. To complete the attack, it remains to construct $|\psi'_{\mathbf{u}}\rangle$ from $\mathbf{y}$; by repeating such a process many times on the same $\mathbf{y}$, we successfully clone. We show (Section 7.4) that in certain general cases how to perform such a construction. Our cases include the case of [KLS22].

Taken together, our attack shows that not only is [KLS22] insecure, but that it quite unlikely that any tweak to the scheme will fix it.

7.3 Indistinguishability of $|\psi'_{\mathbf{u}}\rangle$

Here, we show that our fake quantum money state $|\psi'_{\mathbf{u}}\rangle$ passes verification, despite being a very different state that $|\psi_{\mathbf{u}}\rangle$. We claim that, from the perspective of any efficient verification algorithm, $|\psi'_{\mathbf{u}}\rangle$ and $|\psi_{\mathbf{u}}\rangle$ are indistinguishable. This would mean our attack succeeds.

Toward this end, let $\mathbf{C} \in \mathbb{Z}_q^{m \times (m-\ell)}$ be a matrix whose rows span the space orthogonal to $\mathbf{S}$: $\mathbf{C}^T \cdot \mathbf{S} = 0$. Notice that the state $|\psi'_{\mathbf{u}}\rangle$ can be equivalently constructed by applying the partial measurement of $\mathbf{C}^T \cdot \mathbf{y}$ to $|\psi_{\mathbf{u}}\rangle$

Consider the following problem, which is closely related to "k-LWE"(4):

Problem 4. Let n, m, q, Σ be functions of the security parameter, and D a distribution over $\mathbf{S}$. The $(n, m, q, \Sigma, \ell, D)$-LWE problem is to efficiently distinguish the following two distributions:

$$(\mathbf{A}, \mathbf{A} \cdot \mathbf{r} + \mathbf{e}) \quad \text{and} \quad (\mathbf{A}, \mathbf{C} \cdot \mathbf{r}' + \mathbf{e}) \ ,$$

Where $\mathbf{r}$ is uniform in $\mathbb{Z}_q^n$, $\mathbf{r}'$ is uniform in $\mathbb{Z}_q^{m-\ell}$, and $\mathbf{e}$ is Gaussian of width Σ. We say the problem is *hard* if, for all polynomial time quantum algorithms, the distinguishing advantage is negligible.

In Section B, we explain that in many parameter settings, including importantly the setting of [KLS22], that the hardness of Problem 4 is true (assuming standard lattice assumptions).

With the hardness of Problem 4, we can show the following, which is a generalization of a result of [LZ19] that showed that the SIS hash function is collapsing for super-polynomial modulus:

Theorem 5. *Consider sampling $\mathbf{A}, \mathbf{S}$ as above, and consider any efficient algorithm that, given $\mathbf{A}, \mathbf{S}$, samples a $\mathbf{u}$ and a state $|\phi_{\mathbf{u}}\rangle$ with the guarantee that all the support of $|\phi_{\mathbf{u}}\rangle$ is on vectors $\mathbf{y}$ such that (1) $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{u} \bmod q$ and (2) $|\mathbf{y}|_2 \leq W$.*

Now suppose $|\phi_{\mathbf{u}}\rangle$ is sampled according to this process, and then either (A) $|\phi_{\mathbf{u}}\rangle$ is produced, or (B) $|\phi'_{\mathbf{u}}\rangle$ is produced, where $|\phi'_{\mathbf{u}}\rangle$ is the result of applying the partial measurement of $\mathbf{C}^T \cdot \mathbf{y}$ to the state $|\phi_{\mathbf{u}}\rangle$.

Suppose there exists Σ such that $q/W\Sigma = \omega(\sqrt{\log \lambda})$ such that $(n, m, q, \Sigma, \ell, D)$-LWE is hard. Then cases (A) and (B) are computationally indistinguishable.

Note that an interesting consequence of Theorem 5 in the case $\ell = 0$ is that it shows that the SIS hash function is collapsing for any modulus, under an appropriate (plain) LWE distribution. This improves upon [LZ19], who showed the same but only for super-polynomial modulus. We now give the proof of Theorem 5:

Proof. For an integer t, let $\lfloor \cdot \rceil_t$ denote the function that maps a point $x \in \mathbb{Z}_q$ to the $z \in \{0, \lfloor q/t \rfloor, \lfloor 2q/t \rfloor, \cdot, \lfloor (t-1)q/t \rfloor\}$ that minimizes $|z - x|$. Here, $|z - x|$ is the smallest a such that $z = x \pm a \bmod q$. In other words, $\lfloor \cdot \rceil_t$ is a course rounding function that rounds an $x \in \mathbb{Z}_q$ to one of t points that are evenly spread out in $\mathbb{Z}_q$.

Let ρ be a mixed quantum state, whose support is guaranteed to be on $\mathbf{y}$ such that (1) $\mathbf{A}^T \cdot \mathbf{y} = \mathbf{u} \bmod q$ and (2) $|\mathbf{y}|_2 \leq W$. For a quantum process M acting on ρ, let $M(\rho)$ be the mixed state produced by applying M_i to ρ. We will consider a few types of procedures applied to on quantum states.

M_0: Given $\mathbf{A}$, M_0 is just the partial measurement of $\mathbf{y} \mapsto \mathbf{C}^T \cdot \mathbf{y}$.

M_1^t: Given $\mathbf{A}$, to apply this measurement, first sample an LWE sample $\mathbf{b} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$. Then apply the measurement $\mathbf{y} \mapsto \lfloor \mathbf{b} \cdot \mathbf{y} \rceil_t$. Discard the measurement outcome, and output the remaining state.

Lemma 6. *For any constants t, d, $M_1^t(\rho)$ is statistically close to $\frac{1}{d} M_1^{t \times d}(\rho) + \left(1 - \frac{1}{d}\right) \rho$*

Note that Lemma 6 means that M_1^t can be realized by the mixture of two measurements: $M_1^{t \times d}$ with probability $1/t^2$, and the identity with probability $\left(1 - \frac{1}{d}\right)$. We now give the proof.

Proof. Consider the action of M_1^t on $|\mathbf{y}\rangle\langle\mathbf{y}'|$, for a constant t. First, an LWE sample $\mathbf{b} = \mathbf{A}\cdot\mathbf{r} + \mathbf{e}$ is chosen. Then conditioned on this sample, if $\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t = \lfloor \mathbf{b}\cdot\mathbf{y}'\rceil_t$, the output is $|\mathbf{y}\rangle\langle\mathbf{y}'|$. Otherwise the output is 0. Averaging over all $\mathbf{b}$, we have that

$$M_1^t(|\mathbf{y}\rangle\langle\mathbf{y}'|) = \Pr_{\mathbf{b}}[\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t = \lfloor \mathbf{b}\cdot\mathbf{y}'\rceil_t]$$

where the probability is over $\mathbf{b}$ sampled as $\mathbf{b} = \mathbf{A}\cdot\mathbf{r} + \mathbf{e}$. Recalling that $\mathbf{u} = \mathbf{A}^T\cdot\mathbf{y} = \mathbf{A}^T\cdot\mathbf{y}'$, we have that:

$$\mathbf{b}\cdot\mathbf{y} = \mathbf{r}\cdot\mathbf{u} + \mathbf{e}\cdot\mathbf{y}$$
$$\mathbf{b}\cdot\mathbf{y}' = \mathbf{r}\cdot\mathbf{u} + \mathbf{e}\cdot\mathbf{y}'$$

Now, by our choice of Σ, $|\mathbf{e}\cdot(\mathbf{y}-\mathbf{y}')| < q/t$ for any constant t, except with negligible probability. We will therefore assume this is the case, incurring only a negligible error.

Note that $\mathbf{z} := \mathbf{r}\cdot\mathbf{u}$ is uniform in $\mathbb{Z}_q$ and independent of $\mathbf{e}\cdot\mathbf{y}, \mathbf{e}\cdot\mathbf{y}'$. So measuring $\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t$ is identical to measuring the result of rounding $\mathbf{e}\cdot\mathbf{y}$, except that the rounding boundaries are rotated by a random $\mathbf{z}\in\mathbb{Z}_q$. Since the rounding boundaries are q/t apart, at most a single rounding boundary can be between $\mathbf{e}\cdot\mathbf{y}$ and $\mathbf{e}\cdot\mathbf{y}'$, where "between" means lying in the shorter of the two intervals (of length $|\mathbf{e}\cdot(\mathbf{y}-\mathbf{y}')|$) resulting by cutting the circle $\mathbb{Z}_q$ at the points $\mathbf{e}\cdot\mathbf{y}$ and $\mathbf{e}\cdot\mathbf{y}'$. $\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t = \lfloor \mathbf{b}\cdot\mathbf{y}'\rceil_t$ if and only if no rounding boundary is between them.

Since the cyclic shift $\mathbf{z}$ is uniform each rounding boundary is uniform. Since there are t rounding boundaries and no two of them can between $\mathbf{e}\cdot\mathbf{y}$ and $\mathbf{e}\cdot\mathbf{y}'$, we have that, conditioned on $\mathbf{e}$, the probability $\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t \neq \lfloor \mathbf{b}\cdot\mathbf{y}'\rceil_t$ is therefore $\frac{t}{q}|\mathbf{e}\cdot(\mathbf{y}-\mathbf{y}')|$. Averaging over all $\mathbf{e}$, we have that, up to negligible error:

$$M_1^t(|\mathbf{y}\rangle\langle\mathbf{y}'|) = \left(1 - \frac{t}{q}\mathop{\mathbb{E}}_{\mathbf{e}}[|\mathbf{e}\cdot(\mathbf{y}-\mathbf{y}')|]\right)|\mathbf{y}\rangle\langle\mathbf{y}'|$$

Notice then that $M_1^t(|\mathbf{y}\rangle\langle\mathbf{y}'|) = \frac{1}{d}M_1^{t\times d}(|\mathbf{y}\rangle\langle\mathbf{y}'|) + \left(1-\frac{1}{d}\right)|\mathbf{y}\rangle\langle\mathbf{y}'|$. By linearity, we therefore prove Lemma 6. □

Note that the proof of Lemma 6 also demonstrates that M_0 and M_1^t commute, since their action on density matrices is just component-wise multiplication by a fixed matrix.

M_2^t: Given $\mathbf{A}$, to apply this measurement, first sample an LWE sample $\mathbf{b} = \mathbf{C}\cdot\mathbf{r}' + \mathbf{e}$. Then apply the measurement $\mathbf{y}\mapsto\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t$. Let p_t be the probability that $\lfloor x\rceil_t = \lfloor y\rceil_t$ for uniformly random $x, y\in\mathbb{Z}_q$. Note that for any constant t, $p_t \leq t^{-1} + O(q^{-1})$.

Lemma 7. *For any constant t, $M_2^t(\rho)$ is statistically close to $M_0(M_1^t(\rho)) + p_t(\rho - M_0(\rho))$.*

Note that unlike Lemma 6, the expression in Lemma 7 does not correspond to a mixture of measurements applied to ρ. However, we will later see how to combine Lemma 7 with Lemma 6 to obtain such a mixture.

Proof. The proof proceeds similarly to Lemma 6. We consider the action of M_2^t on $|\mathbf{y}\rangle\langle\mathbf{y}'|$, and conclude that

$$M_2^t(|\mathbf{y}\rangle\langle\mathbf{y}'|) = \Pr_{\mathbf{b}}[\lfloor \mathbf{b}\cdot\mathbf{y}\rceil_t = \lfloor \mathbf{b}\cdot\mathbf{y}'\rceil_t]$$

where the probability is over $\mathbf{b} = \mathbf{C} \cdot \mathbf{r}' + \mathbf{e}$. But now we have that

$$\mathbf{b} \cdot \mathbf{y} = \mathbf{r}'^T \cdot \mathbf{C}^T \mathbf{y} + \mathbf{e} \cdot \mathbf{y}$$
$$\mathbf{b} \cdot \mathbf{y}' = \mathbf{r}'^T \cdot \mathbf{C}^T \mathbf{y} + \mathbf{e} \cdot \mathbf{y}'$$

We consider two cases:

- $\mathbf{C}^T \cdot \mathbf{y} = \mathbf{C}^T \cdot \mathbf{y}'$. This case is essentially identical to the proof of Lemma 6, and we conclude that $\Pr_{\mathbf{b}}[\lfloor \mathbf{b} \cdot \mathbf{y} \rceil_t = \lfloor \mathbf{b} \cdot \mathbf{y}' \rceil_t] = 1 - \frac{t}{q} \mathbb{E}_{\mathbf{e}}[|\mathbf{e} \cdot (\mathbf{y} - \mathbf{y}')|]$. Note that for such $\mathbf{y}, \mathbf{y}'$, we also have

  $$M_0(M_1^t(|\mathbf{y}\rangle\langle\mathbf{y}'|)) + p_t(|\mathbf{y}\rangle\langle\mathbf{y}'| - M_0(|\mathbf{y}\rangle\langle\mathbf{y}'|)) = M_1^t(M_0(|\mathbf{y}\rangle\langle\mathbf{y}'|)) + p_t \times 0 = 1 - \frac{t}{q} \mathop{\mathbb{E}}_{\mathbf{e}}[|\mathbf{e} \cdot (\mathbf{y} - \mathbf{y}')|] \ ,$$

  since M_0 is the identity on such $|\mathbf{y}\rangle\langle\mathbf{y}'|$. Thus, we have the desired equality for $\rho = |\mathbf{y}\rangle\langle\mathbf{y}'|$.

- $\mathbf{C}^T \cdot \mathbf{y} \neq \mathbf{C}_T \cdot \mathbf{y}'$. In this case, $\mathbf{b} \cdot \mathbf{y}$ and $\mathbf{b} \cdot \mathbf{y}'$ are independent and uniform over $\mathbb{Z}_p$. Therefore, $\Pr_{\mathbf{b}}[\lfloor \mathbf{b} \cdot \mathbf{y} \rceil_t = \lfloor \mathbf{b} \cdot \mathbf{y}' \rceil_t] = p_t$. Note that for such $\mathbf{y}, \mathbf{y}'$, we also have

  $$M_0(M_1^t(|\mathbf{y}\rangle\langle\mathbf{y}'|)) + p_t(|\mathbf{y}\rangle\langle\mathbf{y}'| - M_0(|\mathbf{y}\rangle\langle\mathbf{y}'|)) = 0 + p_t|\mathbf{y}\rangle\langle\mathbf{y}'| \ ,$$

  since $M_0(|\mathbf{y}\rangle\langle\mathbf{y}'|) = 0$ in this case.

Thus for each $|\mathbf{y}\rangle\langle\mathbf{y}'|$, we have the desired equality. By linearity, this thus extends to all ρ. □

Combining Lemmas 6 and 7, we obtain:

Corollary 8. *For any constants t, d, $M_2^t(\rho)$ is statistically close to $\frac{1}{d} M_0(M_1^{t \times d}(\rho)) + \left(1 - \frac{1}{d} - p_t\right) M_0(\rho) + p_t \rho$.*

For d such that $1 - \frac{1}{d} - p_t \geq 0$, this represents a mixture of measurements $M_0 \circ M_1^{t \times d}$, M_0, and the identity.

We are now ready to prove Theorem 5. Suppose there is an algorithm A that constructs a mixed state ρ, and then can distinguish ρ from $M_0(\rho)$ with (signed) advantage ϵ. Let d be a positive integer, to be chosen later. Let $\rho_0 = \rho$, and $\rho_i = M_1^{t \times d}(\rho_{i-1})$. Note that for any polynomial i, ρ_i can be efficiently constructed. Let $\epsilon_0 = \epsilon$, and ϵ_i be the (signed) distinguishing advantage of A when given ρ_i vs $M_0(\rho_i)$.

Let δ_i be the (signed) distinguishing advantage of A for $M_2^t(\rho_i)$ and $M_1^t(\rho_i)$. Write $g = 1 - \frac{1}{d} - p_t$. Invoking Lemma 6 and Corollary 8 with d, we have that

$$\delta_i = \frac{1}{d} \epsilon_{i+1} + g \epsilon_i$$

Now, we note that δ_i must be negligible, by the assumed hardness of $(n, m, q, \Sigma, \ell, D)$-LWE. Solving the recursion gives:

$$\epsilon_i (-dg)^{-i} = \epsilon - \frac{1}{d} \sum_{j=0}^{i-1} (-dg)^{-j} \delta_{j+1}$$

Next, assume d is chosen so that dg is a constant greater than 1. Define $T = \sum_{j=0}^{\lambda - 1} (dg)^{-j} = \frac{dg}{dg-1} - 2^{-O(\lambda)}$. Consider the adversary A' for $(n, m, q, \Sigma, \ell, D)$-LWE, which does the following:

- On input $\mathbf{A}, \mathbf{S}, \mathbf{b}$, where $\mathbf{b} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$ or $\mathbf{b} = \mathbf{C} \cdot \mathbf{r}' + \mathbf{e}$, it chooses $j \in [0, \lambda - 1]$ with probability $(dg)^{-j}/T$
- Then it constructs ρ according to A.
- Next, A' computes ρ_j by applying $M_1^{t \times d}$ to ρ for j times.
- Now A' applies the measurement $\mathbf{y} \mapsto \lfloor \mathbf{b} \cdot \mathbf{y} \rceil_t$ to ρ_j, obtaining ρ'_j.
- A' runs the distinguisher for A, obtaining a bit b
- A' outputs b if j is even, $1 - b$ if j is odd.

Note that if $\mathbf{b}$ is $\mathbf{A} \cdot \mathbf{r} + \mathbf{e}$, then $\rho' = M_1^t(\rho_i)$, and if $\mathbf{b}$ is $\mathbf{C} \cdot \mathbf{r}' + \mathbf{e}$, then $\rho' = M_2^t(\rho_i)$. Therefore, the distinguishing advantage of A' is:

$$\delta = \frac{1}{T} \sum_{j=0}^{\lambda-1} (-dg)^{-j} \delta_{j+1}$$

Thus, we have that

$$\epsilon_\lambda (-dg)^{-\lambda} = \epsilon - \frac{T}{d} \delta \ ,$$

Noting that ϵ_λ must trivially be in $[-1/2, 1/2]$, we have that:

$$|\delta| \geq \frac{d}{T} \left(|\epsilon| - \frac{1}{2} (dg)^{-\lambda} \right) \geq d \left(1 - \frac{1}{dg} \right) |\epsilon| - 2^{-O(\lambda)}$$

Thus, if A has non-negligible distinguishing advantage, so does A', breaking the $(n, m, q, \Sigma, \ell, D)$-LWE assumption. This completes the proof of Theorem 5. □

7.4 Constructing $|\psi'_{\mathbf{u}}\rangle$

Here, we explain how to construct $|\psi'_{\mathbf{u}}\rangle$, given just the vector $\mathbf{y}$ that resulted from measuring it. We first observe that, since $|\psi'_{\mathbf{u}}\rangle$ has support only on vectors that differ from $\mathbf{y}$ by multiples of the columns of $\mathbf{S}$, we can write:

$$|\psi'_{\mathbf{u}}\rangle \propto \sum_{\mathbf{t}} \alpha_{\mathbf{y} + \mathbf{S} \cdot \mathbf{t}} |\mathbf{y} + \mathbf{S} \cdot \mathbf{t}\rangle$$

Where $\alpha_{\mathbf{y}}$ is the amplitude of $\mathbf{y}$ in $|\psi\rangle$. This gives a hint as to how to construct $|\psi'_{\mathbf{u}}\rangle$: create a superposition over short linear combinations of $\mathbf{S}$, and then use linear algebra to transition to a superposition over $\mathbf{y} + \mathbf{S} \cdot \mathbf{t}$, weighted according to α. The problem of course is that α may be arbitrary except for having support only on short vectors. Therefore, we do not expect to be able to construct $|\psi'_{\mathbf{u}}\rangle$ in full generality, and instead focus on special (but natural) cases, which suffice for our use.

Wide Gaussian Distributed. Suppose the initial state $|\psi\rangle$ is the discrete Gaussian over the integers: $|\psi\rangle = |\mathbf{\Psi}_{\mathbb{Z}^m, \Sigma, \mathbf{c}}\rangle$ for some center $\mathbf{c}$ and covariance matrix Σ. Then $|\psi'_{\mathbf{u}}\rangle$ is simply

$$|\mathbf{\Psi}_{\mathcal{L} + \mathbf{y}, \Sigma, \mathbf{c}}\rangle$$

Here, $\mathcal{L}$ is the integer lattice generated by the columns of $\mathbf{S}$, and $\mathcal{L} + \mathbf{y}$ is the lattice $\mathcal{L}$ shifted by $\mathbf{y}$. We can construct the state $|\mathbf{\Psi}_{\mathcal{L} + \mathbf{y}, \Sigma, \mathbf{c}}\rangle$ by first constructing $|\mathbf{\Psi}_{\mathcal{L}, \Sigma, \mathbf{c} - \mathbf{y}}\rangle$, and then adding $\mathbf{y}$ to the superposition. Thus, as long as $\mathbf{s}_i^T \cdot \Sigma^{-1} \cdot \mathbf{s}_i \leq 1/\omega(\sqrt{\log \lambda})$ for all i, we can construct the necessary state.

Constant Dimension, Hyper-ellipsoid Bounded. Here, we restrict $\mathcal{L}$ to having a constant number of columns, but greatly generalize the distributions that can be handled.

A hyper-ellipsoid is specified by a positive definite matrix Σ, which defines the set $E_{\Sigma,\mathbf{c}} = \{\mathbf{y} : (\mathbf{y}-\mathbf{c})^T \cdot \mathbf{M} \cdot (\mathbf{y}-\mathbf{c}) \leq 1\}$.

Definition 7 (Good Hyper-ellipsoid)**.** A *good hyper-ellipsoid* for $|\psi\rangle$ is an $E_{\Sigma,\mathbf{c}}$ such that there exists a function $\eta(\lambda)$ and polynomials $p(\lambda), q(\lambda)$ such that, if $|\psi\rangle$ is measured to get a vector $\mathbf{y}$, then each of the following are true except with negligible probability:

- $\mathbf{y} \in E_{\Sigma,\mathbf{c}}$. In other words, $E_{\Sigma,\mathbf{c}}$ contains essentially all the mass of $|\psi\rangle$.
- $|\alpha_{\mathbf{x}}|^2 \leq \eta(\lambda)$. In other words, η is an approximate upper bound on $\alpha_{\mathbf{x}}$.
- If a random vector $\mathbf{x}$ is chosen from $E_{\Sigma,\mathbf{c}} \cap \{\mathbf{y} + \mathbf{S}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^\ell\}$, then with probability at least $1/p(\lambda)$, $|\alpha_{\mathbf{x}}|^2 \geq \eta/q(\lambda)$. In other words, $E_{\Sigma,\mathbf{c}}$ doesn't contain too many points with mass too much lower than η.

Taken together, a good hyper-ellipsoid is one that fits reasonably well around the $|\psi\rangle$. It must contain essentially all the support of $|\psi\rangle$, but can over-approximate it by a polynomial factor.

Lemma 9. *Suppose there is a good hyper-ellipsoid for $|\psi\rangle$, and that $\alpha_{\mathbf{y}}$ can be efficiently computed given any vector $\mathbf{y}$. Then there is a polynomial-time algorithm which constructs $|\psi'_{\mathbf{u}}\rangle$ from $\mathbf{y}$*

Proof. Let $E_{\Sigma,\mathbf{c}}$ be the good hyper-ellipsoid. Let $\mathcal{L}$ be the lattice generated by the columns of $\mathbf{S}$. By assumption, with overwhelming probability if we measure $|\psi\rangle$ to get $\mathbf{y}$, we have $\mathbf{y} \in E_{\Sigma,\mathbf{c}}$. Let $E_{\Sigma',\mathbf{c}'}$ be the ellipsoid that is the intersection of $E_{\Sigma,\mathbf{c}}$ and the affine space $\{\mathbf{y} + \mathbf{S}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{R}^\ell\}$.

Claim 10. *There is PPT algorithm which, given $\mathbf{S}, \Sigma'$, computes $\mathbf{T} = \{\mathbf{r}_1, \cdots, \mathbf{r}_{\ell'}\}$ such that:*

- $\mathbf{r}_i^T \cdot (\Sigma')^{-1} \cdot \mathbf{r}_i \leq 2$ *for all $i \in [\ell']$, and*
- $E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{T}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell'}\} = E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{S}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell}\}$.

Proof. Write $(\Sigma')^{-1}$ as $(\Sigma')^{-1} = \mathbf{U}^T \cdot \mathbf{U}$. Let $\mathbf{S}' = \{\mathbf{s}'_1 = \mathbf{U}\cdot\mathbf{s}_1, \ldots, \mathbf{s}'_\ell = \mathbf{U}\cdot\mathbf{s}_n\}$, and let $\mathcal{L}'$ be the lattice generated by $\mathbf{S}'$. Since ℓ is constant, we can find shortest vectors in $\mathcal{L}'$ in polynomial time. Therefore, compute $\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell$ such that $\mathbf{r}'_i$ is the shortest vector in $\mathcal{L}'$ that is linearly independent from $\{\mathbf{r}'_1, \ldots, \mathbf{r}'_{i-1}\}$. Then let ℓ' be such that $|\mathbf{r}'_{\ell'}|^2 \leq 2$, but $|\mathbf{r}'_{\ell'+1}|^2 > 2$, or $\ell' = \ell$ if no such ℓ' exists.

Finally, let $\mathbf{r}_i = \mathbf{U}^{-1}\cdot\mathbf{r}'_i$. Clearly, we have that $\mathbf{r}_i^T \cdot (\Sigma')^{-1} \cdot \mathbf{r}_i \leq 2$. It remains to show that $E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{T}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell'}\} = E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{S}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell}\}$. First, we notice that the lattice $\mathcal{L}(\mathbf{T})$ spanned by $\mathbf{T}$ is a sub-lattice of $\mathcal{L}(\mathbf{S})$ spanned by $\mathbf{S}$. So one containment is trivial. Now assume toward contradiction that there is a $\mathbf{x} \in E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{S}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell}\}$ that is not in $E_{\Sigma',\mathbf{c}'} \cap \{\mathbf{y} + \mathbf{T}\cdot\mathbf{t} : \mathbf{t} \in \mathbb{Z}^{\ell'}\}$. This means $\mathbf{x}-\mathbf{y}$ is in $\mathcal{L}(\mathbf{S})$. We also have that $(\mathbf{y}-\mathbf{c}')^T\cdot(\Sigma')^{-1}\cdot(\mathbf{y}-\mathbf{c}') \leq 1$ (since and $(\mathbf{x}-\mathbf{c}')^T \cdot (\Sigma')^{-1} \cdot (\mathbf{x}-\mathbf{c}') \leq 1$. By the triangle inequality, we have therefore that $(\mathbf{x}-\mathbf{y})^T \cdot (\Sigma')^{-1} \cdot (\mathbf{x}-\mathbf{y}) \leq 2$.

But then we have that $\mathbf{U}\cdot(\mathbf{x}-\mathbf{y})$ has norm at most 2, lies in $\mathcal{L}'$, and is linearly independent of $\{\mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell'}\}$. This contradicts that $\mathbf{r}'_{\ell'+1}$ (which has norm squared strictly greater than 2) is a shortest vector linearly independent of $\{\mathbf{r}'_1, \ldots, \mathbf{r}'_{\ell'}\}$. This completes the proof of the claim. □

We now return to proving Lemma 9. Let $\beta = \omega(\log\lambda)$. We construct $|\psi'_{\mathbf{u}}\rangle$ in three steps:

- We first construct a state negligibly close to $|\mathbf{\Psi}_{\mathcal{L}+\mathbf{y},\beta\Sigma',\mathbf{c}'}\rangle$, as we did in the Gaussian-distributed case above.
- We then construct the state $|E\rangle$, defined as the uniform superposition over the intersection of $\mathcal{L}+\mathbf{y}$ and $E_{\Sigma',\mathbf{c}'}$. $|E\rangle$ will be obtained from $|\mathbf{\Psi}_{\mathcal{L}+\mathbf{y},\beta\Sigma',\mathbf{c}'}\rangle$ via a measurement.
- Construct $|\psi'_{\mathbf{u}}\rangle$ from $|E\rangle$. This also will be obtained via a measurement.

We now describe the two measurements. We start from the second. Let η, p, q be the values guaranteed by the goodness of $E_{\Sigma,\mathbf{c}}$. Define $\eta_{\mathbf{x}} = 1/\eta$ if $|\alpha_{\mathbf{x}}|^2 \leq \eta$, and otherwise $\eta_{\mathbf{x}} = 1/|\alpha_{\mathbf{x}}|^2$. To obtain $|\psi'_{\mathbf{u}}\rangle$ from $|E\rangle$, we apply the following map in superposition and measure the second register:

$$|\mathbf{x}\rangle \mapsto |\mathbf{x}\rangle \left(\sqrt{\eta_{\mathbf{x}}}\alpha_{\mathbf{x}}|0\rangle + \sqrt{1-|\eta_{\mathbf{x}}\alpha_{\mathbf{x}}|^2}|1\rangle\right)$$

Suppose for the moment that $\eta_{\mathbf{x}} = 1/\eta$ for all $\mathbf{x}$. Then conditioned on the measurement outcome being 0, the resulting state is exactly $|\psi'_{\mathbf{u}}\rangle$. By the guarantee that $E_{\Sigma,\mathbf{c}}$ is good, we have that except with negligible probability over the choice of $\mathbf{y}$, all but a negligible fraction of the support of $|\psi'_{\mathbf{u}}\rangle$ satisfies $\eta_{\mathbf{x}} = 1/\eta$. Therefore, we will assume (with negligible error) this is the case. The probability the measurement is 0 (over the choice of $\mathbf{y}$ as well) is $\mathbb{E}_{\mathbf{x}\leftarrow E_{\Sigma',\mathbf{c}'}}[\alpha_{\mathbf{x}}^2/\eta]$, which, with probability at least $1/p$ over the choice of $\mathbf{y}$, is at least $1/q$. Thus, the overall probability of outputting 0 is inverse polynomial, and in this case we produce a state negligibly close to $|\psi'_{\mathbf{u}}\rangle$.

It remains to construct $|E\rangle$ from $|\mathbf{\Psi}_{\mathcal{L}+\mathbf{y},\beta\Sigma',\mathbf{c}'}\rangle$. This follows a very similar rejection-sampling argument. Let

$$\gamma_{\mathbf{x}} = \begin{cases} e^{-\pi/\beta} \times \sqrt{e^{\pi(\mathbf{x}-\mathbf{c}')^T\cdot(\beta\Sigma')^{-1}\cdot(\mathbf{x}-\mathbf{c}')}} & \text{if } (\mathbf{x}-\mathbf{c}')^T\cdot(\Sigma')^{-1}\cdot(\mathbf{x}-\mathbf{c}') \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that $0 \leq \gamma_{\mathbf{x}} \leq 1$. Now apply to $|\mathbf{\Psi}_{\mathcal{L}+\mathbf{y},\beta\Sigma',\mathbf{c}'}\rangle$ the map $|\mathbf{x}\rangle \mapsto |\mathbf{x}\rangle(\gamma_{\mathbf{x}}|0\rangle + \sqrt{1-\gamma_{\mathbf{x}}^2}|1\rangle)$, and measure the second coordinate. If the measurement outcome is 0, then the resulting state is exactly $|E\rangle$. For $\mathbf{x} \in E_{\Sigma',\mathbf{c}'}$, we have $\gamma_{\mathbf{x}} \geq e^{-\pi/\beta} \geq 1 - o(1)$. Therefore, the probability the measurement outputs 0 is at least $1-o(1)$ times the probability measuring $\mathbf{\Psi}_{\mathcal{L}+\mathbf{y},\beta\Sigma',\mathbf{c}'}$ produces an $\mathbf{x} \in E_{\Sigma',\mathbf{c}'}$. This latter probability is $O_\ell(\beta^{-\ell/2})$, where the constant hidden by the big O depends on ℓ. Since ℓ is constant and β is polynomial (in fact, sub-polynomial), the overall probability is polynomial. This completes the construction of $|\psi'_{\mathbf{u}}\rangle$ and the proof of Lemma 9. □

7.5 Applying to [KLS22]

We can then adapt the quantum money scheme in [KLS22] into the above general framework.

To apply to [KLS22], we apply the above with $n = 1$.

First, recall that the first few entries of $\mathbf{v}'$ are $-\Gamma = (2t\sigma\Delta+1)/2, -1, 1, \Delta/2$, and the rest are random. We then note that we can always arbitrarily re-scale the vector $\mathbf{v}'$, without affecting the scheme, since it only permutes the serial numbers but does not change the scheme. Then we see

that $\mathbf{v}'$ is a random vector, subject to being orthogonal to the following three vectors:

$$\mathbf{s}_0 = \begin{pmatrix} 0 \\ 0 \\ \Delta \\ -2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{s}_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \mathbf{s}_2 = \begin{pmatrix} -2 \\ 2\Gamma = 2t\sigma\Delta + 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

A randomly re-scaled $\mathbf{v}'$ is just a random vector orthogonal to each of these three vectors, which are short and linearly independent.

So our alternative view of [KLS22] is an instance of the above the general scheme with $n = 1$ and short vectors $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$. We also note that we can easily construct a good ellipsoid for their $|\psi\rangle$, which has axis lengths $4, 2k, \sigma\omega(\log\lambda), \cdots, \sigma\omega(\log\lambda)$. By Lemma 9, we can therefore construct as many copies as we would like of the state $|\psi'_{\mathbf{u}}\rangle$, which will fool verification under Conjecture 4. In the next section, we demonstrate that Conjecture 4 holds for their particular parameter settings. Thus their scheme is insecure.

Remark 3. Note that in our re-conceptualized version of [KLS22], the superpositions always have support on vectors whose the first entry equal to 0 or 1. As such, there is no non-zero small multiple of $\mathbf{s}_1$ such that adding this multiple results results in another vector whore first entry is 0 or 1. This means that, in our attack, the superposition $|\psi'_{\mathbf{u}}\rangle$ will actually only have support on shifts by multiples of $\mathbf{s}_0, \mathbf{s}_1$.

8 Invariant Money

From this section on, we discuss our positive results on quantum money/lightning.

We now describe our framework for instantiating quantum money using invariants, or more precisely what we call *walkable* invariants.

Let X, Y be sets, and $I : X \to Y$ an efficiently computable function from X to Y. I will be called the "invariant." We will additionally assume a collection of permutations $\sigma_i : X \to X$ indexed by $i \in [r]$ for some integer r, with the property that the permutations respect the invariant:

$$I(\ \sigma_i(x)\ ) = I(x), \forall i \in [r]$$

In other words, action by each σ_i preserves the value of the invariant. We require that σ_i is efficiently computable given i. r may be polynomial or may be exponential. To make the formalism below simpler, we will be implicitly assuming that there exists a perfect matching between the σ_i such that for any matched $\sigma_i, \sigma_{i'}$, we have $\sigma_{i'} = \sigma_i^{-1}$. Moreover, i' can be found given i. This can be relaxed somewhat to just requiring that σ_i^{-1} can be efficiently computed given i, but requires a slightly more complicated set of definitions.

Given a point x, the orbit of x, denoted $O_x \subseteq X$, is the set of all z such that there exists a non-negative integer k and $i_1, \ldots, i_k \in [r]$ such that $z = \sigma_{i_k}(\sigma_{i_{k-1}}(\cdots\sigma_{i_1}(x)\cdot))$. In other words, O_x is the set of all z "reachable" from x by applying some sequence of permutations. Note that $I(z) = y$

for any $z \in O_x$. We will therefore somewhat abuse notation, and define $I(O_x) = y$. We also let P_y be the set of pre-images of y: $P_y = \{x \in X : I(x) = y\}$.

We will additionally require a couple properties, which will be necessary for the quantum money scheme to compile:

- **Efficient Generation of Superpositions:** It is possible to construct the uniform superposition over X: $|X\rangle := \frac{1}{\sqrt{|X|}} \sum_{x \in X} |x\rangle$.
- **Mixing Walks:** For an orbit O, with a slight abuse of notation let $\sigma_{O,i}$ be the (possibly exponentially large) permutation matrix associated with the action by σ_i on O. Then let $M_O = \frac{1}{r} \sum_{i \in [r]} \sigma_{O,i}$ be the component-wise average of the matrices. Let $\lambda_1(O), \lambda_2(O)$ be the largest two eigenvalues by absolute value[15], counting multiplicities. Note that $\lambda_1(O) = 1$, with corresponding eigenvector the all-1's vector. We need that there is an inverse polynomial δ such that, for every orbit O, $\lambda_2(O) \leq 1 - \delta$. This is basically just a way of saying that a random walk on the orbit using the σ_i mixes in polynomial time.

We call such a structure above a walkable invariant.

8.1 Quantum Money from Walkable Invariants

We now describe the basic quantum money scheme.

Minting. To mint a note, first construct the uniform superposition $|X\rangle$ over X. Then apply the invariant I in superposition and measure, obtaining a string y, and the state collapsing to:

$$|P_y\rangle := \frac{1}{\sqrt{|P_y|}} \sum_{x \in P_y} |x\rangle$$

This is the quantum money state, with serial number y.

Verification. To verify a supposed quantum money state $|\phi\rangle$ with serial number y, we do the following.

- First check that the support of $|\phi\rangle$ is contained in P_y. This is done by simply applying the invariant I in superposition, and measuring if the output is y. If the check fails immediately reject.
- Then apply the projective measurement given by the projection $\sum_{O \subseteq P_y} |O\rangle\langle O|$, where O ranges over the orbits contained in P_y, and $|O\rangle := \frac{1}{\sqrt{|O|}} \sum_{x \in O} |x\rangle$. In other words, project onto states where, for each orbit, the weights of x in that orbit are all identical; weights between different orbits are allowed to be different.

  We cannot perform this measurement exactly, but we can perform it approximately using the fact that $\lambda_2(O) \leq 1 - \delta$. This is described in Section 8.2 below. Outside of Section 8.2, we will assume for simplicity that the measurement is provided exactly.

  If the projection rejects, reject the quantum money state. Otherwise accept.

[15] They are real-valued, since M_O is symmetric, owing to the fact that we assumed the σ_i are perfectly matched into pairs that are inverses of each other.

It is hopefully clear that honestly-generated money states pass verification. Certainly their support will be contained in P_y, and they apply equal weight to each element in an orbit (and in fact, equal weight across orbits).

8.2 Approximate Verification

Here, we explain how to approximately perform the verification projection $V = \sum_{O \subseteq P_y} |O\rangle\langle O|$, using the fact that $\lambda_2(0) \leq 1 - \delta$ for all O. The algorithm we provide is an abstraction of the verification procedure of [FGH+12], except that that work presented the algorithm without any analysis. We prove that the algorithm is statistically close to the projection V, provided the mixing condition $\lambda_2(0) \leq 1 - \delta$ is met.

Theorem 11. *Assume $\lambda_2(0) \leq 1 - \delta$ for all O, for some inverse-polynomial δ. Then there is a QPT algorithm $\tilde{V}$ such that, for any state $|\psi\rangle$, if we let $|\psi'\rangle$ be the un-normalized post-measurement state from applying $\tilde{V}$ to $|\psi\rangle$ in the case $\tilde{V}$ accepts, then $|\psi'\rangle$ is negligibly close to $V|\psi\rangle$.*

Proof. Let $r' = 2r$. Let $\mathcal{R}$ be a register containing a superposition over $1, \ldots, r'$, and $\mathcal{S}$ be the register containing the supposed quantum money state. Define the following:

- The unitary U acting on $\mathcal{R} \otimes \mathcal{S}$ defined as $U = \sum_{i=1}^{r} |i\rangle\langle i| \otimes \sigma_i + \sum_{i=r+1}^{r'} |i\rangle\langle i| \otimes \mathbf{I}$. Here, we use a slight abuse of notation, using σ_i to denote the unitary implementing the classical permutation σ_i. U can be computed efficiently, by our assumption that we can efficiently invert σ_i.
- The state $|\mathbf{1}\rangle = \frac{1}{\sqrt{r'}} \sum_{i=1}^{r'} |i\rangle$
- The projection $P = |\mathbf{1}\rangle\langle\mathbf{1}| \otimes \mathbf{I}$.

Our algorithm is the following:

- Initialize the register $\mathcal{R}$ to $|\mathbf{1}\rangle$.
- Repeat the following $t = \lambda/\delta$ times:
  - Apply the unitary U to $\mathcal{R} \otimes \mathcal{S}$.
  - Apply the measurement corresponding to projection P to $\mathcal{R} \otimes \mathcal{S}$. If the measurement outcome is reject, immediately abort and reject the state.
  - Apply U^{-1}.
- If all t trials above accepted, then accept and output the contends of the $\mathcal{S}$ register.

We now analyze the algorithm above. $M = \frac{1}{r} \sum_{i \in [r]} \sigma_i$. This matrix is symmetric, since we assumed the σ_i come in pairs that are inverses of each other. It is not necessarily positive. For example, consider X that can be divided into two subsets X_0, X_1 such that each σ_i maps X_b to X_{1-b}. Then the vector that places equal weight on all $x \in X$, but makes the values in X_0 positive and X_1 negative will have eigenvalue -1. In general, the eigenvalues of M must be in the real interval $[-1, 1]$. The eigenvectors with eigenvalue 1 are exactly $|O\rangle$ and the space spanned by them,

as O ranges over all possible orbits. By our mixing assumption, all other eigenvalues are at most $1-\delta$.

Suppose $|\psi_\ell\rangle$ is an eigenvector of M, with eigenvalue ℓ. We now explore how the algorithm above behaves on $|\psi_\ell\rangle$. Then we will extend our understanding to non-eigenvector states.

We have that

$$\begin{aligned} PU|\mathbf{1}\rangle|\psi_\ell\rangle &= \frac{1}{\sqrt{r'}}\left(|\mathbf{1}\rangle\langle\mathbf{1}|\otimes\mathbf{I}\right)\cdot\left(\sum_{i=1}^{r}|i\rangle\otimes\sigma_i|\psi_\ell\rangle+\sum_{i=r+1}^{r'}|i\rangle|\psi_\ell\rangle\right) \\ &= \frac{1}{r'}|\mathbf{1}\rangle\otimes\left(\sum_{i=1}^{r}\sigma_i|\psi_\ell\rangle+r|\psi_\ell\rangle\right) \\ &= |\mathbf{1}\rangle\otimes\left(\frac{1}{2}M|\psi_\ell\rangle+\frac{1}{2}|\psi_\ell\rangle\right)=\left(\frac{1+\ell}{2}\right)|\mathbf{1}\rangle|\psi_\ell\rangle \end{aligned}$$

Now consider a general state $|\psi\rangle$, which we write in an eigenbasis for M as $|\psi\rangle=\sum_\ell \alpha_\ell|\psi_\ell\rangle$. Then $PU|\mathbf{1}\rangle|\psi\rangle=\sum_\ell\alpha_\ell\left(\frac{1+\ell}{2}\right)|\psi_\ell\rangle$. After t trials, we have:

$$(PU)^t|\mathbf{1}\rangle|\psi\rangle=|\mathbf{1}\rangle\otimes\sum_\ell\alpha_\ell\left(\frac{1+\ell}{2}\right)^t|\psi_\ell\rangle$$

Let $|\psi'\rangle$ be the state after discarding $|\mathbf{1}\rangle$. Now

$$\begin{aligned} |\ V|\psi\rangle-|\psi'\rangle\ |^2 &= \left|\ |\psi_\ell\rangle-\sum_\ell\alpha_\ell\left(\frac{1+\ell}{2}\right)^t|\psi_\ell\rangle\ \right|^2 \\ &= \left|-\sum_{\ell\neq 1}\alpha_\ell\left(\frac{1+\ell}{2}\right)^t|\psi_\ell\rangle\ \right|^2 \\ &= \sum_{\ell\neq1}|\alpha_\ell|^2\left(\frac{1+\ell}{2}\right)^{2t} \\ &\leq \sum_{\ell\neq1}|\alpha_\ell|^2\left(1-\delta/2\right)^{2t} \\ &= \left(1-|\alpha_1|^2\right)\left(1-\delta/2\right)^{2t}\leq e^{-\lambda} \end{aligned}$$

This bound is negligible. This completes the proof of Theorem 11. □

8.3 Hardness Assumptions

To get an intuition for security, we define assumptions which, together, imply the quantum money (even quantum lightning) scheme is secure. We first introduce some notation. Let $p\in[r]^k$ for some k. Given a starting element $x\in X$, we will interpret p as a path from x, leading to an element $z=\sigma_{p_k}(\sigma_{p_{k-1}}(\cdots\sigma_{p_1}(x)\cdot))$. We will therefore call p a path from x to z. We will say that p is a path *between* x and z if p is a path from x to z or a path from z to x.

Hardness of Path-finding. This is an analog of discrete log, but for our setting. When thinking of X as elliptic curves and the σ as isogenies, path-finding is just the problem of computing isogenies between elliptic curves, which is presumably hard.

In our setting, the assumption says: it should be hard for any quantum algorithm, given points in the same orbit, to find a path between them.

Assumption 1. Consider an adversary A playing the following game:

- The adversary outputs an $x \in X$.
- The challenger then computes a random $z \in O_x$.
- The adversary wins if it can output a path p between x to z.

The *path-finding assumption* is that, for all quantum polynomial-time adversaries A, the probability A wins in the above game is negligible.

Remark 4. Note that technically the challenger in the path-finding game might be inefficient, since it is required to sample a random z and no such explicit procedure is provided by an invariant. However, the game can be made efficient using the statistical property that $\lambda_2(O_x) \leq 1 - \delta$: it can simply do a random walk on O_x to compute z. Such z will be statistically close to uniform.

Knowledge of Path. This is an analog of the "knowledge of exponent" assumption due to Damgård [Dam92]. The knowledge of exponent assumption (KEA) says that, given (g, g^a) for unknown a, it is hard to find (h, h^a) without *also* finding a b such that $h = g^b$. Formalizing KEA classically is a bit subtle, as an adversary can always construct (h, h^a) using some exponent b, but then simply forget it. Or it could encrypt (g, g^a) under an FHE scheme, choose an encryption of a random b, and then homomorphically compute (h, h^a) where $h = g^b$. Then it decrypts the result to get (h, h^a), without ever explicitly writing down b.

In both the cases above, it is nevertheless trivial to figure out what b is by looking at the algorithm. In the first example, the execution transcript will contain b before it is erased. In the FHE example, the secret key is needed to decrypt (h, h^a). Using the same secret key also allows decrypting b.

KEA says that it must be possible to find b *always*, for any algorithm. This is formalized by means of an extractor: given any algorithm A, there exists an extractor E that is given the same inputs as A—importantly, including any random coins of A—that can find the corresponding b whenever A outputs an (h, h^a).

We now explore what such a knowledge assumption looks like quantumly. Of course, quantumly the KEA assumption is not interesting since there is no hardness over groups. But we can nevertheless try to see how we might formalize it. The immediate problem is that quantum algorithms can be probabilistic without having explicit random coins as input. Instead, the quantum algorithm could create a superposition and measure it. This measurement is unpredictable, and un-repeatable. Trying to run the adversary from the same initial inputs will give different answers every time.

This says that a knowledge assumption in the quantum setting must be conditioned on the *output* the adversary produces, rather than the input. Note that in the classical setting, we cared about the output as well (since different outputs would have different b values), but we could connect the inputs to outputs by making the adversary deterministic by considering the random coins as input. In the quantum setting, this is no longer the case.

But if we are only looking at the final state of the algorithm, we run into a different problem. Namely, we are back to the setting where the adversary could have known an explicit path at one point, and then discarded the information. This is potentially even easier quantumly than classically: a quantum algorithm could measure the path in the Fourier basis, which would have the effect of erasing the path.

Our solution is natural: we consider only adversaries A that are unitary. To get the final output, we must measure the registers containing the output. However, these are the *only* measurements performed and there is no measurement on the adversaries internal state. The extractor E is then given the final state of A (as well as the output). E then must be able to find b given this state. Note that the restriction to unitary A is without loss of generality, as any A can be made reversible by Stinespring dilation (basically, instead of measuring, we XOR the registers into some newly created registers).

The classical analog is to restrict to *reversible* classical adversaries, and giving E the *final* internal state of A in addition to A's output. Again, restriction to reversible A is without loss of generality. This notion is actually equivalent to the usual classical KEA: given the output and internal state of A, reverse A to find the input, including the random coins. Then apply the traditional KEA extractor using these random coins.

We now adapt this idea to the path-finding setting, giving the following notion of "Knowledge of Path". We will define two variants, based on whether the invariant is invertible.

Assumption 2. Let A be a quantum polynomial time adversary A that is unitary (in the above sense where there are no measurements except the output registers). Let E be a quantum polynomial time extractor that is given A's final output as well as its final state. Let (x, z) be the A's output, and $p \in [r]^*$ be the output of E.

Let B be the event that (1) $I(x) = I(z)$, but (2) p is not a path between x and z. In other words, B is the event that A outputs two elements with the same invariant but E fails to find a path between them.

The *knowledge of path assumption* is that, for any quantum polynomial time unitary A, there exists a quantum polynomial time E such that $\Pr[B]$ is negligible.

Remark 5. Note that Assumption 2 implies that it is infeasible to find x, z such that x and z are in *different* orbits, but $I(x) = I(z)$. This is because in such case, there does not exist a path from x to z and therefore E must fail.

Invariant Inversion. Sometimes, the invariant I may be invertible. This is not required (or forbidden) for constructing quantum money, but it makes it likely that Assumption 2 is false, since inverting $I(x)$ would give an element z that most likely has no known path to x (and a path may not even exist). Therefore, we will need to explicitly model such an invertible invariant, and modify our assumptions appropriately. So we introduce a classical *randomized* algorithm $I^{-1} : Y \to X$, with the guarantee that $\Pr[I(I^{-1}(y)) = y] = 1$ for all y. We will typically consider the random coins of I^{-1} as an explicit input, writing $I^{-1}(y; t)$.

Since our adversary can find multiple elements with the same invariant by inverting, and presumably elements obtained by inverting have no known path, we need to model this in our extractor. We therefore allow the extractor E to do one of two things:

- It can find a path from x to z, or

- It can find random coins t, a path p, with the requirement that p connects either x or z to $I^{-1}(y;t)$, where $y = I(x) = I(z)$.

In other words, the assumptions requires that the only way to find x, z with the same invariant is to either know a path between them, or at least one of x, z was the result of using the inversion algorithm on y and then following some path. We now give the assumption.

Assumption 3. Let A be a quantum polynomial time adversary A that is unitary (in the above sense where there are no measurements except the output registers). Let E be a quantum polynomial time extractor that is given A's final output as well as its final state. Let (x, z) be the A's output, and $p \in [r]^*, t$ be the output of E. Let B be the event that (1) $I(x) = I(z)$, but (2) p is not a path between any two of $\{x, z, I^{-1}(y;t)\}$.

The *knowledge of path assumption for invertible invariants* is that, for any quantum polynomial time unitary A, there exists a quantum polynomial time E such that $\Pr[B]$ is negligible.

We will also require that I^{-1} is hard to invert. Namely, that, given x, it should be infeasible to come up with coins t such that $I^{-1}(I(x);t) = x$. This is required for our updated knowledge of path assumption to be meaningful.

Assumption 4. Consider an adversary A playing the following game:

- The adversary outputs an $x \in X$. Let $y = I(x)$.
- The challenger then computes a random $z \in O_x$.
- The adversary wins if it can output t and a path p between $I^{-1}(y;t)$ and z.

The *Inversion Inverting assumption* is that, for all quantum polynomial-time adversaries A, the probability A wins in the above game is negligible.

As with Path Finding Assumption (Assumption 1), the challenger in the Inversion Inverting assumption can be made efficient by choosing z as a random walk starting from x. Note that in the experiment, x is only used to specify an orbit O_x; the adversary's goal only depends on z.

8.4 Security Proof for Invariant Money

Theorem 12. *Assuming the Path-Finding assumption (Assumption 1) and the Knowledge of Path Assumption (Assumption 2), the scheme above is secure quantum lightning. If the invariant is invertible, then assuming the Path-Finding assumption (Assumption 1), the Knowledge of Path Assumption for Invertible Invariants (Assumption 3), and the Inversion Inverting assumption (Assumption 4), the scheme above is secure quantum lightning.*

Proof. We prove the case for invertible invariants, the case of non-invertible invariants been very similar and somewhat simpler.

Toward contradiction, let A be a quantum lightning adversary with non-negligible advantage ϵ. By running A for up to λ/ϵ times, stopping at the first success, we can gaurantee that A wins with advantage negligibly close to 1. For simplicity in the following proof, we will assume the success probability is actually 1, incurring only a negligible error.

We then assume without loss of generality that A is unitary, so that the output is a pure state $\sum_{x,z,s}\alpha_{x,z,s}|x,z,s\rangle$, where the first two registers are the supposed quantum money states, and the last register is auxiliary state left over by running A.

Since A passes verification with probability 1, we can instead write the output of A as:

$$\begin{aligned}|\psi\rangle &:= \sum_{s,O_1,O_2:I(O_1)=I(O_2)}\beta_{O_1,O_2,s}|O_1\rangle|O_2\rangle|s\rangle\\ &= \sum_{s,O_1,O_2:I(O_1)=I(O_2)}\frac{\beta_{O_1,O_2,s}}{\sqrt{|O_1||O_2|}}\sum_{x\in O_1,z\in O_2}|x,z,s\rangle\\ &= \sum_{s,x,z:I(x)=I(z)}\frac{\beta_{O_x,O_z,s}}{\sqrt{|O_x||O_z|}}|x,z,s\rangle\end{aligned}$$

where O_1, O_2 range over orbits with the same invariant.

Let E be the extractor guaranteed by applying Assumption 3 to A. Now consider measuring the registers containing x, z, leaving the auxiliary state as

$$|\psi_{O_x,O_z}\rangle \propto \sum_s \beta_{O_x,O_z,s}|s\rangle$$

Then since $I(x) = I(y)$, we have that $E(x, z, |\psi_{O_x,O_z}\rangle)$ outputs p, t such that, with overwhelming probability over x, z, p, t, p is a path between two of $\{x, z, I^{-1}(I(x); t)\}$. B_1 as the event p connects x, z, B_2 as the event p connects $x, I^{-1}(I(x); t)$, and B_3 as the event p connects $z, I^{-1}(I(x); t)$. Let q_1, q_2, q_3 be the probabilities of the events B_1, B_2, B_3. Then $q_1 + q_2 + q_3 \geq 1 - \mathsf{negl}$.

Now we notice that for any $x' \in O_x$ and and $z' \in O_z$, the probability of obtaining x', z' is identical to the probability of obtaining x, z, and the state $|\psi_{O_x,O_z}\rangle = |\psi_{O_{x'},O_{z'}}\rangle = |\psi_{O_1,O_2}\rangle$. In particular, we have that $E(x', z', |\psi_{O_x,O_z}\rangle)$ outputs a path p between $\{x', z', I^{-1}(I(x'); t)\}$ with non-negligible probability, for uniform $x' \in O_x, y' \in O(z)$. Moreover, the quantities q_1, q_2, q_3 are unchanged by using x', z'. We now use this to construct adversaries for path-finding (Assumption 1) and inversion inverting (Assumption 4).

Let A_P be the following path-finding adversary:

- Run A and measure x, z, obtaining the state $|\psi_{O_x,O_z}\rangle$.
- Send z to the challenger, obtaining a random $x' \in O_z$ in response.
- Run $E(x', z, |\psi_{O_x,O_z}\rangle)$ to get p, t. Output p.

Note that A_P simulates exactly A, E in the case where x, z are in the same orbit, which in particular is implied by event B_1. Therefore, A_P outputs a path between x', z with probability at least q_1. By the assumed hardness of path-finding (Assumption 1), q_1 must be negligible.

Let A_I be the following inversion-inverting adversary:

- Run A and measure x, z, obtaining the state $|\psi_{O_x,O_z}\rangle$.
- Send x to the challenger, obtaining a random $x' \in O_x$ in response.
- Run $E(x', z, |\psi_{O_x,O_z}\rangle)$ to get p, t. Output p, t.

Note that A_P simulates exactly A, E, and so p connects x' to $I^{-1}(I(x); t)$ with probability q_2. By the assumed hardness of inversion inverting (Assumption 4), q_2 must be negligible. By an identical argument exchanging the roles of x and z, we must also have q_3 is negligible. This contradicts $q_1 + q_2 + q_3 \geq 1 - \mathsf{negl}$. This completes the security proof. □

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, CCC '09, pages 229–242, Washington, DC, USA, 2009. IEEE Computer Society.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th Annual ACM Symposium on Theory of Computing*, pages 41–60, New York, NY, USA, May 19–22, 2012. ACM Press.

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

[AFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 411–439. Springer, 2020.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 255–268, 2020.

[AL21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 501–530, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[ALL+21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 526–555, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[AMRR11] Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 167–177. IEEE, 2011.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 474–483, 2014.

[BB87] Charles H. Bennett and Gilles Brassard. Quantum public key distribution reinvented. *SIGACT News*, 18(4):51–53, July 1987.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 2004.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 320–331, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

[BDG22] Andriyan Bilyk, Javad Doliskani, and Zhiyong Gong. Cryptanalysis of three quantum money schemes. Cryptology ePrint Archive, Paper 2022/624, 2022. https://eprint.iacr.org/2022/624.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. https://eprint.iacr.org/2020/1024.

[BDS16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures, 2016. https://arxiv.org/abs/1609.09047.

[BF11] Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 544–574, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs - (extended abstract). In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 344–360, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. Csi-fish: efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019.

[BKW17] Dan Boneh, Sam Kim, and David J Wu. Constrained keys for invertible pseudorandom functions. In *Theory of Cryptography Conference*, pages 237–263. Springer, 2017.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[BLW17] Dan Boneh, Kevin Lewi, and David J Wu. Constraining pseudorandom functions privately. In *IACR International Workshop on Public Key Cryptography*, pages 494–524. Springer, 2017.

[BS04] Reinier Bröker and Peter Stevenhagen. Elliptic curves with a given number of points. In Duncan Buell, editor, *Algorithmic Number Theory*, pages 117–131, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[BY91] Gilles Brassard and Moti Yung. One-way group actions. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology – CRYPTO'90*, volume 537 of *Lecture Notes in Computer Science*, pages 94–107, Santa Barbara, CA, USA, August 11–15, 1991. Springer, Heidelberg, Germany.

[CD22a] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh (preliminary version). *Cryptology ePrint Archive*, 2022.

[CD22b] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh (preliminary version). Cryptology ePrint Archive, Paper 2022/975, 2022. https://eprint.iacr.org/2022/975.

[CL14] Alexander Coward and Marc Lackenby. An upper bound on reidemeister moves. *American Journal of Mathematics*, 136(4):1023–1066, 2014.

[CLG08] Denis Charles, Kristin Lauter, and Eyal Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22:93–113, 12 2008.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 556–584, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.

[Col09] Roger Colbeck. Quantum and relativistic protocols for secure multi-party computation, 2009.

[Col19] Andrea Coladangelo. Smart contracts meet quantum cryptography, 2019.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.

[CPDDF+19] Marta Conde Pena, Raul Durán Díaz, Jean-Charles Faugère, Luis Hernández Encinas, and Ludovic Perret. Non-quantum cryptanalysis of the noisy version of aaronson–christiano's quantum money scheme. *IET Information Security*, 13(4):362–366, 2019.

[CS20] Andrea Coladangelo and Or Sattath. A quantum money solution to the blockchain scalability problem. *Quantum*, 4:297, 2020.

[CX22] Shujiao Cao and Rui Xue. The gap is sensitive to size of preimages: Collapsing property doesn't go beyond quantum collision-resistance for preimages bounded hash functions. *Cryptology ePrint Archive*, 2022. CRYPTO 2022.

[CY14] Matthew Coudron and Henry Yuen. Infinite randomness expansion with a constant number of devices. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 427–436, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany.

[DF17] Luca De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 12, 2017.

[DFM20] Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In *IACR International Conference on Public-Key Cryptography*, pages 187–212. Springer, 2020.

[DN21] Nico Döttling and Ryo Nishimaki. Universal proxy re-encryption. In *IACR International Conference on Public-Key Cryptography*, pages 512–542. Springer, 2021.

[Dyn03] Ivan Alekseyevich Dynnikov. Recognition algorithms in knot theory. *Russian Mathematical Surveys*, 58(6):1093, 2003.

[FGH+10] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, Daniel Nagaj, and Peter Shor. Quantum state restoration and single-copy tomography for ground states of hamiltonians. *Physical review letters*, 105(19):190503, 2010.

[FGH+12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter W. Shor. Quantum money from knots. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 276–289, Cambridge, MA, USA, January 8–10, 2012. Association for Computing Machinery.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[JMV09] David Jao, Stephen D. Miller, and Ramarathnam Venkatesan. Expander graphs based on grh with an application to elliptic curve cryptography. *Journal of Number Theory*, 129(6):1491–1504, 2009.

[JQSY19] Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: a candidate for post-quantum cryptography. In *Theory of Cryptography Conference*, pages 251–281. Springer, 2019.

[Kan18] Daniel M. Kane. Quantum money from modular forms, 2018. https://arxiv.org/abs/1809.05925.

[KLS22] Andrey Boris Khesin, Jonathan Z Lu, and Peter W Shor. Publicly verifiable quantum money from random lattices, 2022. https://arxiv.org/abs/2207.13135v2.

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. In *Theory of Cryptography Conference*, pages 31–61. Springer, 2021.

[KSS21] Daniel M. Kane, Shahed Sharif, and Alice Silverberg. Quantum money from quaternion algebras. Cryptology ePrint Archive, Report 2021/1294, 2021. https://eprint.iacr.org/2021/1294.

[Lac15] Marc Lackenby. A polynomial upper bound on reidemeister moves. *Annals of Mathematics*, pages 491–564, 2015.

[Lac16] Marc Lackenby. Elementary knot theory. *arXiv preprint arXiv:1604.03778*, 2016.

[Lac21] Marc Lackenby. An online attack against wiesner's quantum money, 2021. https://www.maths.ox.ac.uk/node/38304.

[LAF+10] Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Jonathan A. Kelner, Avinatan Hassidim, and Peter W. Shor. Breaking and making quantum money: Toward a new quantum cryptographic protocol. In Andrew Chi-Chih Yao, editor, *ICS 2010: 1st Innovations in Computer Science*, pages 20–31, Tsinghua University, Beijing, China, January 5–7, 2010. Tsinghua University Press.

[LPSS14] San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 315–334, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[Lut10] Andrew Lutomirski. An online attack against wiesner's quantum money, 2010. https://arxiv.org/abs/1010.0256.

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[MM22a] Luciano Maino and Chloe Martindale. An attack on sidh with arbitrary starting curve. *Cryptology ePrint Archive*, 2022.

[MM22b] Luciano Maino and Chloe Martindale. An attack on sidh with arbitrary starting curve. Cryptology ePrint Archive, Paper 2022/1026, 2022. https://eprint.iacr.org/2022/1026.

[MMP22] Marzio Mula, Nadir Murru, and Federico Pintore. Random sampling of supersingular elliptic curves. *Cryptology ePrint Archive*, 2022.

[MZ22] Hart Montgomery and Mark Zhandry. Full quantum equivalence of group action dlog and cdh, and more. Cryptology ePrint Archive, Paper 2022/1135, 2022. https://eprint.iacr.org/2022/1135.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[Rei27] Kurt Reidemeister. Elementare begründung der knotentheorie. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 5, pages 24–32. Springer, 1927.

[Rob21] Bhaskar Roberts. Security analysis of quantum lightning. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 562–567, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[Rob22] Damien Robert. Breaking sidh in polynomial time. *Cryptology ePrint Archive*, 2022.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. https://eprint.iacr.org/2006/145.

[RS19] Roy Radian and Sattath. Semi-quantum money. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, page 132–146, New York, NY, USA, 2019. Association for Computing Machinery.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux*, 7(1):219–254, 1995.

[Shm22] Omri Shmueli. Public-key quantum money with a classical bank. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 790–803, 2022.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 127–156, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 408–438, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[Zha22] Mark Zhandry. New constructions of collapsing hashes. Cryptology ePrint Archive, Paper 2022/678, 2022. CRYPTO 2022, https://eprint.iacr.org/2022/678.

A Additional Preliminaries

Worst-Case Lattice Problems and LWE We next define the GapSVP problem, which is the worst-case lattice problem upon which the hardness of LWE is based.

Definition 8. Let n be an integer and $\gamma = \gamma(n) \geq q$ a real number. The (n, γ)-GapSVP problem is the problem of deciding, given a basis $\mathbf{B}$ of an n-dimensional lattice Λ and a number d, whether or not $\lambda_1(\Lambda) \leq d$ or $\lambda_1(\Lambda) > \gamma d$.

We emphasize that GapSVP is a "promise problem" and that a (successful) adversary can output whatever it wants when $d < \lambda_1(\Lambda) \leq \gamma d$.

In his seminal work [Reg05], Regev showed that LWE was hard worst-case lattice problems for uniformly random choices of $\mathcal{D}_{\mathbf{A}}$ and $\mathcal{D}_{\mathbf{R}}$ and when Ψ was defined to be choosing each coordinate

as a (small) discrete Gaussian. To capture this, we cite a theorem from [BLP+13] which itself is derived from Theorem 3.1 of [Reg05] and Theorem 3.1 of [Pei09].

Theorem 13. ***(Theorem 2.16, [BLP+13])*** *Let n, m, and q be positive integers and let $\mathcal{D}^m_{\Psi_\sigma}$ be a distribution over $\mathbb{Z}^m$ where each entry is selected according to a discrete Gaussian distribution with noise rate parameter $\sigma > 2\sqrt{n}$. Then there exists a quantum reduction from worst-case $\left(n, \tilde{O}\left(nq/\sigma\right)\right) - GapSVP$ to $\left(n, m, q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{U}\left(Z_q^n\right), \mathcal{D}^m_{\Psi_\sigma}\right)$-LWE. In addition, if $q \geq 2^{n/2}$, then there is also a classical reduction between those problems.*

We note that our presentation of this theorem differs quite a bit from the presentation in [BLP+13] because they present LWE as a problem over the cycle (the additive group of reals modulo 1) for ease of exposition about the noise parameters, but it makes more sense when working in quantum money setting to present things over the integers.

B On the Hardness of k-LWE

In this section we prove a number of hardness results on k-LWE, showing that the k-LWE instance implied by the [KLS22] scheme is hard, assuming standard lattice assumptions. In addition, we provide additional evidence through proofs that more general instances of k-LWE are likely to be hard, which seemingly indicates that it would be difficult to "tweak" the [KLS22] construction by altering the distributions to gain security.

We start by presenting some useful LWE lemmas from previous work.

B.1 Helpful LWE Lemmas

In this section, we add in some useful lemmas that allow us to change the distribution of the key in the LWE problem ($\mathcal{D}_{\mathbf{r}}$) without affecting the hardness of the underlying problem too much. Looking ahead, we will need to use various versions of the powerful modulus switching lemma from [BLP+13], which requires LWE instances with "low-norm" keys.

We start with a simple folklore lemma, which informally states that LWE with a uniformly sampled random key is at least as hard as LWE with any other secret key distribution.

Lemma 14. *Let n, m, and q be integers, let $\mathcal{D}_{\mathbf{A}}$ and $\mathcal{D}_{\mathbf{r}}$ be distributions over $\mathbb{Z}_q^n$, and let $\mathcal{D}_{\Psi}$ be a distribution over $\mathbb{Z}_q^m$. Any adversary that can solve the $\left(n, m, q, \mathcal{D}_{\mathbf{A}}, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{D}_{\Psi}\right)$-LWE problem with advantage ϵ can be used to solve the $(n, m, q, \mathcal{D}_{\mathbf{A}}, \mathcal{D}_{\mathbf{r}}, \mathcal{D}_{\Psi})$-LWE problem with advantage ϵ.*

Proof. We give an abbreviated proof because this result is simple and well-known. Given an LWE challenge tuple $(\mathbf{A}, \mathbf{t})$ where $\mathbf{t} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$ and $\mathbf{r} \leftarrow \mathcal{D}_{\mathbf{r}}$ or $\mathbf{t}$ is random, sample $\mathbf{r}' \in \mathbb{Z}_q^n$ uniformly at random and add $\mathbf{A} \cdot \mathbf{r}'$ to $\mathbf{t}$. This gives the correct LWE challenge distribution if $\mathbf{t} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$ and is still uniformly random if $\mathbf{t}$ was uniformly random. □

We will also use a lemma from [BLP+13] that says, informally speaking, that LWE with certain parameters where the key is drawn from the noise distribution is at least as hard as when the key is uniform (modulo some small parameter losses). We state this below.

Lemma 15. ***(Lemma 2.12, [BLP+13])*** *Let n, m, and q be positive integers with $q \geq 25$. Let $m' = m - (16n + 4\ln\ln q)$. Consider some parameter $s \geq \sqrt{\ln\left(2n\left(1 + 1/\epsilon\right)/\pi\right)}$. Let $\epsilon < \frac{1}{2}$*

and $\sigma, \sigma' > 0$ be real numbers such that $\sigma' \geq \sqrt{\sigma^2 + s^2}$. Finally, let $\mathcal{D}_{\mathbf{\Psi}_\sigma}$ be a discrete Gaussian distribution with parameter σ.

Any adversary that can solve the $\left(n, m', q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{D}_{\mathbf{\Psi}_{\sigma'}}, \mathcal{D}_{\mathbf{\Psi}_\sigma}\right)$-LWE problem with advantage ϵ' can be used to solve the $\left(n, m, q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{D}_{\mathbf{\Psi}_\sigma}\right)$-LWE problem with advantage $\left(\epsilon' - 8\epsilon\right)/4$. In particular, assuming $\sigma > s$, we can take $s = \sigma$ and set $\sigma' = \sqrt{2}\sigma$.

Note that the two versions of LWE in this reduction have a (slightly) different number of samples (m and m'), but the dimension of the LWE problem (n) is the same.

B.2 Modulus Switching

In their seminal work [BLP+13], Brakerski *et al.* use a technique called *modulus switching* to improve known reductions from the GapSVP problem to the LWE problem. Informally speaking, this modulus switching technique allows those authors to show that LWE in "small modulus" and "high dimension" is roughly equivalent in hardness to LWE in "big modulus" and "low dimension". We state some of their results here since we would eventually like to use the fact that one-dimensional LWE with exponential modulus is hard, which is known from results in [BLP+13].

Rather than present a single instance of their main theorem, we go through two of their corollaries to make it easier to follow. We emphasize that our presentation looks very different from theirs because they consider LWE over the unit cycle $\mathbb{T}$ and we work over the integers.

Lemma 16. ***(Corollary 3.2, [BLP+13])*** *Let n, m, q, and q' be positive integers with $q' > q$, and consider some (B, δ)-bounded distribution $\tilde{\mathcal{D}}$ over $\mathbb{Z}^n$. Let $\epsilon \in \left(0, \frac{1}{2}\right)$ be a parameter and let $\sigma, \sigma > 0$ be real numbers. Finally, let*

$$\sigma' \geq \sqrt{\left(\sigma \frac{q'}{q}\right)^2 + (4/\pi) \ln\left(2n\left(1 + 1/\epsilon\right)\right) \cdot B^2}$$

Then there is an efficient reduction from $\left(n, m, q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \tilde{\mathcal{D}}, \mathcal{D}_{\mathbf{\Psi}_\sigma}\right)$-LWE to $\left(n, m, q', \mathcal{U}\left(\mathbb{Z}_q^n\right), \tilde{\mathcal{D}}, \mathcal{D}_{\mathbf{\Psi}_{\sigma'}}\right)$-LWE that reduces the advantage by at most $\sigma + 14\epsilon m$.

Informally speaking, this lemma gives us a way to increase the modulus of an LWE instance while keeping the "gap between the noise level and the modulus" almost the same. This is a rather counterintuitive result, and even moreso when you consider the fact that it works for essentially arbitrary distributions on LWE secrets. Importantly, q and q' can be (essentially) arbitrary as long as $q' > q$, so we can use this lemma to help us prove LWE hardness for "arbitrary" choices of q' in conjunction with other lemmas that require certain properties of q'.

We next present another modulus switching lemma that lets us go from "high modulus, low dimension" to "normal modulus, normal dimension" instances of LWE. Once again, note that this reduction approximately preserves the gap between the noise rate and the modulus.

Lemma 17. ***(Corollary 3.4, [BLP+13])*** *Consider any positive integers n, m, q, and k such that k divides n, real numbers $\sigma, \sigma' > 0$, a parameter $\epsilon \in \left(0, \frac{1}{2}\right)$, and some (B, δ)-bounded distribution $\tilde{\mathcal{D}}$. In addition, let*

$$\sigma' \geq \sqrt{\left(\sigma q^{k-1}\right)^2 + (4/\pi) \ln\left(2n\left(1 + 1/\epsilon\right)\right) \cdot \left(B q^{k-1}\right)^2}$$

and define $\mathbf{G} = \mathbf{I}_{n/k} \otimes \left(1, q, q^2, ..., q^{k-1}\right)^T$.

Then there is an efficient reduction from $\left(n, m, q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \tilde{\mathcal{D}}, \mathcal{D}_{\mathbf{\Psi}_\sigma}\right)$*-LWE to* $\left(n/k, m, q^k, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathbf{G}\tilde{\mathcal{D}}, \mathcal{D}_{\mathbf{\Psi}_{\sigma'}}\right)$*-LWE that reduces the advantage by at most* $\delta + 14\epsilon m$.

Note that setting $k = n$ gives us hardness for single-dimension LWE (i.e. $n = 1$). We will use this exact setting later in our proofs.

B.3 k-LWE for a Constant Number of Vectors Is Hard

In this section we prove that k-LWE with the appropriate parameters is as hard as regular LWE for *any* distribution on the short vectors, up to a superpolynomial loss in the noise, *assuming that the number of short vectors k is constant.* We state this below. Our proof is inspired by the k-SIS proof of [BF11].

Lemma 18. *Let k, n, m, and q be positive integers, and let $\mathcal{D}_{\mathbf{R}}$ be a distribution over $\mathbb{Z}_q^n$. Let $\mathcal{D}_{\mathbb{\Psi}_\sigma}$ be a discrete Gaussian distribution over $\mathbb{Z}_q^m$ with noise parameter (width) σ. Furthermore, let $\mathcal{D}_{\mathbf{S}}$ be a distributions over $\mathbb{Z}_q^m$ with the additional requirement that $\mathcal{D}_{\mathbf{S}}$ is B-bounded. Let $\mathbf{S} \in \mathbb{Z}_q^{k \times m}$ be a matrix where each* row *is selected from $\mathcal{D}_{\mathbf{S}}$. Let $f(n)$ be a function that is superpolynomial in n.*

Any adversary that can solve the $\left(k, n, m+k, q, \mathcal{D}_{\mathbf{S}}, \mathcal{D}_{\mathbf{r}}, \mathcal{D}_{\mathbb{\Psi}_{\sigma f(n)}}\right)$*-k-LWE problem with advantage ϵ can be used to solve the* $\left(n, m, q, \mathcal{U}\left(\mathbb{Z}_q^n\right), \mathcal{D}_{\mathbf{r}}, \mathcal{D}_{\mathbb{\Psi}_\sigma}\right)$*-LWE problem with advantage $\epsilon - \mathsf{negl}(n)$.*

Proof. Suppose we are given an LWE challenge tuple in the form of $(\mathbf{A}, \mathbf{t})$ where $\mathbf{t} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$ (the "real" case) or $\mathbf{t} = \mathbf{r}$ for some uniformly sampled $\mathbf{r}$. We will show that we can use this LWE challenge to build a k-LWE challenge of the appropriate distribution in a way that succeeds with all but negligible probability. Thus, given an appropriate k-LWE adversary, we can simply feed it our challenge tuple and then mimic that response in the LWE challenge game, winning with probability negligibly close to ϵ. Our reduction proceeds as follows (and borrows heavily from the techniques of [BF11]):

First, suppose we sample $\mathbf{S} \leftarrow \mathcal{D}_{\mathbf{S}} \in \mathbb{Z}_q^{k \times (m+k)}$ and let $\mathbf{S}_{i,j}$ denote the $(i,j)\,th$ entry of $\mathbf{S}$. Suppose we pick k columns of $\mathbf{S}$ such that the $k \times k$ submatrix formed by these columns is full-rank; and note that we can always do this unless $\mathbf{S}$ has rank less than k (in which case, we can just reduce $\mathbf{S}$ by one row and repeat the process). Call this matrix $\mathbf{T}$. Let $\tilde{\mathbf{T}}$ denote the *adjugate* matrix of $\mathbf{T}$; in other words, $\tilde{\mathbf{T}} \cdot \mathbf{T} = det(\mathbf{T}) \cdot \mathbf{I}_k$ and $\tilde{\mathbf{T}} \in \mathbb{Z}^{k \times k}$. Let $\tilde{\mathbf{S}} \in \mathbb{Z}_{k \times m} = \tilde{\mathbf{T}} \cdot \mathbf{S}$, and note that it has the following structure:

$$\tilde{\mathbf{S}} = \begin{bmatrix} det(\mathbf{T}) & 0 & 0 & ... & 0 & \tilde{\mathbf{S}}_{1,k+1} & ... & \tilde{\mathbf{S}}_{1,k+m} \\ 0 & det(\mathbf{T}) & 0 & ... & 0 & \tilde{\mathbf{S}}_{2,k+1} & ... & \tilde{\mathbf{S}}_{2,k+m} \\ 0 & 0 & det(\mathbf{T}) & ... & 0 & \tilde{\mathbf{S}}_{3,k+1} & ... & \tilde{\mathbf{S}}_{3,k+m} \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & det(\mathbf{T}) & \tilde{\mathbf{S}}_{k,k+1} & ... & \tilde{\mathbf{S}}_{k,k+m} \end{bmatrix}$$

Suppose we let $\mathbf{U} \in \mathbb{Z}^{(m+k) \times m}$ be defined in the following way: for each entry $\mathbf{U}_{i,j}$ where $i \leq k$, we set $\mathbf{U}_{i,j} = \tilde{\mathbf{S}}_{i,j+k}$. For each entry $\mathbf{U}_{i,j}$ for $i > k$, we set $\mathbf{U}_{i,j} = -det(\mathbf{T})$ if $i = j$ and 0 otherwise.

Note that, pictorially, this gives us the following structure:

$$\mathbf{U} = \begin{bmatrix} \tilde{\mathbf{S}}_{1,k+1} & \tilde{\mathbf{S}}_{1,k+2} & \tilde{\mathbf{S}}_{1,k+3} & \dots & \tilde{\mathbf{S}}_{1,k+m} \\ \tilde{\mathbf{S}}_{2,k+1} & \tilde{\mathbf{S}}_{2,k+2} & \tilde{\mathbf{S}}_{2,k+3} & \dots & \tilde{\mathbf{S}}_{2,k+m} \\ \tilde{\mathbf{S}}_{3,k+1} & \tilde{\mathbf{S}}_{3,k+2} & \tilde{\mathbf{S}}_{3,k+3} & \dots & \tilde{\mathbf{S}}_{3,k+m} \\ \dots & \dots & \dots & \dots & \dots \\ \tilde{\mathbf{S}}_{k,k+1} & \tilde{\mathbf{S}}_{k,k+2} & \tilde{\mathbf{S}}_{k,k+3} & \dots & \tilde{\mathbf{S}}_{k,k+m} \\ -det(\mathbf{T}) & 0 & 0 & \dots & 0 \\ 0 & -det(\mathbf{T}) & 0 & \dots & 0 \\ 0 & 0 & -det(\mathbf{T}) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -det(\mathbf{T}) \end{bmatrix}$$

We interrupt the reduction to make a couple of claims.

Claim: $\mathbf{S} \cdot \mathbf{U} = 0$. This follows from the fact that $\tilde{\mathbf{S}}$ is orthogonal to $\mathbf{U}$ by definition and $\mathbf{S} = \tilde{T}\mathbf{S}$. Since $\mathbf{T}$ is invertible (over both $\mathbb{Z}$ and $\mathbb{Z}_q$, $\mathbf{S} \cdot \mathbf{U} = 0$ if and only if $\tilde{\mathbf{S}} \cdot \mathbf{U} = 0$.

Claim: no entry in U has value larger than $(2B)^k$. First, note that the determinant of any $k \times k$ matrix with entries in $[-B, B]$ is at most $(2B)^k$, and the determinant of any B-bounded smaller matrices must be smaller than that. Since the entries of the adjugate matrices are themselves determinats of submatrices of $\mathbf{T}$, they must also follow this bound.

Now we may continue with our reduction. Suppose we sample $\mathbf{e}' \leftarrow \mathcal{D}^{m+k}_{\Psi_{\sigma f(n)}}$ and let $\mathbf{A}' \in \mathbb{Z}_q^{m+k} \times n = \mathbf{U} \cdot \mathbf{A}$ and $\mathbf{t}' = \mathbf{U} \cdot \mathbf{t} + \mathbf{e}'$. Additionally, suppose we sample some random matrix $\mathbf{W} \in \mathbb{Z}_q^{m \times m}$ and set $\mathbf{C} = \mathbf{U} \cdot \mathbf{W}$. We claim that the tuple $(\mathbf{S}, \mathbf{A}', \mathbf{C}, \mathbf{t}')$ is an appropriately distributed k-LWE challenge. We show this in a number of claims: first, with some claims that apply to both the "real" and "random" cases, and then we argue these cases separately.

Claim: C is a uniform basis of the kernel of S $\mod q$**.** This follows from the fact that $\mathbf{S} \cdot \mathbf{C} = 0$ over $\mathbb{Z}$ and the ranks of the matrices sum to $m + k$.

Claim: $\mathbf{A}' = \mathbf{U} \cdot \mathbf{A} \mod q$ **is distributed uniformly at random subject to the constraint** $\mathbf{S} \cdot \mathbf{A}' = 0$. Since $\mathbf{A}$ is uniformly random and $\mathbf{U}$ is a basis for all vectors in the kernel of $\mathbf{S}$, we know that $\mathbf{A}'$ is distributed appropriately $\mod q$.

So, at this point we know that $\mathbf{S}$, $\mathbf{A}'$, and $\mathbf{C}$ are distributed appropriately. All that remains is to handle $\mathbf{t}'$ We handle this separately for the "real" and "random" cases below.

The "Real" Case. Assume now that $\mathbf{t} = \mathbf{A} \cdot \mathbf{r} + \mathbf{e}$. Then we have

$$\mathbf{t}' = \mathbf{U}(\mathbf{A} \cdot \mathbf{r} + \mathbf{e}) + \mathbf{e}' = \mathbf{A}' \cdot \mathbf{r} + (\mathbf{U}\mathbf{e} + \mathbf{e}').$$

If k is constant, then $(2B)^k$ is constant and $\mathbf{U}\mathbf{e}$ has no entries larger than $O(m^2\sigma)$ with all but negligible probability since the probability that a discrete Gaussian with parameter σ is larger than $m\sigma$ is negligible. Since $f(n)$ grows faster than any polynomial, we know that $\mathbf{U} \cdot \mathbf{e} + \mathbf{e}'$ is distributed statistically close to just sampling a discrete Gaussian with parameter $\sigma f(n)$ (i.e. how we sampled $\mathbf{e}'$), so we know that $\mathbf{t}'$ is sampled appropriately in this case.

The "Random" Case. Now assume that $\mathbf{t}'$ is distributed uniformly at random over $\mathbb{Z}_q^m$. In this case, we know that $\mathbf{U} \cdot \mathbf{t}$ is distributed uniformly at random over the kernel of $\mathbf{S}$ $\mod q$ and thus

can be written as $\mathbf{C} \cdot \mathbf{r}'$ for a uniformly random $\mathbf{r}'$ as desired. Since $\mathbf{t}' = \mathbf{U} \cdot \mathbf{t} + \mathbf{e}'$, we therefore know that the output distribution is correct in the "random" case as well.

Completing these two cases finishes the reduction and completes the proof. □

B.4 Putting It All Together

In order to show that any adversary that can solve the k-LWE instance implied by the [KLS22] scheme can solve worst-case lattice problems, we just need to put all of our lemmas together. Below, we show a table containing all of our hybrid arguments with approximate factors: we ignore constant factors in the dimensions and polynomial factors in the noise and modulus (when the modulus is exponentially large). We assume $q > (q')^n$ and that the noise distributions are Gaussians. We note that the "Noise" entry for GapSVP is the approximation ratio defined by the problem, not the noise itself.

Assumption	Lattice Dim.	Samples	Modulus	Key Dist.	Noise	Proof Comment
$k-LWE$	1	m	q	Unif.	$\frac{q}{\eta}$	n/a
LWE	1	m	q	Unif.	$\frac{q}{\eta f(n)}$	Lemma B.4
LWE	1	m	q	Noise	$\frac{q}{\eta f(n)}$	Lemma 14
LWE	1	m	$(q')^n$	Noise	$\frac{(q')^n}{\eta f(n)}$	Lemma 16
LWE	n	m	q'	Noise	$\frac{q'}{\eta f(n)}$	Lemma 17
LWE	n	m	q'	Unif.	$\frac{q'}{\eta f(n)}$	Lemma 15
$GapSVP$	n	–	–	–	$\eta f(n)$	Theorem 13

We can state this in a nice lemma.

Lemma 19. *Let n, m, q, and k be integers such that k is a constant. Let $f(n)$ be some function that grows superpolynomially in n. Let q be exponentially large and m polynomially sized in some security parameter n, and let $q' = poly(n) f(n)$. Let σ be a discrete Gaussian parameter such that $\sigma \geq \frac{q}{\eta}$ and let $\mathcal{D}_{\Psi_\sigma}$ denote a discrete Gaussian with parameter σ. Let $\mathcal{D}_{\mathbf{S}}$ be any distribution over $\mathbb{Z}^m$ that outputs vectors bounded by some polynomial in n.*

An adversary that solves the $(k, 1, m, q, \mathcal{D}_{\mathbf{S}}, \mathcal{U}(\mathbb{Z}_q), \mathcal{D}_{\Psi_\sigma})$-k-LWE problem with non-negligible advantage can be used to solve the $\left(\log_{q'}(q), \sigma f(n)\right)$-GapSVP problem with non-negligible advantage.

Proof. The proof follows from a simple hybrid argument outlined in the table above and by just applying the lemmas in sequence without much additional thought. We explain each step below. For the sake of clarity, we ignore small factors in the discussion below (and these don't matter anyway because we are only focused on an asymptotic result).

- We start by reducing to k-LWE in dimension 1 from LWE in dimension 1, with the only loss being a superpolynomial factor in the noise parameter. This follows from the one lemma that did not follow from previous work, lemma .
- The modulus switching lemmas require a key with "small" entries, so we next use a reduction to LWE in dimension 1 with uniform key from LWE in dimension 1 with small key. This is exactly what is stated in lemma 14.

- We now have a dimension 1 LWE instance with a "small" key. We use lemma 16 to reduce to this LWE instance from an LWE instance with a tailored modulus of the form q'^n so that we can easily apply the core modulus switching lemma.
- Given an LWE instance in dimension 1 with modulus q'^n, we apply the core modulus-switching lemma–lemma 17–to reduce to this from an LWE instance in dimension n with modulus q'. This allows us to reach our desired "standard" dimension. We note that the key in both of these instances is still "small" and of the noise distribution.
- We next apply lemma 15 to reduce to LWE with a small key (drawn from the noise distribution, or a similar distribution) from LWE with a uniform key. After this step we have essentially a "standard" LWE problem.
- Finally, we apply the famous theorem of Brakerski *et al.* to reduce to "standard" LWE from the Gap-SVP problem.

This completes the proof, which is essentially just a combination of a new result on k-LWE and an application of modulus switching. □

Note that $\log_{q'}(q)$ will be quite large (possibly even polynomial, depending on the choice of q), so the Gap-SVP problem reduced from here seems very likely to be hard.

C Instantiation Using Elliptic Curves

We next outline how our invariant construction might be instantiated with elliptic curve isogenies. Our candidate construction(s) here are relatively high-level and need substantially more study on their security properties, particularly in light of some recent cryptanalysis on isogenies [CD22b, MM22b].

Elliptic Curve Isogenies. We briefly outline some properties of elliptic curves and isogenies that we will use in our candidate constructions. For a full treatment, we highly recommend [DF17]. *Elliptic curves* are projective curves of genus one with a specified base point. Elliptic curves over finite fields k are often defined in *Weierstrass form*, consisting of all points in the locus of the equation $y^2 = x^3 + ax + b$ and the point at infinity in $\mathbb{P}^2\left(\overline{k}\right)$, where we use $\overline{k}$ to denote the algebraic closure of k. The *j-invariant* of an elliptic curve E in Weierstrass form is defined as

$$j(E) = 1728\frac{4a^3}{4a^3 + 27b^2}.$$

Two elliptic curves E_1 and E_2 are isomorphic if and only if they have the same invariant.

An *isogeny* $\psi : E \to E$ is a surjective group morphism between elliptic curves. We note that isogenies only exist between elliptic curves with the same number of points. We denote the number of points of a curve as $\#E(k)$, or sometimes just $\#E$. Very roughly speaking, isogeny-based cryptography is built on the fact that for certain elliptic curves E and related isogenies ψ, it is possible to efficiently compute $\psi(E)$, but given two elliptic curves E_1 and E_2, it is hard to find an isogeny ψ (or set of isogenies applied in sequence) so that $\psi(E_1) = E_2$.

Background Ideas. Suppose we start with a hypothetical example. Let X be the set of elliptic curves (perhaps represented by their j-invariants) where the number of points on the curve has the form $\ell * q$ for a prime q and some small ℓ. This can be generalized; having several small factors could work as well, or also potentially restricting to super-singular curves.

The invariant could be the number of points on the curve. The $\sigma_{y,i}$ are degree-ℓ isogenies, or if we are generalizing to multiple small factors, σ_i will range over low-degree isogenies. Basically, we choose some arbitrary way of mapping $[r]$ to kernels of the ℓ-torsion, and then σ_i is applying the isogeny defined by that kernel. The inverse of an isogeny is just the dual isogeny, which can be efficiently computed.

The orbits O then correspond to sets of elliptic curves that can be reached by sequences of degree-ℓ isogenies. Under some Ramanujan-Petersson conjecture [CLG08] (or alternatively GRH [JMV09]), action by small-degree isogenies gives an expander. This should give us $\lambda_2 \leq 1 - \delta$ as needed. We also see that isogenies preserve the number of points on the curve, so the invariant property is satisfied.

For security, the path finding is probably hard, as it is essentially the basis of isogeny cryptography [Cou06, RS06]. The plain knowledge of path is false: since we can compute elliptic curves with a given size, we have an invertible invariant. However, the knowledge of path for invertible invariants could possibly be true: perhaps the only way of obtaining two elliptic curves with the same number of points is to either:

- Sample an elliptic curve E_1 and then follow a sequence of isogenies from it
- Sample an elliptic curve E_1, compute the number of points on the curve, and then construct a curve E_2 with that many points using the known algorithms for doing so (or potentially follow a sequence of isogenies from E_2).

If so, then the knowledge of path assumption holds. Lastly, the invariant inversion assumption could plausibly be true: given an elliptic curve, it seems likely that an adversary cannot construct random coins for the elliptic curve construction procedure that produce the given elliptic curve.

Instantiating These Ideas. While it may seem straightforward to build a candidate quantum lightning scheme using the template above, it unfortunately is not so straightforward. To start, generating superpositions over X, where X is the set of all ellitpic curves with some (even polynomially likely) property seems difficult. In fact, we do not even know how to generate a uniform superposition over all elliptic curves efficiently. Recent work [MMP22] explains the known approaches to sampling uniform *supersingular* elliptic curves classically, and unfortunately none of it is amenable to sampling a uniform quantum superposition. Many of the most common ways to sample elliptic curves use random walks, and generating superpositions this way makes it difficult to avoid a hard index erasure problem [AMRR11].

There may also be a duality: the easier the sets of elliptic curves are to sample, the more difficult it is to prove or argue security (since the number of isogenies that are computable may be less or less sophisticated). We leave it open to future work to instantiate our framework (or something similar) from elliptic curve isogenies, and instead lay out a rough sketch of what such an instantiation might be like below. To do this, we start by making a conjecture on the efficient samplability of certain elliptic curves.

Conjecture 20. There exists an efficient quantum algorithm $\mathcal{A}$ to sample a uniform superposition over some distribution $\mathcal{E}$ of elliptic curves over some finite field $\mathbb{F}_p$ with the following properties:

- Given two random elliptic curves $E_1, E_2 \in \mathcal{E}$ such that E_1 and E_2 are isogenous, there is no efficient (quantum) algorithm to find an isogeny ψ such that $\psi(E_1) = E_2$. This is analogous to the *path-finding assumption* holding.

- Let $I(E_0)$ denote the number of points on an elliptic curve, and consider an algorithm $I^{-1} : \mathbb{Z} \times \{0,1\}^{\ell}$ that takes an integer k and a random bit string b and outputs an elliptic curve E_0 with k points. We require the following two properties (which imply the knowledge of path assumption for invertible invariants, and the inversion inverting assumption be true):

  - Given any algorithm that outputs two elliptic curves E_1 and E_2 with k points, there exists an extractor E that can either compute an isogeny ψ such that $\psi(E_1) = E_2$ or it can find randomness t and an isogeny ψ' such that either $\psi'(I^{-1}(k,t)) = E_1$ or $\psi'(I^{-1}(k,t)) = E_2$.
  - No efficient adversary can do the following: sample an (arbitrary) elliptic curve E_1 with k points, and then, for a randomly selected elliptic curve E_2, compute randomness t and an isogeny ψ such that $\psi(I^{-1}(k,t)) = E_2$.

We note that our chosen invarant I is the number of points on the elliptic curve. This is necessary because this is the set of curves that are isogenous. Note that it is possible, in general, to compute an elliptic curve with a certain number of points [BS04], so we need to use the invertible invariant form of our quantum lightning construction. We note that $\mathcal{A}$ could, in theory, restrict the set of curves to supersingular curves.

We emphasize that, while we cannot currently come up with algorithms to satisfy this conjecture, it does not seem like a fundamentally impossible problem to us. In fact, it might be doable if we knew of an algorithm to sample a uniform superposition of (all) elliptic curves over $\mathbb{F}_p$. If we had such a superposition, we could compute the number of points using Schoof's algorithm [Sch95] or some related algorithm and store this in an adjacent register. Then, we could compute a bit that indicates whether or not the number of points on the curve satisfies some property, and then measure this bit. If 1, we could continue (and thus have a superposition over curves where the number of points had some property), and if 0, abort. If successful, we now have a uniform superposition over elliptic curves where the number of points on the curve satisfies some property.[16]

We next provide a an instantiation of a quantum lightning scheme from elliptic curve isogenies assuming that Conjecture 20 holds. A construction of quantum lightning from elliptic curve isogenies might look something like the following:

Minting. To mint an instance of quantum money/lightning, we would do the following:

- Sample a uniform superposition over all non-degenerate elliptic curves using the (conjectured) sampling algorithm $\mathcal{A}$.

- Measure the number of points on the curve (i.e. compute the invariant I). This becomes the serial number of the note.

[16] We would need this measurement to output 1 with polynomial probability, which would heavily restrict our properties we could use here.

Verification. Our verification procedure would be very similar to as described in the walkable invariant construction.

- To verify, we would apply isogenies that induce a random walk on the isogeny graph. We simply need to apply enough isogenies so that the graph "mixes".
- Then we simply apply the check as described in the walkable invariant construction to ensure that we still have a uniform superposition over the orbits.

We do not know if it is even possible to instantiate such a scheme, as it would likely require new ideas in isogeny-based cryptography. However, we think it is an enticing direction for future research. We note that, given Conjecture 20, security immediately follows from the security of our invariant money construction. So, if it is possible to come up with algorithms that satisfy the conjecture, we can build secure quantum lightning from elliptic curve isogenies.

D Functional Encryption-inspired Instantiation

We present another candidate instantiation of invariant quantum money. We call it a "functional encryption-inspired" instantiation because many components are functional encryption-like, but the security properties we look for are very different than what is typically required in functional encryption definitions.

At a high level, we need the following components:

- A secret-key functional encryption scheme FE for general functions with the following additional properties:
  - An *invertible* rerandomization algorithm ReRand that allows for ciphertext rerandomization
  - Obliviously sampleable ciphertexts.
- A (collision resistant) invariant function H or a family of such functions $\mathcal{H}$.

We will specify the properties we need for the above building blocks in detail.

D.1 Re-randomizable Functional Encryption

We first have the basic algorithms for an FE scheme:

Basic Functional Encryption. A secret-key functional encryption scheme FE consists of the following (basic) algorithms:

- $\mathsf{FE.Setup}(1^\lambda) \to (\mathsf{pp}, \mathsf{msk})$: a polynomial time algorithm that takes the security parameter as input and outputs public parameters pp and a master secret key msk.
- $\mathsf{FE.keygen}(\mathsf{msk}, f) \to \mathsf{sk}_\mathsf{f}$: a polynomial time algorithm that takes as input the master secret key msk and a function description f and outputs a corresponding secret key sk_f.
- $\mathsf{FE.Enc}(\mathsf{msk}, m, r) \to \mathsf{ct}$: a polynomial time algorithm that takes the master secret key msk, a message m and randomness r, outputs a ciphertext ct.

- $\mathsf{FE.Dec}(\mathsf{sk_f}, \mathsf{ct}) \to \mathsf{m}$: a polynomial time algorithm that takes a secret key $\mathsf{sk_f}$ and ciphertext encrypting message m and outputs a result y.

We note that Setup and keygen may additionally take randomness, but we omit that in our description for simplicity. To construct quantum money, we need the following additional properties:

Re-randomization We additionally need the following algorithm:

- $\mathsf{FE.ReRand}(\mathsf{pp}, \mathsf{ct}_{m,r}, r_\delta) \to \mathsf{ct}_{m,r'}$: takes in public parameters pp, a ciphertext $\mathsf{ct}_{m,r} = \mathsf{FE.Enc}(\mathsf{msk}, m, r)$, a string r_δ; outputs a new ciphertext $\mathsf{ct}_{m,r'}$.

We also require the ReRand algorithm to be *invertible*: there exists an efficient ReRand^{-1} such that if $\mathsf{ct}_{m,r'} \leftarrow \mathsf{ReRand}(\mathsf{pp}, \mathsf{ct}_{m,r}, r_\delta)$, we can compute $\mathsf{ct}_{m,r} \leftarrow \mathsf{ReRand}^{-1}(\mathsf{pp}, \mathsf{ct}_{m,r'}, r_\delta)$.

Randomness Recoverability In order for the re-randomization algorithm(and its inverse) to be useful in the context of our scheme, we also need an efficient procedure to recover the randomness from the ciphertext:

- $\mathsf{RecoverR}(\mathsf{msk}, \mathsf{ct}_{m,r}) \to r$: a deterministic algorithm takes in the the master secret key and a ciphertext $\mathsf{ct}_{m,r}$; outputs the randomness r.

Note that in our scheme, it suffices to let $\mathsf{RecoverR}$ output both the message m and randomness r. Therefore, we can just let $\mathsf{RecoverR}(\mathsf{msk}, \cdot)$ and the decryption using the master secret key be the same algorithm.

Obliviously Sampleable Ciphertexts. There exists a bijective function $G : \{0,1\}^{n+\ell} \to \mathcal{C}$ where $\mathcal{C}$ is the space of all possible ciphertexts, n is the length of message and ℓ is the length of the randomness used in encryption. We also give out the inverse function $G^{-1} : \mathcal{C} \to \{0,1\}^{n+\ell}$.

Note that this function G is independent of the encryption function $\mathsf{FE.Enc}(\mathsf{msk}, \cdot)$, and, informally speaking, G should not be "useful" to any adversary attempting to attack the scheme. One simple example of G is the identity function, when the ciphertexts are all possible strings in $\{0,1\}^{n+m}$. However, we define G more generally since it may not be the case that the ciphertext space is dense.

Remark 6. An alternative requirement to obliviously sampleable ciphertexts is to require that the encryptions of random messages are statiscally close to uniform random strings. We note that this is essentially equivalent to G being the identity function.

Correctness. A functional encryption scheme is correct for a family of functions $\mathcal{F}$ if for all deterministic $f \in \mathcal{F}$, all messages $m \in \mathcal{M}$, all randomness $r \in \mathcal{R}$:

$$\Pr\left[\mathsf{Dec}(\mathsf{sk_f}, \mathsf{ct}) = \mathsf{f}(\mathsf{m}) \;\middle|\; \begin{array}{c} (\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda), \\ \mathsf{sk_f} \leftarrow \mathsf{keygen}(\mathsf{msk}, \mathsf{f}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{msk}, m, r) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Note that the above correctness should also hold for re-randomized ciphertexts: that is, for all f, all messages $m \in \mathcal{M}$, all randomness $r, r_\delta \in \mathcal{R}$:

$$\Pr\left[\mathsf{Dec}(\mathsf{sk_f}, \mathsf{ct}') = \mathsf{f}(\mathsf{m}) \;\middle|\; \begin{array}{c} (\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda, k), \\ \mathsf{sk_f} \leftarrow \mathsf{keygen}(\mathsf{msk}, \mathsf{f}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{msk}, m, r) \\ \mathsf{ct}' \leftarrow \mathsf{ReRand}(\mathsf{pp}, \mathsf{ct}, r_\delta) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Regarding encryption security, we do not require the usual indistinguishability based definition. We instead present the following "Hardness of path finding" definition, analogous to 1:

Definition 9 ((Single-key) Ciphertext Path-Finding Security Game)**.** The above FE scheme is secure if for all functions f and all admissible non-uniform QPT $\mathsf{A}_1, \mathsf{A}_2$ with quantum advice $\{\langle\psi_{\mathsf{aux}\lambda}\}\rangle_{\lambda\in\mathbb{N}}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$:

$$\Pr\left[p = \mathsf{A}_2(1^\lambda, \mathsf{pp}, \mathsf{ct}_1, |\mathsf{st}\rangle) \;\middle|\; \begin{array}{c} (\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{ct}_{m_0,r_0}|\mathsf{st}\rangle) \leftarrow \mathsf{A}_1(1^\lambda, \mathsf{pp}, \mathsf{sk_f} \leftarrow \mathsf{keygen}(\mathsf{msk}, \mathsf{f})) \\ \mathsf{ct}_{m_0,r_1} \leftarrow \mathsf{Enc}(\mathsf{msk}, m_0, r_1), r_1 \leftarrow \mathcal{R}, r_1 \neq r_0 \end{array}\right] \leq \mathsf{negl}(\lambda)$$

The above p needs to be a path between ct_{m_0,r_0} and ct_{m_0,r_1}. That is, a sequence of inputs of the form (ct, r_δ) to the rerandomization algorithm $\mathsf{FE.ReRand}(\mathsf{pp}, \cdots)$ that starts from ct_{m_0,r_0} and ends with ct_{m_0,r_1}.

$\mathcal{A}$ is admissible if and only if it makes a single key query to the oracle $\mathsf{keygen}(\mathsf{msk}, \cdot)$.

Note that in the functional key query phase of the above security game, the adversary may attempt send a superposition query to the challenger; but the challenger can measure the register that stores the circuit f's description so that it does not get to query on a superposition of circuits.

Remark 7. Informally speaking, the above game allows the adversary to choose a ciphertext, and then the challenger chooses a ciphertext which is an encryption of the same message as the adversarially chosen ciphertext. We note that this is a stronger notion of security than if the challenger sampled two random ciphertexts of the same (randomly chosen) message, and that this strengthened notion of security is essential for our path-finding assumption to hold.

Knowledge of Path Assumption for Rerandomizable FE We also make analogous assumptions to 2 on the knowledge of re-randomization path between two random encryptions of the same message:

Assumption 5. Let A be a quantum polynomial time adversary A that is unitary (in the above sense where there are no measurements except the output registers), given $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{sk_H} \leftarrow \mathsf{keygen}(\mathsf{msk}, \mathsf{H})$.

Let E be a quantum polynomial time extractor that is given A's final output as well as its final state. Let $(\mathsf{ct}_1, \mathsf{ct}_2)$ be the A's output, and p be the output of E.

Let B be the event that (1) $H(\mathsf{ct}_1) = H(\mathsf{ct}_2)$, but (2) p is not a path between ct_1 and ct_2. In other words, B is the event that A outputs two elements with the same invariant but E fails to find a path between them.

The *knowledge of path assumption for rerandomizable FE* is that, for any quantum polynomial time unitary A, there exists a quantum polynomial time E such that $\Pr[B]$ is negligible.

Other variants of the assumption 3,4 can be made analogously.

Remark 8. We can in fact modify the above requirement of $H(\mathsf{ct}_1) = H(\mathsf{ct}_2)$ into "ct_1 and ct_1 are encryptions of the same message m". As we will see later, the case that they are not of the same message can be ruled out by collision resistance of H.

D.2 Quantum Money from Re-randomizable FE

Our candidate quantum money construction from re-randomizable FE follows from the framework of walkable invariants in 8.1. We will describe its instantiation with a rerandomizable FE FE with the properties described previously, as well as a collision resistant hash function.

Setup. the setup algorithm runs the FE setup and samples $H \leftarrow \mathcal{H}$, computes $\mathsf{sk}_\mathsf{H} \leftarrow \mathsf{FE.keygen}(\mathsf{msk}, \mathsf{H})$ of collision resistant hash function H. Publishes $\mathsf{FE.pp}, \mathsf{sk}_\mathsf{H}$.

Minting. First, prepare a uniform superposition over all strings of length $n + \ell$, where n is the message length and ℓ is the randomness length $\sum_{x\in\{0,1\}^{n+\ell}} |x\rangle$.

- Compute the oblivious sampling function G in an output register to obtain $\sum_{x\in\{0,1\}^{n+\ell}} |x\rangle|G(x)\rangle$.
- Use the inverse of the oblivious sampling function, G^{-1} to remove the input register $\sum_{x\in\{0,1\}^{n+\ell}} |x + G^{-1}(G(x))\rangle|G(x)\rangle = \sum_{x\in\{0,1\}^{n+\ell}} |G(x)\rangle$. Now equivalently, we obtain a uniform superposition of all possible ciphertexts $\sum_{m,r} |\mathsf{ct}_{m,r}\rangle$.
- Compute $\mathsf{FE.Dec}(\mathsf{sk}_\mathsf{H}, \cdot)$ coherently on the state above, measure the output register to obtain serial number y and money state:

$$|P_y\rangle := \frac{1}{\sqrt{|P_y|}} \sum_{m:H(m)=y;r} |\mathsf{ct}_{m,r}\rangle$$

  where P_y is the set of pre-images of y.

Verification. The verification procedure is the same as the one described in 8.1. Here, the permutation σ is the rerandomization operation $\mathsf{FE.ReRand}(\mathsf{pp}, \cdot, r_\delta)$, specified by randomness r_δ (σ^{-1} corresponds to ReRand^{-1}, accordingly). The "orbit" O_m for message m corresponds to all possible encryptions of m.

We note that correctness and security should follow immediately from the invariant money scheme if the path-finding assumptions and knowledge of path assumptions hold.

D.3 Candidate Construction for Re-randomizable FE

We briefly sketch a candidate construction for the FE scheme with the above properties we need. We leave the construction as a sketch because we do not know how to fully instantiate it, but think a full, concrete instantiation from reasonably trusted assumptions is excellent future work.

- The encryption procedure is a permutation $P(\mathsf{msk}_\mathsf{Eval}, \cdot)$ with puncturable secret key $\mathsf{msk}_\mathsf{Eval}$. To show the path-finding security defined in 9, we (informally) characterize the security of the puncturable permutation we use as follows:

– Let us denote the secret key as $\mathsf{msk}_{\mathsf{Eval}}$. The adversary A is allowed to make queries on both forward evaluations $P(\mathsf{msk}_{\mathsf{Eval}}, \cdot)$ as well as inversion evaluations $P^{-1}(\mathsf{msk}_{\mathsf{Eval}}, \cdot)$; it is also given a "suffix" rerandomization program described as the above rerandomization program in the FE scheme. A can then submit a challenge "prefix" m. The challenger samples a random r and appends it to m[17], computes $\mathsf{Eval}_{m,r} = P(\mathsf{msk}_{\mathsf{Eval}}, m||r)$ and puncture the key at value $\mathsf{Eval}_{m,r}$. It gives $\mathsf{Eval}_{m,r}$ and the punctured key $\mathsf{msk}^*_{\mathsf{Eval}}$ to A. A finally outputs a value v and wins if and only if $v = P^{-1}(\mathsf{msk}_{\mathsf{Eval}}, \mathsf{Eval}_{m,r})$.

– Note that the usual indistinguishability based security notion(i.e. pseudorandomness of the evaluation at punctured points) does not work here, since A knows the evaluation starts with prefix m; we therefore rely on a search-type security.

- To encrypt, we apply $P(\mathsf{msk}_{\mathsf{Eval}}, \cdot)$ on the concatenation of message m and randomness r. We consider the master secret key msk for FE to contain both the (forward) evaluation key of the permutation and the key for inversion.

- The functional decryption key for a function f is an obfuscated program that hardcodes $\mathsf{msk}_{\mathsf{Eval}}$ and function f: on input ciphertext $P(\mathsf{msk}_{\mathsf{Eval}}, m||r)$, it decrypts the ciphertext to obtain m using $\mathsf{msk}_{\mathsf{Eval}}$; then outputs $f(m)$. The functional decryption procedure is hence simply running the obfuscation program on the ciphertext.

- Randomness recoverability follows from invertibility of the permutation. The rerandomization algorithm is also an obfuscated program that hardcodes $\mathsf{msk}_{\mathsf{Eval}}$ and takes input ciphertext $P(\mathsf{msk}_{\mathsf{Eval}}, m||r)$ and r_δ: it inverts the input ciphertext to obtain both m and r; computes $r' = r \oplus r_\delta$; finally outputs the re-encryption $P(\mathsf{msk}_{\mathsf{Eval}}, m||r')$. This re-randomizarion procedure is clearly invertible.

- Oblivious sampleability follows from the fact that encryptions of random messages in the above scheme are uniform random strings. This follows from the fact that P is a permutation.

- For our CRHF H, we would like a post-quantum CRHF candidate, for example the SIS hash function[18].

Most of the above programs are constructible assuming indistinguishability obfuscation[GGH+16, DN21, BKW17, BLW17]. The permutation we need is trickier to handle: one possible notion we can take use is prefix-constrained PRP discussed in [BKW17] and its construction remains an open problem. Besides, we need the evaluation key to be puncturable. [BKW17] also pointed out that puncturable PRP is impossible for the usual security notion of pseudorandomness at punctured points. But since we do not require our permutation to have such strong indistinguishability based security, the impossibility does not apply here.

[17] More specifically, r is sampled through rejection sampling so that it does not collide with any suffix in previously queried $m||r'$ and $P(\mathsf{msk}_{\mathsf{Eval}}, m||r')$ with the challenge prefix m. A is not allowed to query on the challenge value $\mathsf{Eval}_{m,r}$ after the challenge phase. These queries can all be seen as classical for the same reason in definition 9.

[18] Note that using a collapsing hash function as the invariant in an invariant money scheme does not lead to an attack: intuitively, it only "collapses" the superposition over different orbits, but the superposition of all elements in the same orbit remains hard to clone, which our verification essentially checks.

Ciphertext Path-Finding security Now we base the path-finding security (see definition 9) on the permutation above. We first consider a weaker security called selective security: the adversary has to commit to the challenge messages m_0, m_1 at the beginning of the security game, before seeing the public parameters pp.

- After the adversary A_1 submits ct_{m_0,r_0} to the reduction and the reduction submits it to an inversion oracle of the permutation challenger to obtain (m_0, r_0). The reduction then submits m_0 as its challenge prefix to the permutation challenger.
- The challenger samples r_1, append it to m_1. It computes a value $P(\mathsf{msk}_{\mathsf{Eval}}, m_0||r_1)$ and puncture the key at this value. Let us call this value ct_{m_0,r_1} and the punctured key $\mathsf{msk}^* = \mathsf{puncture}(\mathsf{msk}_{\mathsf{Eval}}, \mathsf{ct}_{m_0,r_1})$.
- The reduction receives ct_{m_0,r_1} and the punctured key msk^*. It can therefore prepare the following program's obfuscation as the functional decryption key on the adversary's query f:
  - Input: ct
  - Harcoded: $\mathsf{msk}^*, \mathsf{ct}_{m_0,r_1}, y = f(m_0)$
  - If $\mathsf{ct} = \mathsf{ct}_{m_0,r_1}$: output y.
  - Else: compute $(m||r) \leftarrow P(\mathsf{msk}^*, \mathsf{ct})$; output $f(m)$.

  The reduction also sends ct_{m_0,r_1} as the challenge ciphertext.
- Suppose the adversary is able to produce a path between ct_{m_0,r_0} and ct_{m_0,r_1}, then the reduction can use the path to compute r_1 and thus knows $m_0||r_1$, which is the evaluation of $P^{-1}(\mathsf{msk}^*, \cdot)$ at the punctured point ct_{m_0,r_1}, while presumably it should not be able to compute this value.

In the quantum money scheme, we need to give out the functional key before the adversary hands in the forged money states(i.e. the challenge ciphertext) and thus we need adaptive security instead of selective security. This can be achieved through complexity leveraging (the reduction needs to guess the correct m_0; guessing r_0 is not necessary) with subexponential security assumption, as it is conventionally dealt with for FE [BB04].

Quantum Lightning Security. We discuss the security of the above quantum lightning scheme based on functional encryption.

First, the verification correctness is satisfied since the re-randomization procedure is an invertible permutation on the ciphertexts of the same message indexed by the randomness, and therefore 8.2 applies.

For security: suppose there is an adversary that produces two valid money states with the same serial number and suppose measuring both of the money states in computational basis, there are possible two events for the money states it produced:

- Event 1: both states produce two ciphertexts $\mathsf{ct}_{m,r_1}, \mathsf{ct}_{m,r_2}$, which are encryptions of the same message, with all but negligible probability.
- Event 2: with non-negligible probability, two money states give encryptions of different messages, $\mathsf{ct}_{m_1,r_1}, \mathsf{ct}_{m_2,r_2}$, where $H(m_1) = H(m_2)$.

First, we can observe the probability that Event 2 happens is negligible: otherwise such an adversary will help break the collision resistance of H (this case can also be covered by the knowledge-of-path assumption 5 if we do not require the two outputs $(\mathsf{ct}_1, \mathsf{ct}_2)$ by A to be of the same message. But we can rule it out completely here by collision resistance).

To rule out Event 1, we first take use of the hardness of ciphertext path-finding security 9 that we have shown. Then combining with the knowledge-of-path assumptions defined in 5, we would be able to argue that Event 1 happens with negligible probability, just as shown in section 8.4, thus ruling out any quantum lightning adversary. The proof would be largely identical so we omit it here.

Ideally we would want to rule out Event 1 without the knowledge-of-path assumptions. However, such a proof is likely to be beyond the power of existing functional encryption or even indistinguishability obfuscation techniques.

E Instantiation from Classical Oracles and Group Actions

We present yet another instantiation of invariant quantum money. In this case, we present a *classical* oracle-based scheme, whereas some previous lightning schemes are based on quantum oracles.

This scheme provides an alternative view on both of our isogenies over elliptic curve instantiation and our functional encryption instantiation: group actions can be viewed as an abstraction for certain isogeny-based cryptography, and the oracles we will use are abstractions of the algorithms in the functional encryption scheme.

We show that any adversary that can break this scheme can be used to solve the discrete logarithm problem over (quantum-accessible) generic group actions, assuming the knowledge of path property.

E.1 Preliminaries: Cryptographic Group Actions

First and foremost we provide some preliminaries for group actions.

We define cryptographic group actions following Alamati *et al.* [ADMP20], which are based on those of Brassard and Yung [BY91] and Couveignes [Cou06]. Our presentation is borrowed from [MZ22].

Definition 10. (Group Action) A group G is said to *act on* a set X if there is a map $\star : G \times X \to X$ that satisfies the following two properties:

1. Identity: If e is the identity of G, then $\forall x \in X$, we have $e \star x = x$.
2. Compatibility: For any $g, h \in G$ and any $x \in X$, we have $(gh) \star x = g \star (h \star x)$.

We may use the abbreviated notation $(G, X, \star)$ to denote a group action. We extensively consider group actions that are *regular*:

Definition 11. A group action $(G, X, \star)$ is said to be *regular* if, for every $x_1, x_2 \in X$, there exists a *unique* $g \in G$ such that $x_2 = g \star x_1$.

We emphasize that most results in group action-based cryptography have focused on regular actions. As emphasized by [ADMP20], if a group action is regular, then for any $x \in X$, the map $f_x : g \mapsto g \star x$ defines a bijection between G and X; in particular, if G (or X) is finite, then we must have $|G| = |X|$.

In this paper, unless we specify otherwise, we will work with *effective* group actions (EGAs). An effective group action $(G, X, \star)$ is, informally speaking, a group action where all of the (well-defined) group operations and group action operations are efficiently computable, there are efficient ways to sample random group elements, and set elements have unique representation.

In this work we will also use the *group action discrete logarithm* problem.

Definition 12. (Group Action Discrete Logarithm) Given a group action $(G, X, \star)$ and distributions $(\mathcal{D}_X, \mathcal{D}_G)$, the group action discrete logarithm problem is defined as follows: sample $g \leftarrow \mathcal{D}_G$ and $x \leftarrow \mathcal{D}_X$, compute $y = g \star x$, and create the tuple $T = (x, y)$. We say that an adversary solves the group action discrete log problem if, given T and a description of the group action and sampling algorithms, the adversary outputs g.

E.1.1 A Generic Group Action Framework

In this section, we present the generic group action framework from [MZ22]. Their framework is based on the generic group framework of Shoup [Sho97]. The following is taken mostly verbatim from [MZ22].

Let G be a group of order n, let X be a set that is representable by bit strings of length m, and let $(G, X, \star)$ be a group action. We define additional sets S_G and S_X such that they have cardinality of at least n and 2^m, respectively. We define *encoding functions* of σ_G and σ_X on S_G and S_X, respectively, to be injective maps of the form $\sigma_G : G \to S_G$ and $\sigma_X : X \to S_X$.

A generic algorithm $\mathcal{A}$ for $(G, X, \star)$ on (S_G, S_X) is a probabilistic algorithm that behaves in the following way. It takes as input two *encoding lists* $(\sigma_G(g_1), ..., \sigma_G(g_k))$ and $(\sigma_X(x_1), ..., \sigma_X(x_{k'}))$ where each $g_i \in G$ and $x_i \in X$ and where σ_G and σ_X are encoding functions of G on S_G and X on S_X, respectively. As the algorithm executes, it may consult two oracles, $\mathcal{O}_G$ and $\mathcal{O}_X$.

The oracle $\mathcal{O}_G$ takes as input two strings y, z representing group elements and a sign "+" or "−", computes $\sigma_G\left(\sigma_G^{-1}(y) \pm \sigma_G^{-1}(z)\right)$. The oracle $\mathcal{O}_X$ takes as input a string y representing a group element and string z representing a set element, and computes $\sigma_X\left(\sigma_G^{-1}(y) \star \sigma_X^{-1}(z)\right)$. As is typical in the literature, we can force all queries to be on either the initial encoding lists or the results of previous queries by making the string length m very long. We typically measure the running time of the algorithm by the number of oracle queries.

It is also possible extend the generic group action model to the quantum setting, where we allow *quantum* queries to the oracles. We model quantum queries in the usual way: $\mathcal{O}_G \sum_{y,z,\pm,w} \alpha_{y,z,\pm,w}|y, z, \pm, w\rangle = \sum_{y,z,\pm,w} \alpha_{y,z,\pm,w}|y, z, \pm, w \oplus \mathcal{O}_G(y, z, \pm)\rangle$ and $\mathcal{O}_X \sum_{y,z,w} \alpha_{y,z,w}|y, z, w\rangle = \sum_{y,z,w} \alpha_{y,z,w}|y, z, w \oplus \mathcal{O}_X(y, z)\rangle$.

E.1.2 Post-Quantum Instatiation of Group Actions

Some isogeny based group actions are CSIDH [CLM+18], CSI-FiSh [BKV19] as well as its derivatives/applications [DFM20]. We refer the readers to [AFMP20] for a detailed discussion on the classification of various isogeny protocols into group action definitions.

Some other group actions as post-quantum candidates are not isogeny-based, for example [JQSY19].

Remark 9. A few very recent works [CD22b, MM22b, Rob22] break SIDH by showing how to solve the discrete log problem. However, the attack exploits certain extra points that are made public in

SIDH, and these points are exactly one of the reasons that SIDH is *not* a group action. In particular, the attack does not seem to apply to CSI-FISH or CSIDH, the main instantiations of group actions.

E.2 The Oracles

We begin by defining some oracles that we will use.

Let n_0, n_1, n_2 be positive integers. Let G be a prime-order cyclic group such that $|G| >> n_2$, and suppose we have a generic *regular* group action defined by $(G, X, \star)$. We also require $n_0 >> n_1 + |G|$.

We work in the Generic Group Action framework defined in the previous section E.1.1. We will sometimes assume that the oracles $\mathcal{O}_G, \mathcal{O}_X$ are given implicitly.

To implement our invariant function. we first need a collision-resistant function(or a random oracle):

$$H : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$$

With this in mind, we can define other functions as well. We define a set of four functions as follows. These will be our "setup" and "utility" oracles. We need $|S| = 2^{n_0}$; P, Q are bijective funtions (we will generally model these as random oracles subject to certain constraints.).

$$Q : \{0,1\}^{n_0} \to S$$
$$Q^{-1} : S \to \{0,1\}^{n_0}$$
$$P : \{0,1\}^{n_1} \times X \to S$$
$$P^{-1} : S \to \{0,1\}^{n_1} \times X$$

We include a few extra restrictions. We first require that, for all tuples of bit strings $\mathbf{x} \in \{0,1\}^m$ and elements $g \in G$, we have $Q^{-1}(Q(\mathbf{x}, g))$ and $P^{-1}(P(\mathbf{x}, g))$. Moreover, we require that the *output sets* of P and Q be *identical*, although the mappings themselves are random subject to this constraint (and thus, almost certainly different).

We next define our invariant oracle as follows. Let S be the set of bit strings in the output space of P and Q.

$$I : S \to \{0,1\}^{n_2}$$

Note that we are assuming that I takes as input an element $e_{\mathbf{x},x} = P(\mathbf{x}, x)$ of S and returns the invariant $H(\mathbf{x})$ computed in an output register. Note that we can instantiate oracle I as follows: first apply P^{-1} on $e_{\mathbf{x},x}$ to get back to $\mathbf{x}||x$, then apply the function H on $\mathbf{x}$. Since I can only be accessed as an oracle, functionality of P^{-1} is not given out.

Finally, we must define our "random walking" function R:

$$R : S \times G \to S$$

We define R's functionality as: $\forall x \in X, \mathbf{x} \in \{0,1\}^{n_1}, g \in G : R(P(\mathbf{x}, x), g) = P(\mathbf{x}, g \star x)$. R can be implemented using P^{-1} on element $e_{\mathbf{x},x} = P(\mathbf{x}, x)$ to recover $(\mathbf{x}, x)$; then it calls the group action oracle $\mathcal{O}_G$ to apply $g \star x$; finally it outputs the result by applying P on $(\mathbf{x}, g \star x)$. Naturally, to apply a reverse random walk, one would use g^{-1}.

E.3 Quantum Money from Oracles and Group Actions

As before, the quantum money(lightning) scheme in this section follows from the framework of walkable invariants in 8.1. We will define the scheme and then prove that it instantiates a secure walkable invariant.

Setup the setup algorithm generates all of the oracles defined in the previous section. It then publishes Q, Q^{-1}, H, I and R, as well as a description of all the parameters and the relevant oracles for the generic group action $(G, X, \star)$. Note that P, P^{-1} are kept secret.

Minting First, prepare a uniform superposition over all strings of length n_0, giving us the following: $\sum_{s\in\{0,1\}^{n_0}} |s\rangle$.

- Compute the oblivious sampling function Q in an output register to obtain $\sum_{s\in\{0,1\}^{n_0}} |s\rangle|Q(s)\rangle$.
- Use the inverse of the oblivious sampling function, Q^{-1} to remove the input registers $\sum_{s\in\{0,1\}^{n_0}} |s\rangle$. This allows us to obtain a uniform superposition of all possible values $\sum_{s\in\{0,1\}^{n_0}} |Q(s)\rangle$. Equivalently, we have obtained $\sum_{e\in S} |e\rangle = \sum_{\mathbf{x}\in\{0,1\}^{n_1}, x\in X} |P(\mathbf{x}, x)\rangle$.
- Compute the invariant function I coherently on the state above, and then measure the output register to obtain serial number y and money state:

$$|M_y\rangle := \frac{1}{\sqrt{|M_y|}} \sum_{\mathbf{x}:H(\mathbf{x})=y;x} |P(\mathbf{x}, x)\rangle$$

  where M_y is the set of pre-images of y.

Verification The verification procedure is straightforward and the same as the one in 8.1. Note that this is even simpler than defined above because we required G to be a prime-order cyclic group.

Here, the permutation σ is the rerandomization oracle R specified by randomness $g \in G$. The "orbit" $O_{\mathbf{x}}$ corresponds to all possible evaluations of P on a fixed bit string $\mathbf{x}$.

E.4 Quantum Lightning Security

We also have a reasonably reliable hard problem to reduce our path-finding hardness to, namely, the group action discrete log problem.

To prove security, we will first show that the Path-Finding hardness holds in the GGA framework.

Definition 13 (Path-Finding Game for Group Actions)**.** Consider an adversary A playing the following game:

- Give the adversary access to Q, Q^{-1}, R, I (and the oracles in GGA framework).
- The adversary outputs an $e_{\mathbf{x},x} \in S$.
- The challenger then computes a random $e_{\mathbf{x},z} \in O_{\mathbf{x}}$, where the "orbit" $O_{\mathbf{x}}$ corresponds to the set of all possible evaluations of P on a fixed bit string $\mathbf{x}$.
- The adversary wins if it can output a path p between $e_{\mathbf{x},x}$ to $e_{\mathbf{x},z}$, where a path is a sequence of *classical* queries to oracle R.

We next show that for all quantum polynomial-time adversaries A, the probability A wins in the above game is negligible.

Lemma 21. *Any adversary that breaks the Path-Finding game in 13 can be used to solve discrete logarithm on $(G, X, \star)$ (definition 12).*

Proof. This is almost immediate due to the construction of our scheme. Suppose there exists an adversary that outputs a path, i.e. a sequence of classical oracle queries made to R to get from $e_{\mathbf{x},x}$ to $e_{\mathbf{x},z}$, then one can output a sequence of group elements $g_1, \cdots, g_k$ such that $z = (g_k \cdots g_1) \star x$. □

We give a variant for the Knowledge of Path assumption 2 in the generic group action framework:

Assumption 6. Let A be a quantum polynomial time adversary A that is unitary (in the above sense where there are no measurements except the output registers), given oracle access to Q, Q^{-1}, R, I.

Let E be a quantum polynomial time extractor that is given A's final output as well as its final state. Let (e_x, e_z) be the A's output, and p be the output of E.

Let B be the event that (1) $I(e_x) = I(e_z)$, but (2) p is not a path between x and z. In other words, B is the event that A outputs two elements with the same invariant but E fails to find a path between them.

The *knowledge of path assumption for group actions* is that, for any quantum polynomial time unitary A, there exists a quantum polynomial time E such that $\Pr[B]$ is negligible.

The above event B can again be divided into two cases: (1) A's outputs e_x, e_z are in different "orbits", i.e. $e_x = P(\mathbf{x}_1, x)$ and $e_z = P(\mathbf{x}_2, z)$ for some $\mathbf{x}_1 \neq \mathbf{x}_2, H(\mathbf{x}_1) = H(\mathbf{x}_2)$; (2) e_x, e_z are in the same orbit. The first case is ruled out by the collision resistance of H. Thus we can modify the above assumption into requiring that $e_x.e_z$ are in the same orbit.

We can also analogously transform 3, 4 to the group action version.

Therefore, using the path-finding hardness for group actions shown above 21 together with the above assumptions, the security proof will be almost identical to section 8.4, ruling out any quantum lightning adversary.

.

F Instantiation Using Knots

The inspiration for our invariant scheme was the construction of quantum money from knots in [FGH+12]. In this section, we explain how this construction can be modelled in our framework and what assumptions on knots need to be true in order for our security proof to apply. While it's possible that their verification procedure may not satisfy correctness(the Markov chain does not mix) and the security property is challenging to investigate, we believe an alternative view would provide insight into proving/breaking their scheme.

On Knots. Recall that a knot is an embedding of the circle into three-dimensional Euclidean space, or more informally, as the authors of [FGH+12] nicely put it, a loop of string in three dimensions. Informally speaking, two knots are considered to be *equivalent* if they can be morphed into each other without "cutting the string". We say that a function is a *knot invariant* if it has the same value on all equivalent knots.

Knots (of a certain maximum size) can be represented by *planar grid diagrams*, which are $d \times d$ grids containing exactly d Xs and dOs such that each row and column of the grid have exactly one X and one O, and there is no space in the grid with both an X and an O. Alternatively, knots can be represented by a tuple of two disjoint permutations of size d. We note that knots with smaller numbers of "elements" (i.e. with less than d Xs and Os) can be gracefully represented on $d \times d$ grids by leaving certain columns and rows blank.

Reidemeister moves [Rei27] are three types of local moves that can be applied to a planar grid diagram with the properties that any two knots that are reachable from each other by Reidemeister moves are equivalent and that any two equivalent knots must be reachable by the application of a finite number of Reidemeister moves (although the best bound on the minimum number of moves is enormous [CL14]).

It will be useful to think of Reidemeister moves as a set R acting on the set of planar grid diagrams of size d S_d. While each move $r \in R$ has an inverse, it is unfortunately impossible to model this interaction as a group action because the application of Reidemeister moves may not be associative.

We defer a full description of knots, Reidemeister moves, and associated concepts to [FGH+12]; we will not need the full formalism to convey our ideas here.

F.1 A General Description of the [FGH+12] Scheme

We next provide a general description of the quantum money from knots scheme in [FGH+12]. We omit some of the details of the scheme (some of which are important for its security) so we can present the scheme in its most general form.

Minting. To mint a note, start by constructing a *specific* superposition over grid diagrams of size d, which we refer to as $|S_d\rangle$ over S_d. In particular, a uniform distribution over all possible knots represented by $d \times d$ planar grid diagrams is *not* chosen for security reasons. Instead, knots are weighted in the superposition based on the number of "elements" present in the grid diagram representation according to a Gaussian distribution centered on $\frac{d}{2}$.

After generating $|S_d\rangle$, compute a knot invariant A on the superposition $|S_d\rangle$, store it in an adjacent register, and then measure it, getting some value y. The serial number is the value of this invariant, and the money state becomes $\frac{1}{\sqrt{|\sum_{A(S_d)=y}|}} \sum_{A(S_d)=y} |S_d\rangle$. In [FGH+12], the authors use the Alexander polynomial as the knot invariant of choice.

Verification. To verify, a quantum verification procedure based on a classical Markov chain is applied. Essentially, many randomly chosen Reidemeister moves are applied to the money state superposition with the restriction that Reidemeister moves that would expand the planar grid diagram beyond d dimensions are ignored. This (somewhat) simulates a (not necessarily uniform) random walk on the graph of equivalent knots.

In other words, the verification procedure is almost identical to our process described in section 8. The set of all Reidemeister moves on $d \times d$ grid diagrams form something very close to a set of permutations $\sigma_i : S_d \to S_d$; it turns out to be possible to construct permutations on grid diagrams from Reidemeister moves such that the permutation set enables all possible Reidemeister moves.[19]

F.2 Correctness Properties

In order to fit in our quantum money framework, we require two essential properties of a scheme: efficient generation of superpositions and mixing walks.

[19] To be precise, we just need to pair Reidemeister moves that increase the number of elements in the grid diagram with a corresponding move that "undoes" the transformation, and choose which move we apply based on the number of elements in the grid diagram.

Efficient Generation of Superpositions. The authors of [FGH+12] show clearly how to generate the appropriate superpositions. It is a slightly long presentation so we defer to their paper.

Mixing Walks. In the context of our framework, "orbits" are knots and "permutations" are Reidemeister moves. Our framework requires uniform superpositions over orbits, while the construction in [FGH+12] uses non-uniform superpositions. However, if the mixing process preserves the starting superposition as the authors of [FGH+12] conjecture, then their knot scheme could be fit into our framework with relatively minor modifications to the framework.

However, a formal proof that mixing occurs would be quite difficult and likely involve solving several longstanding open problems in knot theory. A uniform mixing process for knots could only work in polynomial time if the number of Reidmeister moves between all equivalent knots of a certain size were bounded. Unfortunately, the only known bound for the number of Reidemeister moves between arbitrary knots invoves a "tower of exponentials" function [CL14]. While it is known that the number of Reidemeister moves between unknot[20] representations of a certain size is polynomially bounded [Lac15], there are no known results for more general knots.

It may be the case that the restriction to $d \times d$ grids may make proving mixing easier, but we could not effectively utilize this fact.

We emphasize that the lack of a good mixing algorithm may not constitute an attack: it could still be the case that it is hard for an adversary to find a superposition that is unchanged by the existing mixing algorithm. But these sorts of statements and proofs are outside of our framework.

F.3 Security Properties

If we want to build secure quantum lightning from knots using our framework, we would need to show that two assumptions are true: the hardness of path-finding assumption and the knowledge of path assumption.

Hardness of Path-finding. It has long been conjectured that the *knot equivalence problem*–in other words, distinguishing whether or not two knots are equivalent or not–is hard [Dyn03]. The hardness of path assumption in this scheme would correspond to actually finding the Reidemeister move set necessary to transform one knot into another equivalent knot, and any efficient algorithm for this would immediately imply an efficient algorithm for the knot recognition problem. It is known that determining if two knots are equivalent or not is decidable [Lac16], but it is not even known if the problem is in NP. Presumably if there were polynomial-length numbers of moves between equivalent knots of certain sizes, the problem would be in NP, but this is not known.

On the other hand, Lackenby [Lac21] has recently announced a quasipolynomial algorithm for recognizing the unknot. If this could be generalized to other knot equivalences (which may or may not be possible), it would spell trouble for basing cryptographic primitives on the hardness of knot equivalence.

Knowledge of Path Assumption. It is more difficult to assess the knowledge of path assumption over knots. Intuitively, this asks whether or not it is possible to create two equivalent knots without knowing a path between them (in particular, the path must only involve knots that fit on $d \times d$

[20] The unknot is the simplest possible knot: just a circle.

planar grids). We certainly do not see an easy way to do this, but to our knowledge no one has ever studied this problem.