Quantum Lightning Never Strikes the Same State Twice

Mark Zhandry

Princeton University

Abstract. Public key quantum money can be seen as a version of the quantum no-cloning theorem that holds even when the quantum states can be verified by the adversary. In this work, we investigate *quantum lightning* where no-cloning holds *even when the adversary herself generates the quantum state to be cloned.* We then study quantum money and quantum lightning, showing the following results:

- We demonstrate the usefulness of quantum lightning beyond quantum money by showing several potential applications, such as generating random strings with a proof of entropy, to completely decentralized cryptocurrency without a block-chain, where transactions is instant and local.
- We give Either/Or results for quantum money/lightning, showing that either signatures/hash functions/commitment schemes meet very strong recently proposed notions of security, or they yield quantum money or lightning. Given the difficulty in constructing public key quantum money, this suggests that natural schemes do attain strong security guarantees.
- We show that instantiating the quantum money scheme of Aaronson and Christiano [STOC'12] with *indistinguishability obfuscation* that is secure against quantum computers yields a secure quantum money scheme. This construction can be seen as an instance of our Either/Or result for signatures, giving the first separation between two security notions for signatures from the literature.
- Finally, we give a plausible construction for quantum lightning, which we prove secure under an assumption related to the multi-collision resistance of degree-2 hash functions. Our construction is inspired by our Either/Or result for hash functions, and yields the first plausible standard model instantiation of a *non-collapsing* collision resistant hash function. This improves on a result of Unruh [Eurocrypt'16] which is relative to a quantum oracle.

1 Introduction

Unlike classical bits, which can be copied ad nauseum, quantum bits — called qubits — cannot in general be copied, as a result of the Quantum No-Cloning Theorem. No-cloning has various negative implications to the handling of quantum information; for example it implies that classical error correction cannot be applied to quantum states, and that it is impossible to transmit a quantum state over a classical channel. On the flip side, no-cloning has tremendous potential for cryptographic purposes, where the adversary is prevented from various strategies that involve copying. For example, Wiesner [37] shows that if a quantum state is used as a banknote, no-cloning means that an adversary cannot duplicate the note. This is clearly impossible with classical bits. Wiesner's idea can also be seen as the starting point for quantum key distribution [9], which can be used to securely exchange keys over a public channel, even against computationally unbounded eavesdropping adversaries.

In this work, we investigate no-cloning in the presence of computationally bounded adversaries, and it's implications to cryptography. To motivate this discussion, consider the following two important applications:

- A public key quantum money scheme allows anyone to verify banknotes. This remedies a key limitation of Wiesner's scheme, which requires sending the banknote back to the mint for verification. The mint has a *secret* classical description of the banknote which it can use to verify; if this description is made public, then the scheme is completely broken. Requiring the mint for verification represents an obvious logistical hurdle. In contrast, a public key quantum money scheme can be verified locally without the mint's involvement. Yet, even with the ability to verify a banknote, it is impossible for anyone (save the mint) to create new notes.
- Many cryptographic settings such as multiparty computation require a random string to be created by a trusted party during a set up phase. But what if the randomness creator is not trusted? One would still hope for some way to verify that the strings it produces are still random, or at least have some amount of (min-)entropy. At a minimum, one would hope for a guarantee that their string is different from any previous or future string that will be generated for anyone else. Classically, these goals are impossible. But quantumly, one may hope to create proofs that are unclonable, so that only a single user can possibly ever receive a valid proof for a particular string.

The settings above are subtly different from those usually studied in quantum cryptography. Notice that in both settings above, a computationally unbounded adversary can always break the scheme. For public key quantum money, the following attack produces a valid banknote from scratch in exponential-time: generate a random candidate quantum money state and apply the verification procedure. If it accepts, output the state; otherwise try again. Similarly, in the verifiable randomness setting, an exponential-time adversary can always run the randomness generating procedure until it gets two copies of the same random string, along with two valid proofs for that string. Then it can give the same string (but different valid proofs) to two different users. With the current state of knowledge of complexity theory, achieving security against a computationally bounded adversary means computational assumptions are required; in particular, both scenarios imply at a minimum one-way functions.

Unfortunately, most of the techniques developed in quantum cryptography are inherently information theoretic, and porting these techniques over to the computational setting can be tricky task. For example, whereas information-theoretic security can be often proved directly, computational security must always be proved by a reduction to the underlying hard computational problem.

We stress that the underlying problem should still be a *classical* problem (that is, the inputs and outputs of the problem are classical), rather than some quantum problem that talks about manipulating quantum states. For one, we want a violation of the assumption to lead to a mathematically interesting result, and this seems much more likely for classical problems. Furthermore, it is much harder for the research community to study and analyze a quantum assumption, since it will be hard to isolate the features of the problem that make it hard. For this work, we want to:

Combine no-cloning and computational assumptions about classical problems to obtain no-cloning-with-verification.

In addition to the underlying assumption being classical, it would ideally also be one that has been previously studied by cryptographers, and ideally used in other cryptographic contexts. This would give the strongest possible evidence that the assumption, and hence application, are secure.

For now, we focus on the setting of public key quantum money. Constructing such quantum money from a classical hardness assumption is a surprisingly difficult task. One barrier is the following. Security would be proved by reduction, an algorithm that interacts with a supposed quantum money adversary and acts as an adversary for the the underlying *classical* computational assumption. Note that the adversary expects as input a valid banknote, which the reduction must supply. Then it appears the reduction should somehow use the adversary's forgery to break the computational assumption. But if the reduction can generate a single valid banknote, there is nothing preventing it from generating a second — recall that the underlying assumption is classical, so we cannot rely on the assumption to provide us with an un-clonable state. Therefore, if the reduction works, it would appear that the reduction can create two banknotes for itself, in which case it can break the underlying assumption without the aid of the adversary. This would imply that the underlying assumption is in fact false.

The above difficulties become even more apparent when considering the known public key quantum money schemes. The first proposed scheme by Aaronson [2] had no security proof, and was subsequently broken by Lutomirski et al. [29]. The next proposed scheme by Farhi et al. [21] also has no security proof, though this scheme still remains unbroken. However, the scheme is complicated, and it is unclear which quantum states are accepted by the verification procedure; it might be that there are dishonest banknotes that are both easy to construct, but are still accepted by the verification procedure.

Finally, the third candidate by Aaronson and Christiano [3] actually *does* prove security using a classical computational problem. However, in order to circumvent the barrier discussed above, the classical problem has a highly non-standard format. They observe that a polynomial-time algorithm can, by random guessing, produce a valid banknote with some exponentially-small probability

p, while random guessing can only produce n valid banknotes with probability p^n. Therefore, their reduction first generates a valid banknote with probability p, runs the adversary on the banknote, and then uses the adversary's forgery to increase its success probability for some task. This reduction strategy requires a very carefully crafted assumption, where it is assumed hard to solve a particular problem in polynomial time with exponentially-small probability p, even though it can easily be solved with probability p^2.

In contrast, typical assumptions in cryptography involve polynomial-time algorithms and *inverse-polynomial* success probabilities, rather than exponential. (Sub)exponential hardness assumptions are sometimes made, but even then the assumptions are usually closed under polynomial changes in adversary running times or success probabilities, and therefore make no distinction between p and p^2. In addition to the flavor of assumption being highly non-standard, Aaronson and Christiano's assumption — as well as their scheme — have been subsequently broken [32, 1].

Turning to the verifiable randomness setting, things appear even more difficult. Indeed, our requirements for verifiable randomness imply an even stronger version of computational no-cloning: an adversary should not be able to copy a state, even if it can verify the state, and moreover *even if it devised the original state itself.* Indeed, without such a restriction, an adversary may be able to come up with a dishonest proof of randomness, perhaps by deviating from the proper proof generating procedure, that it *can* clone arbitrarily many times. Therefore, a fascinating objective is to

Obtain a no-cloning theorem, even for settings where the adversary controls the entire process for generating the original state.

1.1 This Work: Strong Variants of No-Cloning and Connections to Post-Quantum Cryptography

In this work, we study strong computational variants of quantum no-cloning, in particular public key quantum money, and uncover interesting relationships between no-cloning and various cryptographic applications.

Quantum Lightning Never Strikes the Same State Twice The old adage about lightning is of course false, but the idea nonetheless captures some of the features we would like for the verified randomness setting discussed above. Suppose a magical randomness generator could go out into a thunderstorm, and "freeze" and "capture" lightning bolts as they strike. Every lightning bolt will be different. The randomness generator then somehow extracts a fingerprint or serial number from the frozen lightning bolt (say, hashing the image of the bolt from a particular direction). The serial number will serve as the random string, and the frozen lightning bolt will be the proof of randomness; since every bolt is different, this ensures that the bolts, and hence serial numbers, have some amount of entropy.

Of course, it may be that there are other ways to create lightning other than walking out into a thunderstorm (Tesla coils come to mind). We therefore would like that, no matter how the lightning is generated, be it from thunderstorms or in a carefully controlled laboratory environment, every bolt has a unique fingerprint/serial number.

We seek a complexity-theoretic version of this magical frozen lightning object, namely a phenomenon which guarantees different outcomes every time, no matter how the phenomenon is generated. We will necessarily rely on quantum no-cloning — since in principle a classical phenomenon can be replicated by starting with the same initial conditions — and hence we call our notion *quantum lightning*. Quantum lightning, roughly, is a strengthening of public key quantum money where the procedure to generate new banknotes itself is public, allowing anyone to generate banknotes. Nevertheless, it is impossible for an adversary to construct two notes with the same serial number. This is a surprising and counter-intuitive property, as the adversary knows how to generate banknotes, and moreover has full control over how it does so; in particular it can deviate from the generation procedure any way it wants, as long as it is computationally efficient. Nonetheless, it cannot devise a malicious note generation procedure that allows it to construct the same note twice. This concept of quantum money can be seen as a formalization of the concept of "collision-free" public key quantum money due to Lutomirski et al. [29].

Slightly more precisely, a quantum lightning protocol consists of two efficient (quantum) algorithms. The first is a bolt generation procedure, or storm, 🌧, which generates a quantum state $|↯\rangle$ on each invocation. The second algorithm, Ver, meanwhile verifies bolts as valid and also extracts a fingerprint/serial number of the bolt. For correctness, we require that (1) Ver always accepts bolts produced by 🌧, (2) it does not perturb valid bolts, and (3) that it will always output the same serial number on a given bolt.

For security, we require the following: it is computationally infeasible to produce two bolts $|↯_0\rangle$ and $|↯_1\rangle$ such that Ver accepts both and outputs identical serial numbers. This is true for even for *adversarial* storms 😈 — those that depart from 🌧 or produce entangled bolts — so long as 😈 is efficient.

Applications. Quantum lightning as described has several interesting applications:

- **Quantum money.** Quantum lightning easily gives quantum money. A banknote is just a bolt, with the associated serial number signed by the bank using an arbitrary classical signature scheme. Any banknote forgery must either forge the bank's signature, or must produce two bolts with the same serial number, violating quantum lightning security.
- **Verifiable min-entropy**. Quantum lightning also gives a way to generate random strings along with a proof that the string is random, or at least has min-entropy. Indeed, consider an adversarial bolt generation procedure that produces bolts such that the associated serial number has low min-entropy. Then by running this procedure several times, one will eventually obtain in polynomial time two bolts with the same serial number, violating security.

Therefore, to generate a verifiable random string, generate a new bolt using 🌧. The string is the bolt's serial number, and 🌧 serves as a proof of min-entropy, which is verified using Ver.

- **Decentralized Currency.** Finally, quantum lightning yields a simple new construction of totally decentralized digital currency. Coins are just bolts, except the serial number must hash to a string that begins with a certain number of 0's. Anyone can produce coins by generating bolts until the hash begins with enough 0's. Moreover, verification is just Ver, and does not require any interaction or coordination with other users of the system. This is an advantage over classical cryptocurrencies such as BitCoin, which require a large public and dynamic ledger, and requires a pool of miners to verify transactions. Our protocol does have significant limitations relative to classical cryptocurrencies, which likely make it only a toy object. We hope that further developments will yield a scheme that overcomes these limitations.

Connections to Post-quantum Security One simple folklore way to construct a state that can only be constructed once but never a second time is to use a collision-resistant hash function H. First, generate a uniform superposition of inputs. Then apply the H in superposition, and measure the result y. The state collapses to the superposition $|\psi_y\rangle$ of all pre-images x of y.

Notice that, while it is easy to sample states $|\psi_y\rangle$, it is impossible to sample two copies of the same $|\psi_y\rangle$. Indeed, given two copies of $|\psi_y\rangle$, simply measure both copies. Since these are superpositions over many inputs, each state will likely yield a different x. The two x's obtained are both pre-images of the same y, and therefore constitute a collision for H.

The above idea does not yet yield quantum lightning. For verification, one can hash the state to get the serial number y, but this alone is insufficient. For example, an adversarial storm can simply choose a random string x, and output $|x\rangle$ twice as its two copies of the same state. Of course, $|x\rangle$ is not equal to $|\psi_y\rangle$ for any y. However, the verification procedure just described does not distinguish between these two states.

What one needs therefore is mechanism to distinguish a random $|x\rangle$ from a random $|\psi_y\rangle$. Interestingly, as observed by Unruh [35], this is exactly *the opposite* what one would normally want from a hash function. Consider the usual way of building a computationally binding commitment from a collision resistant hash function: to commit to a message m, choose a random r and output $H(m,r)$. Classically, this is computationally binding by the collision resistance of H: if an adversary can open the commitment to two different values, this immediately yields a collision for H. Unruh [35] shows in the quantum setting, collision resistance — even against quantum adversaries — is not enough. Indeed, he shows that for certain hash functions H it may be possible for the adversary to produce a commitment, and only afterward decide on the committed value. Essentially, the adversary constructs a superposition of pre-images $|\psi_y\rangle$ as above, and then uses particular properties of H to perturb $|\psi_y\rangle$ so that it becomes a different superposition of pre-images of y. Then one simply de-commits to any message by first modifying the superposition and then measuring. This does not violate the collision-resistance of H: since the adversary cannot copy $|\psi_y\rangle$, the adversary can only ever perform this procedure once and obtain only a single de-commitment.

To overcome this potential limitation, Unruh defines a notion of *collapsing* hash functions. Roughly, these are hash functions for which $|x\rangle$ and $|\psi_y\rangle$ are *indistinguishable*. Using such hash functions to build commitments, one obtains *collapse-binding* commitments, for which the attack above is impossible. Finally, he shows that a random oracle is collapse binding.

More generally, an implicit assumption in many classical settings is that, if an adversary can modify one value into another, then it can produce both the original and modified value simultaneously. For example, in a commitment scheme, if a classical adversary can de-commit to both 0 or 1, it can then also simultaneously de-commit to both 0 and 1 by first de-committing to 0, and then re-winding and de-committing to 1. Thus it is natural classically to require that it is impossible to simultaneously produce de-commitments to both 0 and 1. Similarly, for signatures, if an adversary can modify a signed message m_0 into a signed message m_1, then it can simultaneously produce two signed messages m_0, m_1. This inspires the Boneh-Zhandry [10, 11] definition of security against quantum adversaries, which says that after seeing a (superposition of) signed messages, the adversary cannot produce two signed messages.

However, a true quantum adversary may be able, for some schemes, to set things up so that it can modify a (superposition) of values into one of many possibilities, but still only be able to ever produce a single value. For example, it may be that an adversary sees a superposition of signed messages that always begin with 0, but somehow modifies the superposition to obtain a signed message that begins with a 1. This limitation for signatures was observed by Garg, Yuen, and Zhandry [24], who then give a much stronger notion to fix this issue[1].

Inspired by the above, we formulate a series Either/Or results for quantum lightning and quantum money. In particular, in Section 4, we show, roughly,

Theorem 1 (informal). *If H is a hash function that is collision resistant against quantum adversaries, then either (1) H is collapsing or (2) it can be used to build quantum lightning without any additional computational assumptions.*[2]

The construction of quantum lightning is inspired by the outline above. One difficulty is that above we needed a perfect distinguisher, whereas a collapsing adversary may only have a non-negligible advantage. To obtain an actual quantum lightning scheme, we need to repeat the scheme in parallel many times to boost the distinguish advantage to essentially perfect. Still, defining verification so that

[1] Garg et al. only actually discuss message authentication codes, but the same idea applies to signatures

[2] Technically, there is a slight gap due to the difference between *non-negligible* and *inverse polynomial*. Essentially what we show is that the theorem holds for fixed values of the security parameter, but whether (1) or (2) happens may vary across different security parameters.

we can prove security is a non-trivial task. Indeed, it is much harder to analyze what sorts of invalid bolts might be accepted by the verification procedure, especially since we know virtually nothing about the types of states the given adversary for collapsing accepts.

For example, in order to base security on collision resistance, we would like to say that if a bolt passes verification, we can measure it and obtain a collision. But then we need that the classical test (namely evaluating $H(x)$) *and* the quantum test (namely, that it is superposition) both succeed *simultaneously*. Unfortunately, these two tests are non-commuting operations, so it is impossible to test both with certainty simultaneously. If we perform the classical test before the quantum test, it could be that the second test perturbs the quantum state so that it is in superposition, but no longer a superposition of pre-images. Similarly, if we perform the quantum test first, it could be that running the classical test collapses the state to a singleton. In this case, measuring two accepting bolts could give us the same pre-image, so we do not get a collision.

Using a careful argument, we show nonetheless how to verify and prove security. The intuition is to only perform a single test, and which test is performed is chosen at random independent of the input. We demonstrate that if a state had a reasonably high probability of passing, then it must have *simultaneously* had a noticeable probability of passing each of the two tests. This is enough to get a collision. Next, we just repeat the scheme many times in parallel; now if a bolt even has a non-negligible chance of passing, one of the components must have a high chance of passing, which in turn gives a collision.

Next, we move on to other Either/Or results. We show that:

Theorem 2 (informal). *Any* non-interactive *commitment scheme that is computationally binding against quantum adversaries is either collapse-binding, or it can be used to build quantum lightning without any additional computational assumptions.*

The above theorem crucially relies on the commitment scheme being non-interactive: the serial number of the bolt is the sender's single message, along with his private quantum state. If the commitment scheme is not collapse-binding, the sender's private state can be verified to be in superposition. If an adversary produces two identical bolts, these bolts can be measured to obtain two openings, violating computational binding. In contrast, in the case of interactive commitments, the bolt should be expanded to the transcript of the interaction between the sender and receiver. Unfortunately, for quantum lightning security, the transcript is generated by an adversary, who can deviate from the honest receiver's protocol. Since the commitment scheme is only binding when the receiver is run honestly, we cannot prove security in this setting.

Instead, we consider the weaker goal of constructing public key quantum money. Here, since the mint produces bolts, the original bolt is honestly generated. The mint then signs the transcript using a standard signature scheme (which can be built from one-way functions, and hence implied by commitments). If the adversary duplicates this banknote, it is duplicating an honest commitment transcript, but the note can be measured to obtain two different openings, breaking computational binding. This gives us the following:

Theorem 3 (informal). *Any interactive commitment scheme that is computationally binding against quantum adversaries is either collapse-binding, or it can be used to build public key quantum money without any additional computational assumptions.*

Finally, we extend these ideas to quantum money and digital signatures:

Theorem 4 (informal). *Any one-time signature scheme that is Boneh-Zhandry secure is either Garg-Yuen-Zhandry secure, or it can be used to build public key quantum money without any additional computational assumptions.*

Given the difficulty of constructing public key quantum money (let alone quantum lightning), the above results suggest that most natural constructions of collision resistant hash functions, including all of those used in practice, are likely already collapsing, with analogous statements for commitment schemes and signatures. If they surprisingly turn out to not meet the stronger quantum notions, then we would immediately obtain a construction of public key quantum money from simple tools.

Notice that using our Either/Or results give a potential route toward proving the security of quantum money/lightning in a way that avoids the barrier discussed above. Consider building quantum money from quantum lightning, and in turn building quantum lightning from a collision-resistant non-collapsing hash function. Recall that a banknote is a bolt, together with the mint's signature on the bolt's serial number. A quantum money adversary either (1) duplicates a bolt to yield two bolts with the same serial number (and hence same signature), or (2) produces a second bolt with a different serial number, as well as a forged signature on that serial number. Notice that (2) is impossible simply by the unforgeability of the mint's signature. Meanwhile, in proving that (1) is impossible, our reduction actually *can* produce arbitrary quantum money states (for this step, we assume the reduction is given the signing key). The key is that the reduction on its own *cannot* produce the same quantum money state twice, but it *can* do so using the adversary's cloning abilities, allowing it to break the underlying hard problem.

Quantum Money From Obfuscation We now consider the task of constructing public key quantum money. One possibility is based on Aaronson and Christiano's broken scheme [3]. In their scheme, a quantum banknote $|\$\rangle$ is a uniform superposition over some subspace S, that is known only to the bank. The quantum Fourier transform of such a state is the uniform superposition over the dual subspace $S^\perp$. This gives a simple way to check the banknote: test if $|\$\rangle$ lies in S, and whether it's Fourier transform lies in $S^\perp$. Aaronson and Christiano show that the only state which can pass verification is $|\$\rangle$.

To make this scheme public key, one gives out a mechanism to test for membership in S and $S^\perp$, without actually revealing $S, S^\perp$. This essentially means obfuscating the functions that decide membership. Aaronson and Christiano's scheme can be seen as a candidate obfuscator for subspaces. While unfortunately their obfuscator has since been broken, one may hope to instantiate their scheme using recent advances in general-purpose program obfuscation, specifically indistinguishability obfuscation (iO) [7, 23].

On the positive side, Aaronson and Christiano show that their scheme is secure if the subspaces are provided as quantum-accessible black boxes, giving hope that some obfuscation of the subspaces will work. Unfortunately, proving security relative to iO appears a difficult task. One limitation is the barrier discussed above, that any reduction must be able to produce a valid banknote, which means it can also produce two banknotes. Yet at the same time, it somehow has to use the adversary's forgery (a second banknote) to break the iO scheme. Note that this situation is different from the quantum lightning setting, where there were many valid states, and no process could generate the same state twice. Here, there is a single valid state (the state $|\$\rangle$), and it would appear the reduction must be able to construct this precise state exactly once, but not twice. Such a reduction would clearly be impossible. As discussed above Aaronson and Christiano circumvent this issue by using a non-standard type of assumption; their technique is not relevant for standard definitions of iO.

In Section 5, we prove the security of Aaronson and Christiano's scheme using iO. Our solution is to separate the proof into two phases. In the first, we change the spaces obfuscated from $S, S^\perp$ to T_0, T_1, where T_0 is a random unknown subspace containing S, and T_1 is a unknown random subspace containing $S^\perp$. This modification can be proved undetectable using a weak form of obfuscation we define, called subspace-hiding obfuscation, which in turn is implied by iO. Note that in this step, we even allow the reduction to know S (but not T_0, T_1), so it can produce as many copies of $|\$\rangle$ as it would like to feed to the adversary. The reduction does not care about the adversary's forgery directly, only whether or not the adversary successfully forges. If the adversary forges when given obfuscations of $S, S^\perp$, it must also forge under T_0, T_1, else it can distinguish the two cases and hence break the obfuscation. By using the adversary in this way, we avoid the apparent difficulties above.

In the next step, we notice that, conditioned on T_0, T_1, the space S is a random subspace between $T_1^\perp$ and T_0. Thus conditioned on T_0, T_1, the adversary clones a state $|\$\rangle$ defined by a random subspace S between $T_1^\perp$ and T_0. The number of possible S is much larger than the dimension of the state $|\$\rangle$, so in particular the states cannot be orthogonal. Thus, by no-cloning, duplication is impossible. We need to be careful however, since we want to rule out adversaries that forge with even very low success probabilities. To do so, we need to precisely quantify the no-cloning theorem, which we do. We believe our new no-cloning theorem may be of independent interest. We note that when applying no-cloning, we do not rely on the secrecy of T_0, T_1, but only that S is hidden. Intuitively, there are exponentially many more S's between T_0, T_1 than the dimension of the space $|\$\rangle$ belongs to, so no-cloning implies that a forger has negligible success probability. Thus we reach a contradiction, showing that the original adversary could not exist.

We also show how to view Aaronson and Christiano's scheme as a signature scheme; we show that the signature scheme satisfies the Boneh-Zhandry definition, but not the strong Garg-Yuen-Zhandry notion. Thus, we can view Aaronson and Christiano's scheme as an instance of our Either/Or results, and moreover provide the first separation between the two security notions for signatures.

We note that our result potentially relies on a much weaker notion of obfuscation that full iO, giving hope that security can be based on weaker assumptions. For example, an intriguing open question is whether or not recent constructions of obfuscation for certain evasive functions [36, 27] based on LWE can be used to instantiate our notion of subspace hiding obfuscation. This gives another route toward building quantum money from hard lattice problems. This is particularly important at the present time, where the security of iO in the quantum setting is somewhat uncertain (see below for a discussion).

Constructing Quantum Lightning In Section 6, we finally turn to actually building quantum lightning, and hence giving another route to quantum money. Following our Either/Or results, we would like a *non-collapsing* collision-resistant hash function. Unfortunately, Unruh's counterexample does not yield an explicit construction. Instead, he builds on techniques of [5] to give a hash function relative to a *quantum* oracle[3]. As it is currently unknown how to obfuscate quantum oracles with a meaningful notion of security, this does not give even a candidate construction of quantum lightning. Instead, we focus on specific standard-model constructions of hash functions. Finding suitable hash functions is surprisingly challenging; we were only able to find a single family of candidates, and leave finding additional candidates as a challenging open problem.

To motivate our construction, we consider the following approach to building quantum lightning from the short integer solution (SIS) problem. In SIS, an underdetermined system of homogeneous linear equations is given, specified by a wide matrix $\mathbf{A}$, and the goal is to find a solution consisting of "small" entries; that is, a "short" vector $\mathbf{x}$ such that $\mathbf{A}.\mathbf{x} = 0$. For random linear constraints, SIS is conjectured to be computationally difficult, which is backed up by reductions from the hardness of worst-case lattice problems [30]. SIS gives a simple collision resistant hash function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$, where the domain is constrained to be small; given a collision $\mathbf{x}, \mathbf{x}'$, one obtains a SIS solution as $\mathbf{x} - \mathbf{x}'$.

One may hope that SIS is also non-collapsing, in which case we would obtain quantum lightning. One (failed) attempt to obtaining a collapsing distinguisher is the following. Start with superposition of "short" vectors $\mathbf{x}$, weighted by a Gaussian function. When $f_{\mathbf{A}}$ is applied, the superposition collapses to a superposition over short vectors $\mathbf{x}$ that all have the same value of $\mathbf{A} \cdot \mathbf{x}$. This will be a bolt in the scheme, and the serial number will be the common hash. To verify the bolt, we first check the hash. Then, to verify the bolt is in superposition, we apply the

[3] that is, the oracle itself performs quantum operations

quantum Fourier transform. Note that if $\mathbf{x}$ were a uniform superposition over *all* vectors, the QFT would give a uniform superposition over all vectors in the row-span of $\mathbf{A}$ (with some phase terms). Instead, since $\mathbf{x}$ is a superposition over "short" vectors, using the rules of Fourier transforms is possible to show that the QFT gives a superposition over vectors of the form $\mathbf{r} \cdot \mathbf{A} + \mathbf{e}$, where $\mathbf{r}$ is a random row vector, and $\mathbf{e}$ is a Gaussian-weighted random short row vector.

Intuitively, we just need to distinguish these types of vectors from random vectors. Unfortunately, distinguishing $\mathbf{r} \cdot \mathbf{A} + \mathbf{e}$ from random for a random matrix $\mathbf{A}$ is an instance of the Learning With Errors (LWE) problem, which is widely believed to be comptuationally intractable, as evidenced by quantum reductions from worst-case lattice problems [33].

We therefore need to "break" LWE by given some trapdoor information. The usual way to break LWE is to provide a short vector $\mathbf{t}$ in the kernel of $\mathbf{A}$. Then, to distinguish an input $\mathbf{u}$, simply compute $\mathbf{u} \cdot \mathbf{t}$, and check if the result is small. In the case $\mathbf{u} = \mathbf{r} \cdot \mathbf{A} + \mathbf{e}$, then $\mathbf{u} \cdot \mathbf{t} = \mathbf{e} \cdot \mathbf{t}$, which will be small. In contrast, if $\mathbf{u}$ is random, $\mathbf{t} \cdot \mathbf{u}$ will be large with overwhelming probability.

Unfortunately, the trapdoor $\mathbf{t}$ is a SIS solution! In particular, in order for the distinguisher to work, one can show that $\mathbf{t}$ needs to be somewhat smaller than the size bound on the domain of $f_{\mathbf{A}}$. With such a trapdoor, it is therefore easy to manufacture collisions for $f_{\mathbf{A}}$, so $f_{\mathbf{A}}$ is no longer collision-resistant. Worse yet, it is straightforward to use the trapdoor to come up with a superposition of inputs that fools the distinguisher.

We do not know how to make the above approach work, as all ways we are aware of for breaking LWE involve handing out a SIS solution. One possible approach would be to obfuscate an LWE distinguisher that has the trapdoor hardcoded. This allows for distinguishing LWE samples without *explicitly* handing out a SIS solution. However, it might be possible to construct a SIS solution from any such distinguishing program.

We now turn to our actual construction. Our idea is to use linear equations over restricted domains as in SIS, but will restrict the domain in different ways. In particular, we will view vectors as specifying symmetric matrices (that is, an $(n+1)n/2$-dimensional vector will correspond to an $n \times n$ symmetric matrix, with the vector entries specifying the upper-triangular part of the matrix). Instead of restricting the size of entries, will will instead restrict the *rank* of the symmetric matrix. Our construction then follows the rough outline of the SIS-based approach above, intuitively using rank as a stand-in for vector norm. By switching from vector norm to matrix rank, we are able to arrive at a construction whose security follows from a plausible computational assumption.

A bolt is then a superposition over rank-bounded matrices satisfying the linear constraints. Analogous to the SIS approach, we are able to show that applying the Quantum Fourier transform on such bolts results in a state whose support consists of matrices $\mathbf{A}$ that can be written as $\mathbf{A} = \mathbf{B} + \mathbf{C}$, where $\mathbf{B}$ is a sum of a few known matrices (based on the precise linear functions), whereas $\mathbf{C}$ is an arbitrary low-rank matrix. We show how to generate the constraints along with a public "trapdoor" which allows for such matrices can be identified.

Our trapdoor is simply a row rank matrix in the kernel of the linear constraints, analogous to how the LWE trapdoor is a short vector in the kernel.

One may be rightfully concerned at this point, as our trapdoor has the same form as domain elements for our hash function. Indeed, if the rank of the trapdoor was smaller than the rank of the domain, the trapdoor would completely break the construction. Importantly for our construction, we show that this matrix can have higher rank than the allowed inputs to the hash function; as such, it does not appear useful for generating collisions.

Our scheme can easily be proved secure under the assumed collision-resistance of our hash function. Unfortunately, this assumption is false. Indeed, the family of matrices $\mathbf{B}^T\mathbf{B}$ for wide and short matrices $\mathbf{A}$ is low rank. By evaluating our hash function on such matrices, we turn it into a degree-2 polynomial over the $\mathbf{B}$ matrices. Unfortunately, Ding and Yang [20] and Applebaum et al. [6] show that such hash functions are not collision resistant[4].

However, we will apply a simple trick in order to get our scheme to work. Namely, we show how to use the attacks above to actually generate *superpositions* over k colliding inputs for some parameter k that depends on the various parameters of the scheme. At the same time, the attacks do not seem capable of generating collisions beyond k. We will therefore set our bolt to be this superposition over several colliding inputs. Now, we can apply our testing procedure to each of the inputs separately to verify the bolt. If an adversary creates two bolts with the same serial number, we can measure to obtain $2k$ colliding inputs. By assuming the plausible $2k$*-multi-collision resistance* of our hash functions, we obtain security.

Our construction requires a common reference string, namely the sequence of linear constraints and the trapdoor. We show that we can convert our scheme into the common *random* string (crs) model by using the common random string to generate the trapdoor and linear constraints.

1.2 Related Works

Quantum Money. Lutomirski [28] shows another weakness of Wiesner's scheme: a merchant, who is allowed to interact with the mint for verification, can use the verification oracle to break the scheme and forge new currency. Public key quantum money is necessarily secure against adversaries with a verification oracle, since the adversary can implement the verification oracle for itself. Several alternative solutions to the limitations of Wiesner's scheme have been proposed [31, 25], though the "ideal" solution still remains public key quantum money.

Randomness Expansion and Certifiable Randomness. Colbeck [16] proposed the idea of a classical experimenter, interacting with several potentially untrustworthy quantum devices, can expand a small random seed into a certifiably random longer

[4] Technically, they only show this is true if the degree-2 polynomials are random, whereas ours are more structured, but we show that their analysis extends to our setting as well

seed. Subsequent to our work, Brakerski et al. [12] consider certifiable randomness in the computational setting. Of of these results are related, but entirely different from, our version of verifiable randomness. In particular, their protocols are interactive and privately verifiable, but allows for a classical verifier. In contrast, our protocol is non-interactive (in the crs model) and publicly verifiable, but requires a quantum verifier.

Obfuscation and Multilinear Maps. There is a vast body of literature on strong notions of obfuscation, starting with the definitional work of Barak et al. [7]. Garg et al. [23] propose the first obfuscator plausibly meeting the strong notion of iO, based on cryptographic multilinear maps [22, 17, 26]. Unfortunately, there have been numerous attacks on multilinear maps, which we do not fully elaborate on here. There have been several quantum attacks [18, 4, 15, 14] on obfuscators, but there are still schemes that remain unbroken. Moreover, there has been some success in transforming applications of obfuscation to be secure under assumptions on lattices [13, 36, 27], which are widely believed to be quantum hard. We therefore think it plausible that subspace-hiding obfuscation, which is all we need for this work, can be based on similar lattice problems. Nonetheless, obfuscation is a very active area of research, and we believe that one of the current obfuscators so some future variant will likely be secure quantum resistant.

Computational No-cloning. We note that computational assumptions and no-cloning have been combined in other contexts, such as Unruh's revocable time-released encryption [34]. We note however, that these settings do not involve verification, the central theme of this work.

2 Preliminaries

Throughout this paper, we will let λ be a security parameter. When inputted into an algorithm, λ will be represented in unary. A function $\epsilon(\lambda)$ is *negligible* if for any inverse polynomial $1/p(\lambda)$, $\epsilon(\lambda) < 1/p(\lambda)$ for sufficiently large λ. A function is *non-negligible* if it is not negligible, that is there exists an inverse polynomial $1/p(\lambda)$ such that $\epsilon(\lambda) \geq 1/p(\lambda)$ infinitely often.

2.1 Quantum Computation

Here, we very briefly recall some basics of quantum computation. A quantum system Q is defined over a finite set B of classical states. A **pure** state over Q is an L_2-normalized vector in $\mathbb{C}^{|B|}$, which assigns a (complex) weight to each element in B. We will think of pure states as column vectors. The pure state that assigns weight 1 to x and weight 0 to each $y \neq x$ is denoted $|x\rangle$.

A pure state $|\phi\rangle$ can be manipulated by performing a unitary transformation U to the state $|\phi\rangle$. We will denote the resulting state as $|\phi'\rangle = U|\phi\rangle$. A unitary is *quantum polynomial time* (QPT) if it can be represented as a polynomial-sized circuit of gates from a finite gate set. $|\phi\rangle$ can also be measured; the measurement outputs the value x with probability $|\langle x|\phi\rangle|^2$. The normalization of $|\phi\rangle$ ensures that the distribution over x is indeed a probability distribution. After measurement, the state "collapses" to the state $|x\rangle$. Notice that subsequent measurements will always output x, and the state will always stay $|x\rangle$.

We define the Euclidean distance $\||\phi\rangle - |\psi\rangle\|$ between two states as the value $\left(\sum_x |\alpha_x - \beta_x|^2\right)^{\frac{1}{2}}$ where $|\phi\rangle = \sum_x \alpha_x|x\rangle$ and $|\psi\rangle = \sum_x \beta_x|x\rangle$.

We will be using the following lemma:

Lemma 1 ([8]). *Let $|\varphi\rangle$ and $|\psi\rangle$ be quantum states with Euclidean distance at most ϵ. Then, performing the same measurement on $|\varphi\rangle$ and $|\psi\rangle$ yields distributions with statistical distance at most 4ϵ.*

2.2 Public Key Quantum Money

Here, we define public key quantum money. We will slightly modify the usual definition [2], though the definition will be equivalent under simple transformations.

- We only will consider what Aaronson and Christiano [3] call a quantum money *mini-scheme*, where there is just a single valid banknote. It is straightforward to extend to general quantum money using a signatures
- We will change the syntax to more closely resemble our eventual quantum lightning definition, in order to clearly compare the two objects.

Quantum money consists of two quantum polynomial time algorithms $\mathsf{Gen}, \mathsf{Ver}$.

- Gen takes as input the security parameter, and samples a banknote $|\$\rangle$
- Ver verifies a banknote, and if the verification is successful, produces a serial number for the note.

For correctness, we require that verification always accepts money produced by Gen. We also require that verification does not perturb the money. Finally, since Ver is a quantum algorithm, we must ensure that multiple runs of Ver on the same money will always produce the same serial number. This is captured by the following two of requirements:

- For a money state $|\$\rangle$, let $H_\infty(|\$\rangle) = -\log_2 \min_s \Pr[\mathsf{Ver}(|\$\rangle) = s]$ be the min-entropy of s produced by applying Ver to $|\$\rangle$, were we do not count the rejecting output $\perp$ as contributing to the min-entropy. We insist that $\mathbb{E}[H_\infty(|\$\rangle)]$ is negligible, the expectation over $|\$\rangle \leftarrow \mathsf{Gen}(1^\lambda)$. This ensures the serial number is essentially a deterministic function of the money.
- For a money state $|\$\rangle$, let $|\psi\rangle$ be the state left over after running $\mathsf{Ver}(|\$\rangle)$. We insist that $\mathbb{E}[|\langle\psi|\$\rangle|^2] \geq 1 - \mathsf{negl}(\lambda)$, where the expectation is over $|\$\rangle \leftarrow \mathsf{Gen}(1^\lambda)$, and any affect Ver has on $|\psi\rangle$. This ensures that verification does not perturb the money.

For security, consider the following game between an adversary A and a challenger

- The challenger runs $\mathsf{Gen}(1^\lambda)$ to get a banknote $|\$\rangle$. It runs Ver on the banknote to extract a serial number s.

- The challenger sends $|\$\rangle$ to A.
- A produces two candidate quantum money states $|\$_0\rangle, |\$_1\rangle$, which are potentially entangled.
- The challenger runs Ver on both states, to get two serial numbers s_0, s_1.
- The challenger accepts if and only if both runs of Ver pass, and the serial numbers satisfy $s_0 = s_1 = s$.

Definition 1. *A quantum money scheme* $(\mathsf{Gen}, \mathsf{Ver})$ *is secure if, for all QPT adversaries A, the probability the challenger accepts in the above experiment is negligible.*

3 Quantum Lightning

3.1 Definitions

The central object in a quantum lightning system is a lightning bolt, a quantum state that we will denote as $|↯\rangle$. Bolts are produced by a storm, 🌧, a polynomial time quantum algorithm which takes as input a security parameter λ and samples new bolts. Additionally, there is a quantum polynomial-time bolt verification procedure, Ver, which serves two purposes. First, it verifies that a supposed bolt is actually a valid bolt; if not it rejects and outputs $\bot$. Second, if the bolt is valid, it extracts a fingerprint/serial number of the bolt, denoted s.

Rather than having a single storm 🌧 and single verifier Ver, we will actually have a family $\mathcal{F}_\lambda$ of (🌧, Ver) pairs for each security parameter. We will have a setup procedure $\mathsf{SetupQL}(1^\lambda)$ which samples a (🌧, Ver) pair from some distribution over $\mathcal{F}_\lambda$.

For correctness, we have essentially the same requirements as quantum money. We require that verification always accepts bolts produced by 🌧. We also require that verification does not perturb the bolt. Finally, since Ver is a quantum algorithm, we must ensure that multiple runs of Ver on the same bolt will always produce the same fingerprint. This is captured by the following two of requirements:

- For a bolt $|↯\rangle$, let

  $$H_\infty(|↯\rangle, \mathsf{Ver}) = -\log_2 \min_s \Pr[\mathsf{Ver}(|↯\rangle) = s]$$

  be the min-entropy of s produced by applying Ver to $|↯\rangle$, were we do not count the rejecting output $\bot$ as contributing to the min-entropy. We insist that $\mathbb{E}[H_\infty(|↯\rangle, \mathsf{Ver})]$ is negligible, where the expectation is over (🌧, Ver) $\leftarrow$ $\mathsf{SetupQL}(\lambda)$ and $|↯\rangle \leftarrow$ 🌧. This ensures the serial number is essentially a deterministic function of the bolt.
- For a bolt $|↯\rangle$, let $|\psi\rangle$ be the state left over after running $\mathsf{Ver}(|↯\rangle)$. We insist that $\mathbb{E}[|\langle\psi|↯\rangle|^2] \geq 1 - \mathsf{negl}(\lambda)$, where the expectation is over (🌧, Ver) $\leftarrow$ $\mathsf{SetupQL}(\lambda)$, $|↯\rangle \leftarrow$ 🌧, and any affect Ver has on $|\psi\rangle$. This ensures that verification does not perturb the bolt.

Remark 1. We note that it suffices to only consider the first requirement, since the serial number is essentially a deterministic function of the bolt. Indeed, by un-computing the Ver computation after obtaining the serial number, a straightforward calculation shows the result will be negligibly close to the original state.

For security, informally, we ask that no adversarial storm ⛈ can produce two bolts with the same serial number. More precisely, consider the following experiment between a challenger and a malicious bolt generation procedure ⛈:

- The challenger runs (🌧, Ver) $\leftarrow \mathsf{SetupQL}(1^\lambda)$, and sends (🌧, Ver) to ⛈.
- ⛈ produces two (potentially entangled) quantum states $|↯_0\rangle, |↯_1\rangle$, which it sends to the challenger
- The challenger runs Ver on each state, obtaining two fingerprints s_0, s_1. The challenger accepts if and only if $s_0 = s_1 \neq \perp$.

Definition 2. *A quantum lightning scheme has* uniqueness *if, for all QPT adversarial storms ⛈, the probability the challenger accepts in the game above is negligible in λ.*

4 Either/Or Results

4.1 Infinity-Often Security

Before describing our Either/Or results, we need to introduce a non-standard notion of security. Typically, a security statement says that no polynomial-time adversary can win some game, except with negligible probability. A violation of the security statement is a polynomial-time adversary that can win with *non*-negligible probability; that is, some probability ϵ that is lower bounded by an inverse-polynomial *infinitely often.* In our proofs below, we use such an adversary to devise a scheme for another problem. But to actually get an efficient scheme, we need the adversary's success probability to actually be inverse-polynomial, not non-negligible. This motivates the notion of *infinitely often security.* A scheme has infinitely-often security if security holds for an infinite number of security parameters, but not necessarily all security parameters. It is straightforward to modify all security notions in this work to infinitely-often variants.

4.2 Collision Resistant Hashing

A hash function is a function H that maps large inputs to small inputs. We will considered keyed functions, meaning it takes two inputs: a key $k \in \{0,1\}^\lambda$, and the actual input to be compressed, $x \in \{0,1\}^{m(\lambda)}$. The output of H is $n(\lambda)$ bits. For the hash function to be useful, we will require $m(\lambda) \gg n(\lambda)$.

The usual security property for a hash function is collision resistance, meaning it is computationally infeasible to find two inputs that map to the same output.

Definition 3. *H is collision resistant if, for any QPT adversary A,* $\Pr[H(x_0) = H(x_1) \wedge x_0 \neq x_1 : (x_0, x_1) \leftarrow A(k), k \leftarrow \{0,1\}^\lambda] < \mathsf{negl}(\lambda)$

Unruh [35] points out weaknesses in the usual collision resistance definition, and instead defines a stronger notion called *collapsing*. Intuitively, it is easy for an adversary to obtain a superposition of pre-images of some output, by running H on a uniform superposition and then measuring the output. Collapsing requires, however, that this state is computationally indisitnguishable from a random input x. More precisely, for an adversary A, consider the following experiment between A and a challenger

- The challenger has an input bit b.
- The challenger chooses a random key k, which it gives to A.
- A creates a superposition $|\psi\rangle = \sum_x \alpha_x |x\rangle$ of elements in $\{0,1\}^{m(\lambda)}$.
- In superposition, the challenger evaluates $H(k, \cdot)$ to get the state $|\psi'\rangle = \sum_x \alpha_x |x, H(k,x)\rangle$
- Then, the challenger either:
  - If $b = 0$, measures the $H(k,x)$ register, to get a string y. The state $|\psi'\rangle$ collapses to $|\psi_y\rangle \propto \sum_{x:H(k,x)=y} \alpha_x |x, y\rangle$
  - If $b = 1$, measures the entire state, to get a string $x, H(k,x)$. The state $|\psi'\rangle$ collapses to $|x, H(k,x)\rangle$
- The challenger returns whatever state remains of $|\psi'\rangle$ (namely $|\psi_y\rangle$ or $|x, H(k,x)\rangle$) to A.
- A outputs a guess b' for b. Define $\texttt{Collapse-Exp}_b(A, \lambda)$ as b'.

Definition 4. *H is collapsing if, for all QPT adversaries A,*
$|\Pr[\texttt{Collapse-Exp}_0(A,\lambda) = 1] - \Pr[\texttt{Collapse-Exp}_1(A,\lambda) = 1]| < \mathsf{negl}(\lambda)$

Theorem 5. *Suppose H is collision resistant. Then both of the following are true:*

- *Either H is collapsing, or H can be used to build a quantum lightning scheme that is infinitely often secure.*
- *Either H is infinitely often collapsing, or H can be used to build a quantum lightning scheme that is secure.*

Proof. Let A be a collapsing adversary; the only difference between the two cases above are whether A's advantage is non-negligible or actually inverse polynomial. The two cases are nearly identical, but the inverse polynomial case will simplify notation. We therefore assume that H is not infinitely-often collapsing, and will design a quantum lightning scheme that is secure.

Let A_0 be the first phase of A: it receives a hash key k as input, and produces a superposition of pre-images, as well as it's own internal state. Let A_1 be the second phase of A: it receives the internal state from A_0, plus the superposition of input/output pairs returned by the challenger. It outputs 0 or 1.

Define $q_b(\lambda) = \Pr[\texttt{Collapse-Exp}_b(A,\lambda) = 1]$. By assumption, we have that $|q_0(\lambda) - q_1(\lambda)| \geq 1/p(\lambda)$ for some polynomial p. We will assume $q_0 < q_1$, the other case handled analogously.

For an integer r, consider the function $H^{\otimes r}(k, \cdot)$ which takes as input a string $x \in (\{0,1\}^{m(\lambda)})^r$, and outputs the vector $(H(k, x_1), \ldots, H(k, x_r))$. The collision resistance of H easily implies the collision resistance of $H^{\otimes r}$, for any polynomial r. Moreover, we will use A to derive a collapsing adversary $A^{\otimes r}$ for $H^{\otimes r}$ which has near-perfect distinguishing advantage. $A^{\otimes r}$ works as follows.

- First, it runs A_0 in parallel r times to get r independent states $|\psi_i\rangle$, where each $|\psi_i\rangle$ contains a superposition of internal state values, as well as inputs to the hash function.
- It assembles the r superpositions of inputs into a superposition of inputs for $H^{\otimes r}$, which it then sends to the challenger.
- The challenger responds with a potential superposition over input/output pairs (through the output value in $(\{0,1\}^{n(\lambda)})^r$ is fixed).
- $A^{\otimes r}$ disassembles the input/output pairs into r input/output pairs for H.
- It then runs A_1 in parallel r times, on each of the corresponding state/input/output superpositions, to get bits $b'_1, \ldots, b'_r$.
- $A^{\otimes r}$ then computes $f = (\sum_i b'_i)/r$, the fraction of b'_i that are 1.
- If $f > (q_0 + q_1)/2$ (in other words, f is closer to q_1 than it is to q_0), A outputs 1; otherwise it outputs 0.

Notice that if $A^{\otimes r}$'s challenger uses $b = 0$ (so it only measures the output registers), this corresponds to each A seeing a challenger with $b = 0$. In this case, each b'_i with be 1 with probability q_0. This means that f will be a (normalized) Binomial distribution with expected value q_0. Analogously, if $b = 1$, each b'_i will be 1 with probability q_1, so f will be a normalized Binomial distribution with expected value q_1. Since $q_1 - q_0 \geq 1/p(\lambda)$, we can use Hoeffding's inequality to choose r large enough so that in the $b = 0$ case, $f < (q_0 + q_1)/2 = q_0 + 1/2p(\lambda)$ except with probability $2^{-\lambda}$. Similarly, in the $b = 1$ case, $f > (q_0 + q_1)/2 = q_1 - 1/2p(\lambda)$ except with probably $2^{-\lambda}$. This means $A^{\otimes r}$ outputs the correct answer except with probability $2^{-\lambda}$.

We now describe a first attempt at a quantum lightning scheme:

- $\mathsf{SetupQL}_0$ simply samples and outputs a random hash key k. This key will determine 🌩$_0$, Ver_0 as defined below.
- 🌩$_0$ runs $A_0^{\otimes r}(k)$, where r is as chosen above and $A_0^{\otimes r}$ represents the first phase of $A^{\otimes r}$.
  When $A_0^{\otimes r}$ produces a superposition $|\psi\rangle$ over inputs $x \in \{0,1\}^{rm}$ for $H^{\otimes r}(k, \cdot)$ as well as some private state, 🌩$_0$ applies $H^{\otimes r}$ in superposition, and measures the result to get $y \in \{0,1\}^{rn}$.
  Finally, 🌩$_0$ outputs the resulting state $|\lightning\rangle = |\psi_y\rangle$.
- Ver_0 on input a supposed bolt $|\lightning\rangle$, first applies $H^{\otimes r}(k, \cdot)$ in superposition to the input registers to obtain y, which it measures. It saves y, which will be the serial number for the bolt.
  Next, consider two possible tests Test_0 and Test_1. In Test_0, run $A_1^{\otimes r}$ — the second phase of $A^{\otimes r}$ — on the $|\lightning\rangle$ and measure the result. If the result is 1 (meaning $A^{\otimes r}$ guesses that the challenger measured the entire input/output registers), then abort and reject. Otherwise if the result is 0 (meaning $A^{\otimes r}$

guess that the challenger only measured the output), then it un-computes $A_1^{\otimes r}$. Note that since we measured the output of $A_1^{\otimes r}$, un-computing does not necessarily return the bolt to its original state.

Test_1 is similar to Test_0, except that the input registers x are measured before running $A_1^{\otimes r}$. This measurement is not a true measurement, but is instead performed by copying x into some private registers. Moreover, the abort condition is flipped: if the result of applying $A_1^{\otimes r}$ is 0, then abort and reject. Otherwise un-compute $A_1^{\otimes r}$, and similarly "un-measure" x by un-computing x from the private registers.

Ver_0 chooses a random c, and applies Test_c. If the test accepts, then it outputs the serial number y, indicated that it accepts the bolt.

Correctness. For a valid bolt, Test_0 corresponds to the $b = 0$ challenger, in which case we know $A_1^{\otimes r}$ outputs 0 with near certainty. This means Ver continues, and when it un-computes, the result will be negligibly close to the original bolt. Similarly, Test_1 corresponds to the $b = 1$ challenger, in which case $A_1^{\otimes r}$ outputs 1 with near certainty. Un-computing returns the bolt to (negligibly close to) its original state. For a valid bolt, the serial number is always the same. Thus, 🌧, Ver satisfy the necessary correctness requirements.

Security. Security is more tricky. Suppose instead of applying a random Test_c, Ver_0 applied both tests. The intuition is that if Ver accepts, it means that the two possible runs of $A_1^{\otimes r}$ would output different results, which in turn means that $A_1^{\otimes r}$ detected whether or not the input registers were measured. For such detection to even be possible, it must be the case that the input registers are in superposition. Then suppose an adversarial storm ⛈ generates two bolts $|\lightning_0\rangle, |\lightning_1\rangle$ that are potentially entangled such that both pass verification with the same serial number. Then we can measure both states, and the result will (with reasonable probability) be two distinct pre-images of the same y, representing a collision. By the assumed collision-resistance of H (and hence $H^{\otimes r}$), this will means a contradiction.

The problem with the above informal argument is that we do not know how $A_1^{\otimes r}$ will behave on non-valid bolts that did not come from $A_0^{\otimes r}$. In particular, maybe it passes verification with some small, but non-negligible success probability. It could be that after passing Test_0, the superposition has changed significantly, and maybe is no longer a superposition over pre-images of y, but instead a single pre-image. Nonetheless, if the auxiliary state registers are not those generated by $A_0^{\otimes r}$, it may be that the second test still accepts — for example, it may be that if $A^{\otimes r}$'s private state contains a particular string, it will always accept; normally this string would not be present, but the bolt that remains after performing one of Test_c may contain this string. We have to be careful to show that this case cannot happen, or if it does there is still nonetheless a way to extract a collision.

Toward that end, we only choose a single test at random. We will first show a weaker form of security, namely that an adversary cannot produce two bolts that are both accepted with probability close to 1 and have the same serial number.

Then we will show how to modify the scheme so that it is impossible to produce bolts that are even accepted with small probability.

Consider a bolt where, after measuring $H(k, \cdot)$, the inputs registers are *not* in superposition at all. In this case, the measurement in Test_1 is redundant, and we therefore know that both runs of Test_c are the same, except the acceptance conditions are flipped. Since the choice of test is random, this means that such a bolt can only pass verification with probability at most $1/2$.

More generally, suppose the bolt was in superposition, but most of the weight was on a single input x_0. Precisely, suppose that when measuring the x registers, x_0 is obtained with probability $1 - \alpha$ for some relatively small α. We prove:

Claim. Consider a quantum state $|\phi\rangle$ and a projective partial measurement on some of the registers. Let $|\phi_x\rangle$ be the state left after performing the measurement and obtaining x. Suppose that some outcome of the measurement x_0 occurs with probability $1 - \alpha$. Then $\||\phi_{x_0}\rangle - |\phi\rangle\| < \sqrt{2\alpha}$

Proof. First, the $|\phi_x\rangle$ are all orthogonal since the measurement was projective. Let $\Pr[x]$ be the probability that the partial measurement obtains x. It is straightforward to show that $|\phi\rangle = \sum_y \sqrt{\Pr[x]}\beta_x|\phi_x\rangle$ for some β_x of unit norm. The overall phase can be taken to be arbitrary, so we can set $\beta_{x_0} = 1$. Then we have $\langle\phi_{x_0}|\phi\rangle = \sqrt{1-\alpha}$. This means $\||\phi_{x_0}\rangle - |\phi\rangle\|^2 = 2 - 2(\langle\phi_{x_0}|\phi\rangle) = 2 - 2\sqrt{1-\alpha} \leq 2\alpha$ for $\alpha \in [0, 1]$. □

Now, suppose for the bolt that Test_0 passes with probability t. Suppose $\alpha \leq 1/200$. Then Test_1 can only pass with probability at most $3/2 - t$. This is because with probability at least $199/200$, the measurement in Test_1 yields x_0. Applying Claim 4.2, the result in this case is at most a distance $\sqrt{2/200} = \frac{1}{10}$ from the original bolt. In this case, since the acceptance criteria for Test_1 is the opposite of Test_0, the probability Test_1 passes is at most $1 - t + \frac{4}{10}$ by Lemma 1. Over all then, Test_1 passes with probability at most $(199/200)\left(1 - t + \frac{4}{10}\right) + (1/200) \leq \frac{3}{2} - t$.

Therefore, since the test is chosen at random, the probability of passing the test is the average of the two cases, which is at most $\frac{3}{4}$ regardless of t. Therefore, for any candidate pair of bolts $|\text{ϟ}_0\rangle|\text{ϟ}_1\rangle$, either:

(1) If the bolts are measured, two different pre-images of the same y, and hence a collision for $H^{\otimes r}$, will be obtained with probability at least $1/200$

(2) The probability that both bolts accept and have the same serial number is at most $\frac{3}{4}$.

Notice that if $|\text{ϟ}_0\rangle, |\text{ϟ}_1\rangle$ are produced by an adversarial storm ⛈, then event (1) can only happen with negligible probability, else we obtain a collision-finding adversary. Therefore, we have that for any efficient ⛈, except with negligible probability, the probability that both bolts produced by ⛈ accept and have the same serial number is at most $\frac{3}{4}$.

In the full scheme, a bolt is simply a tuple of λ bolts produced by 🌧$_0$, and the serial number is the concatenation of the serial numbers from each constituent bolt. The above analysis show that for any efficient adversarial storm ⛈ that produces two bolt sequences $|\lightning_b\rangle = (|\lightning_{b,1}\rangle, \ldots, |\lightning_{b,\lambda}\rangle)$, the probability that both sequences completely accept and agree on the serial numbers is, except with negligible probability, at most $\left(\frac{3}{4}\right)^\lambda$, which is negligible. Thus we obtain a valid quantum lightning scheme. □

5 Quantum Money from Obfuscation

In this section, we show that, assuming injective one-way functions exist, applying indisitnguishability obfuscation to Aaronson and Christiano's abstract scheme [3] yields a secure quantum money scheme.

5.1 Obfuscation

Definition 5. *A* subspace hiding obfuscator *(shO) for a field $\mathbb{F}$ and dimensions d_0, d_1 is a PPT algorithm shO such that:*

- **Input.** shO *takes as input the description of a linear subspace $S \subseteq \mathbb{F}^n$ of dimension $d \in \{d_0, d_1\}$. For concreteness, we will assume S is given as a matrix whose rows form a basis for S.*
- **Output.** shO *outputs a circuit $\hat{S}$ that computes membership in S. Precisely, let $S(x)$ be the function that decides membership in S. Then*

$$\Pr[\hat{S}(x) = S(x) \forall x : \hat{S} \leftarrow \mathsf{shO}(S)] \geq 1 - \mathsf{negl}(n)$$

- **Security.** *For security, consider the following game between an adversary and a challenger, indexed by a bit b.*
  - *The adversary submits to the challenger a subspace S_0 of dimension d_0*
  - *The challenger chooses a random subspace $S_1 \subseteq \mathbb{F}^n$ of dimension d_1 such that $S_0 \subseteq S_1$. It then runs $\hat{S} \leftarrow \mathsf{shO}(S_b)$, and gives $\hat{S}$ to the adversary*
  - *The adversary makes a guess b' for b.*

  The adversary's advantage is the the probability $b' = b$, minus $1/2$. shO is secure if, all PPT adversaries have negligible advantage.

In the full version [38], we show the following theorem, which demonstrates that *indistinguishability obfuscation* can be used to build subspace-hiding obfuscation:

Theorem 6. *If injective one-way functions exist, then any indistinguishability obfuscator, appropriately padded, is also a subspace hiding obfuscator for field $\mathbb{F}$ and dimensions d_0, d_1, as long as $|\mathbb{F}|^{n-d_1}$ is exponential.*

5.2 Quantum Money from Obfuscation

Here, we recall Aaronson and Christiano's [3] construction, when instantiated with a subspace-hiding obfuscator.

Generating Banknotes. Let $\mathbb{F} = \mathbb{Z}_q$ for some prime q. Let λ be the security parameter. To generate a banknote, choose n a random even integer that is sufficiently large; we will choose n later, but it will depend on q and λ. Choose a random subspace $S \subseteq \mathbb{F}^n$ of dimension $n/2$. Let $S^\perp = \{x : x \cdot y = 0 \forall y \in S\}$ be the dual space to S.

Let $|\$_S\rangle = \frac{1}{|\mathbb{F}|^{n/4}} \sum_{x \in S} |x\rangle$. Let $P_0 = \mathsf{shO}(S)$ and $P_1 = \mathsf{shO}(S^\perp)$. Output $|\$_S\rangle, P_0, P_1$ as the quantum money state.

Verifying banknotes. Given a banknote state, first measure the program registers, obtaining P_0, P_1. These will be the serial number. Let $|\$\rangle$ be the remaining registers. First run P_0 in superposition, and measure the output. If P_0 outputs 0, reject. Otherwise continue. Notice that if $|\$\rangle$ is the honest banknote state $|\$_S\rangle$ and P_0 is the obfuscation of S, then P_0 will output 1 with certainty.

Next, perform the quantum Fourier transform (QFT) to $|\$\rangle$. Notice that if $|\$\rangle = |\$_S\rangle$, now the state is $|\$_{S^\perp}\rangle$. Next, apply P_1 in superposition and measure the result. In the case of an honest banknote, the result is 1 with certainty. Finally, perform the inverse QFT to return the state. In the case of an honest banknote, the state goes back to being exactly $|\$_S\rangle$. The above shows that the scheme is correct. Next, we argue security:

Theorem 7. *If* shO *is a secure subspace-hiding obfuscator for* $d_0 = n/2$ *and some* d_1 *such that both* $|\mathbb{F}|^{n-d_1}$ *and* $|\mathbb{F}|^{d_1 - n/2}$ *are exponentially-large, then the construction above is a secure quantum money scheme.*

Corollary 1. *If injective one-way functions and iO exist, then quantum money exists*

Proof. We now prove Theorem 7 through a sequence of hybrids

- H_0 is the normal security experiment for quantum money. Suppose the adversary, given a valid banknote, is able to produce two banknotes that pass verification with probability ϵ.
- H_1: here, we recall that Aaronson and Christiano's scheme is *projective*, so verification is equivalent to projecting onto the valid banknote state. Verifying two states is equivalent to projecting onto the product of two banknote states. Therefore, in H_1, instead of running verification, the challenger measures in the basis containing $|\$_S\rangle \times |\$_S\rangle$, and accepts if and only if the output is $|\$_S\rangle \times |\$_S\rangle$. The adversary's success probability is still ϵ.
- H_2: Here we invoke the security of shO to move P_0 to a higher-dimensional space. P_0 is moved to a random d_1 dimensional space containing S_0.
  We prove that the adversary's success probability in H_2 is negligibly close to ϵ. Suppose not. Then we construct an adversary B that does the following. B chooses a random $d_0 = n/2$-dimensional space S_0. It queries the challenger on S_0, to obtain a program P_0. It then obfuscates $S_0^\perp$ to obtain P_1. B constructs the quantum state $|\$_{S_0}\rangle$, and gives $P_0, P_1, |\$_{S_0}\rangle$ to A. A produces two (potentially entangled) quantum states $|\$_0\rangle|\$_1\rangle$. B measures in a basis containing $|\$_{S_0}\rangle \otimes |\$_{S_0}\rangle$, and outputs 1 if and only if $|\$_{S_0}\rangle \otimes |\$_{S_0}\rangle$.

If B is given P_0 which obfuscates S_0, then A outputs 1 with probability ϵ, since it perfectly simulates A's view in H_1. If P_0 obfuscates a random space containing S_0, then B simulates H_2. By the security of shO, we must have that B outputs 1 with probability at least $\epsilon - \mathsf{negl}$. Therefore, in H_2, A succeeds with probability $\epsilon - \mathsf{negl}$.

- H_3: Here we invoke security of shO to move P_1 to a random d_1-dimensional space containing $S_0^\perp$. By an almost identical analysis to he above, we have that A still succeeds with probability at least $\epsilon - \mathsf{negl}$.
- H_4. Here, we change how the subspaces are constructed. First, a random space T_0 of dimension d_1 is constructed. Then a random space T_1 of dimension d_1 is constructed, subject to $T_0^\perp \subseteq T_1$. These spaces are obfuscated using shO to get programs P_0, P_1. Aa random $n/2$-dimensional space S_0 is chosen such that $T_1^\perp \subseteq S_0 \subseteq T_0$. S_0 is used to construct the state $|\$_{S_0}\rangle$, which is given to A along with P_0, P_1. Then during verification, the space S_0 is used again.

  The distribution on spaces is identical to that in H_3, to A succeeds in H_4 with probability $\epsilon - \mathsf{negl}$.

Since on average over T_0, T_1, A succeeds with probability $\epsilon - \mathsf{negl}$, there exist fixed $T_0, T_1, T_0^\perp \subseteq T_1$, such that the adversary succeeds for these T_0, T_1 with probability at least $\epsilon - \mathsf{negl}$.

We now construct a no-cloning adversary C. C is given a state $|\$_{S_0}\rangle$ for a random S_0 such that $T_1^\perp \subseteq S_0 \subseteq T_0$, and it tries to clone $|\$_{S_0}\rangle$. To do so, it constructs obfuscations P_0, P_1 of T_0, T_1 using shO, and gives them along with $|\$_{S_0}\rangle$ to A. C then outputs whatever A outputs. C's probability of cloning is exactly the probability A succeeds in H_4, which is $\epsilon - \mathsf{negl}$. This gives an instance of the no-cloning problem. In the full version [38], we prove that the probability of cloning in this instance is at most $2|\mathbb{F}|^{-n'/2} = 2|\mathbb{F}|^{d_1 - n/2}$, which is exponentially small by the assumptions of the theorem. □

6 Constructing Quantum Lightning

6.1 Background

Degree-2 Polynomials over $\mathbb{Z}_q$. Consider a set $\mathcal{A}$ of n degree-2 polynomials over m variables in $\mathbb{Z}_q$ for some large prime q. Let $f_{\mathcal{A}} : \mathbb{Z}_q^m \to \mathbb{Z}_q^n$ be the function that evaluates each of the polynomials in $\mathcal{A}$ on its input. We will be interested in the compressing case, where $n < m$

As shown by [20, 6], the function $f_{\mathcal{A}}$ is *not* collision resistant when the coefficients of the polynomials are random. Here, we recreate the proof, and also discuss the multi-collision resistance of the function.

To find a collision for $f_{\mathcal{A}}$, choose a random $\Delta \in \{0,1\}^m$. We will find a collision of the form $\mathbf{x}, \mathbf{x}+\Delta$. The condition that $\mathbf{x}, \mathbf{x}+\Delta$ collide means $P(\mathbf{x}+\Delta) - P(\mathbf{x}) = 0$ for all polynomials in $\mathcal{A}$. Now, since P has degree 2, all the order-2 terms in $\mathbf{x}$ in this difference will cancel out, leaving only terms that are linear in $\mathbf{x}$ (and potentially quadratic in Δ). This gives us a system of linear equations over $\mathbf{x}$, which we can solve provided the equations are consistent. As shown in [6], these equations are consistent with overwhelming probability provided $n \leq m$.

This attack can be generalized to find $k+1$ colliding inputs. Choose random $\Delta_1, \dots, \Delta_k$. We will compute an $\mathbf{x}$ such that $\mathbf{x}, \mathbf{x}+\Delta_1, \dots, \mathbf{x}+\Delta_k$ form $k+1$ colliding points. Each Δ_j generates a system of n equations for $\mathbf{x}$ as described above. Let $B = B_{\Delta_1,\dots,\Delta_k}$ be the matrix consisting of all the rows of B_{Δ_j} as j varies. As long as B is full rank, a solution for $\mathbf{x}$ is guaranteed. Again, B will be full rank with overwhelming probability, provided $m \geq kn$. However, if $m \ll kn$, this procedure will fail, and it therefore appears reasonable to assume the multi-collision resistance of such functions.

Using the above, we can even generate superpositions over $k+1$ inputs such that all the inputs map to the same output. Consider the following procedure:

- Generate the uniform superposition $|\phi_0\rangle \propto \sum_{\Delta_1,\dots,\Delta_k} |\Delta_1, \dots, \Delta_k\rangle$
- Write $\mathbf{\Delta} = (\Delta_1, \dots, \Delta_k)$ In superposition, run the computation above that maps $\mathbf{\Delta}$ to the affine space $S_{\mathbf{\Delta}}$ such that, for all $\mathbf{x} \in S$, $f_{\mathcal{A}}(\mathbf{x}) = f_{\mathcal{A}}(\mathbf{x}+\Delta_j)$ for all j. This will be an affine space of dimension $m - nk$ with overwhelming probability. Then construct a uniform superposition of elements in $S_{\mathbf{\Delta}}$. The resulting state is then: $|\phi_1\rangle \propto \sum_{\mathbf{\Delta}} \frac{1}{\sqrt{|S_{\mathbf{\Delta}}|}} \sum_{\mathbf{x}\in S_{\mathbf{\Delta}}} |\Delta, \mathbf{x}\rangle$
- Next, in superposition, compute $f_{\mathcal{A}}(\mathbf{x})$, and measure the result to get a string $\mathbf{y}$. The resulting state is $|\phi_y\rangle \propto \sum_{\mathbf{\Delta},\mathbf{x}\in S_{\mathbf{\Delta}}:f_{\mathcal{A}}(\mathbf{x})=\mathbf{y}} \frac{1}{\sqrt{|S_{\mathbf{\Delta}}|}} |\mathbf{x}, \mathbf{\Delta}\rangle$
- Finally, in superposition, map $(\mathbf{x}, \Delta_1, \dots, \Delta_k)$ to $(\mathbf{x}, \mathbf{x}+\Delta_1, \dots, \mathbf{x}+\Delta_k)$. The resulting state is $|\psi_y\rangle \propto \sum_{\mathbf{\Delta},\mathbf{x}\in S_{\mathbf{\Delta}}:f_{\mathcal{A}}(\mathbf{x})=\mathbf{y}} \frac{1}{|S_{\mathbf{\Delta}}|} |\mathbf{x}, \mathbf{x}+\Delta_1, \dots\rangle$
  We note that the support of this state is *all* vectors $(\mathbf{x}_0, \dots, \mathbf{x}_k)$ such that $f_{\mathcal{A}}(\mathbf{x}_i) = \mathbf{y}$ for all $i \in [0, k]$. Moreover, for all but a negligible fraction, the weight $|S_{\mathbf{\Delta}}|$ is the same, and so the weights for these components are the same. Even more, the total weight of the other points is negligible. Therefore, the this state is negligibly close to the state $\sum_{\mathbf{x}_0,\dots,\mathbf{x}_k:f_{\mathcal{A}}(\mathbf{x}_i)=\mathbf{y}\forall i} |\mathbf{x}_0, \dots, \mathbf{x}_k\rangle = \left(\sum_{\mathbf{x}:f_{\mathcal{A}}(\mathbf{x})=\mathbf{y}} |\mathbf{x}\rangle\right)^{\otimes(k+1)} \propto |\psi'_{\mathbf{y}}\rangle^{\otimes(k+1)}$, where $|\psi'_{\mathbf{y}}\rangle \propto \sum_{\mathbf{x}:f_{\mathcal{A}}(\mathbf{x})=\mathbf{y}} |\mathbf{x}\rangle$.

Linear Functions over Rank-Constrained Matrices. Here, we consider a related problem. Consider a set of n linear functions $\mathcal{A}$ over rank-d matrices in $\mathbb{Z}_q^{m\times m}$. Since q is large, a random rank-d matrix in $\mathbb{Z}_q^{m\times m}$ will have it's first d columns span the entire column space. Therefore, most rank constrained matrices can be written as $(\mathbf{A} \;\; \mathbf{A}\cdot\mathbf{B})$ for a $m \times d$ matrix $\mathbf{A}$ and a $d \times (m-d)$ matrix $\mathbf{B}$.

Let $f_{\mathcal{A}} : \mathbb{Z}_q^{m^2} \to \mathbb{Z}_q^n$ be the function that evaluates each of the functions in $\mathcal{A}$ on its input. We can therefore think of $f_{\mathcal{A}}$ as a degree-2 polynomial over $\mathbf{A}, \mathbf{B}$. Note, however, that in this case, the function is bipartite: it can be divided into two sets of variables ($\mathbf{A}$ and $\mathbf{B}$) such that it is linear in each set. This means we can easily invert the function by choosing an arbitrary selection for one of the sets of variables, and then solving for the other.

Linear Functions over Rank-Constrained Symmetric Matrices. By instead considering only symmetric matrices, we essentially become equivalent to degree-2 polynomials. In particular, $\mathbf{A} \cdot \mathbf{A}^T$ for $\mathbf{A} \in \mathbb{Z}_q^{m \times d}$ is a symmetric rank-d matrix. Moreover, any degree-2 polynomial over $\mathbb{Z}_q^m$ can be represented as a linear polynomial over rank-1 symmetric matrices by tensoring the input with itself.

Note, however, that since $\mathbb{Z}_q$ is not a closed field, in general we cannot decompose any symmetric rank-d matrix into $\mathbf{A} \cdot \mathbf{A}^T$ (though we can over the closure). Therefore, linear functions over rank-constrained symmetric matrices can be seen as a slightly relaxed version of the degree-2 polynomials discussed above. In particular it is straightforward to generalize the algorithm for generating superpositions of colliding inputs to generate uniform superpositions of low-rank matrices that collide.

6.2 Hardness Assumption

Our assumption will have parameters n, m, q, d, e, k, to be described in the following discussion. Let $\mathcal{D}$ be the set of symmetric $m \times m$ matrices over $\mathbb{Z}_q$ whose rank is at most d. We will alternately think of $\mathcal{D}$ as matrices, as well as vectors by writing out all of the $\binom{m+1}{2}$ entries on and above the diagonal.

Let $\mathcal{A}$ be a set of n linear functions over $\mathcal{D}$, which we will think of as being n linear functions over $\binom{m+1}{2}$ variables. Consider the function $f_{\mathcal{A}} : \mathcal{D} \to \mathbb{Z}_q^n$ given by evaluating each linear function in $\mathcal{A}$.

As discussed above, we could imagine assuming that $f_{\mathcal{A}}$ is multi-collision resistant for a random set of linear functions $\mathcal{A}$. However, in order for our ultimate bolt verification procedure to work, we will need $\mathcal{A}$ to have a special form.

$\mathcal{A}$ is sampled as follows. Let $\mathbf{R} \in \mathbb{Z}_q^{e \times m}$ be chosen at random. Consider the set of symmetric matrices $\mathbf{A} \in \mathbb{Z}_q^{m \times m}$ such that $\mathbf{R} \cdot \mathbf{A} \cdot \mathbf{R}^T = 0$. This is a linear subspace of dimension $\binom{m+1}{2} - \binom{e+1}{2}$ (note that since $\mathbf{B}$ is symmetric, $\mathbf{R} \cdot \mathbf{A} \cdot \mathbf{R}^T$ is guaranteed to be symmetric, so we only get $\binom{e+1}{2}$ equations). We can think of each $\mathbf{A}$ as represented by its $\binom{m+1}{2}$ upper-triangular entries, which gives us an equation over $\binom{m+1}{2}$ variables. Let $\mathcal{A}$ be a basis for this space of linear functions. Thus, we set $n = \binom{m+1}{2} - \binom{e+1}{2}$. Note that we will *not* keep $\mathbf{R}$ secret. Rank d symmetric matrices in $\mathbb{Z}_q^{m \times m}$ have $\binom{d+1}{2} + d \times (m-d) = d \times m - \binom{d}{2}$ degrees of freedom. Therefore, the function $f_{\mathcal{A}}$ will be compressing provided that $d \times m - \binom{d}{2} > n = \binom{m+1}{2} - \binom{e+1}{2}$.

By choosing $f_{\mathcal{A}}$ in this way, we provide a "trapdoor" $\mathbf{R}$ that will be used for verifying bolts. This trapdoor is a rank-e matrix in the kernel of $f_{\mathcal{A}}$. If $e < d$, this would allow us to compute many colliding inputs, as, for any rank-1 $\mathbf{S}$, the whole affine space $\mathbf{S} + \alpha \mathbf{R}$ has rank at most $e + 1 \leq d$ and maps to the same value. However, if we choose $e > d$, $\mathbf{R}$ does not appear to let us find collisions.

Our Assumption. We now make the following hardness assumption. We say a hash function f is k-multi-collision resistant (k-MCR) if it is computationally infeasible to find k colliding inputs.

Assumption 8 *There exists some functions n, d, e, k in m such that $n = \binom{m+1}{2} - \binom{e+1}{2} < d \times m - \binom{d}{2}$, $kn \leq d \times m - \binom{d}{2} < (2k+1)n$, and $e > d$, such that $f_{\mathcal{A}}$ as sampled above is $(2k+2)$-MCR, even if $\mathbf{R}$ is given to the adversary.*

For example, we can choose e, d such that $m = e + d - 1$, in which case $n = d \times m - \binom{d}{2} - e$. By choosing $e \approx d$, we have $d \times m - \binom{d}{2} \ll 3n$, so we can set k to be 1. We therefore assume that it is computationally infeasible to find 4 colliding inputs to $f_{\mathcal{A}}$.

We stress that this assumption is highly speculative and untested. In the full version [38], we discuss in more depth possible attacks on the assumption, as well as weakened versions that are still sufficient for our purposes.

6.3 Quantum Lightning

We now describe our quantum lightning construction.

Parameters. Our scheme will be parameterized by integers n, m, q, d, e, k.

Setup. To set up the quantum lightning scheme, simply sample $\mathcal{A}, \mathbf{R}$ as above, and output $\mathcal{A}, \mathbf{R}$.

Bolt Generation. We generate a bolt $|\lightning_{\mathbf{y}}\rangle$ as a superposition of $k+1$ colliding inputs, following the procedure described above. The result is statistically close to $|\lightning'_{\mathbf{y}}\rangle^{\otimes(k+1)}$ where $|\lightning'_{\mathbf{y}}\rangle$ is the equally-weighted superposition over rank-d symmetric matrices such that applying $f_{\mathcal{A}}$ gives $\mathbf{y}$. We will call $|\lightning'_{\mathbf{y}}\rangle$ a mini-bolt

Verifying a bolt. Full verification of a bolt will run a mini verification on each of the $k+1$ mini-bolts. Each mini verification will output an element in $\mathbb{Z}_q^n \cup \{\perp\}$. Full verification will accept and output y only if each mini verification accepts and outputs the same string y. We now describe the mini verification.

Roughly, our goal is to be able to distinguish $|\lightning'_{\mathbf{y}}\rangle$ for some $\mathbf{y}$ from any singleton state. We will output $\mathbf{y}$ in this case, and for any other state, reject.

Mini verification on a state $|\phi\rangle$ will proceed in two steps. Recall that superposition is over the upper triangular portion of $m \times m$ matrices. We first apply, in superposition, the procedure that flips some external bit if the input does not correspond to a matrix of rank at most d. The bit is initially set to 0. Then we measure this bit, and abort if it is 1. Notice that for the honest $|\lightning'_{\mathbf{y}}\rangle$ state, this will pass with certainty and not affect the state.

In the next step, we apply the procedure that evaluates $f_{\mathcal{A}}$ in superposition, and flips some external bit if the result is not $\mathbf{y}$. The bit is initially set to 0. Then we measure this bit, and abort if it is 1. Notice that for the honest $|\lightning'_{\mathbf{y}}\rangle$ state, this will pass with certainty and not affect the state.

At this point, if we have not aborted, our state is a superposition of pre-images of $f_{\mathcal{A}}$ which correspond to symmetric rank-d matrices. If our input was $|\lightning'_{\mathbf{y}}\rangle$, the state is the uniform superposition over such pre-images.

Next, we verify that the state is in superposition and not a singleton state. To do so, we perform the quantum Fourier transform (QFT) to the state. We now analyze what the QFT does to $|\lightning_{\mathbf{y}}\rangle$.

The support of $|\lightning'_{\mathbf{y}}\rangle$ is the intersection of sets S, T where S is the set of all pre-images of $\mathbf{y}$ (not necessarily rank constrained) and T is the set of all rank-d matrices. We analyze the Fourier transform applied to each set separately.

Recall that the Fourier transform takes the uniform superposition over the kernel of a matrix to the uniform superposition over its row-span. Therefore, the superposition over pre-images of 0 is just the uniform superposition of symmetric matrices $\mathbf{A}$ such that $\mathbf{R}\cdot\mathbf{A}\cdot\mathbf{R}^T = 0$ (or technically, just the upper triangular part). The fact that the superposition lies in a coset of the kernel simply introduces a phase term to each element in the superposition.

In the full version [38], we prove the following claim:

Claim. The Fourier transform of the uniform superposition over upper-triangular parts of rank d symmetric matrices is negligibly close to the uniform superposition over upper triangular parts of rank $m - d$ symmetric matrices

Putting this together, since multiplication in the primal domain becomes convolution in the Fourier domain, after we apply the Fourier transform to our mini bolt state, the result is the the superposition of upper triangular parts of matrices $\mathbf{A} + \mathbf{B}$ where $\mathbf{B}$ is symmetric and rank $m - d$ and $\mathbf{A}$ is symmetric such that $\mathbf{R} \cdot \mathbf{A} \cdot \mathbf{R}^T = 0$. The superposition is uniform, though there will be a phase factor associated with each element

We therefore compute $\mathbf{R} \cdot (\mathbf{A} + \mathbf{B}) \cdot \mathbf{R}^T = \mathbf{R} \cdot \mathbf{B} \cdot \mathbf{R}^T$ and compute the rank. Notice that the rank is at most $m - d$ for honest bolt states. Therefore, if the rank is indeed at most $m - d$ we will accept, otherwise we will reject. Next, we un-compute $\mathbf{R} \cdot (\mathbf{A} + \mathbf{B}) \cdot \mathbf{R}^T$, and undo the Fourier transform. The analysis above shows that for an honest state $|\lightning_{\mathbf{y}}\rangle$, we will accept with overwhelming probability, and the final both state will be negligibly close to the original bolt.

Note that, in contrast, if the bolt state is a singleton state, then the Fourier transform will result in a uniform superposition over all symmetric matrices; when we apply$\mathbf{R} \cdot (\cdot) \cdot \mathbf{R}^T$, the result will have rank e with overwhelming probability. So we set $m - d < e$ to have an almost perfect distinguishing advantage.

Security. We now prove security. Consider a quantum adversary A that is given $\mathcal{A}$ and tries to construct two (possibly entangled) bolts $|\lightning_0\rangle, |\lightning_1\rangle$. Assume toward contradiction that with non-negligible probability, verification accepts on both bolts, and outputs the same serial number $\mathbf{y}$.

Conditioned on acceptance, by the above arguments the resulting mini bolts must all be far from singletons when we trace out the other bolts. This means that if we measure the mini-bolts, the resulting superpositions will have high min-entropy. Therefore, we measure all $2k + 2$ mini bolts, and we obtain $2k + 2$ colliding inputs that are distinct except with negligible probability. This violates our hardness assumption.

Theorem 9. *If Assumption 8 holds, then the scheme above is a secure quantum lightning scheme.*

In the full version [38], we show how to modify our construction to get a collapse-non-binding hash function.

References

1. S. Aaronson. http://www.scottaaronson.com/blog/?p=2854.

2. S. 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.
3. S. Aaronson and P. Christiano. Quantum money from hidden subspaces. In H. J. Karloff and T. Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012.
4. M. R. Albrecht, S. Bai, and L. Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178. Springer, Heidelberg, Aug. 2016.
5. A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, Oct. 2014.
6. B. Applebaum, N. Haramaty, Y. Ishai, E. Kushilevitz, and V. Vaikuntanathan. Low-complexity cryptographic hash functions. In C. H. Papadimitriou, editor, *ITCS 2017*, volume 4266, pages 7:1–7:31, 67, Jan. 2017. LIPIcs.
7. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, Aug. 2001.
8. C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, Oct. 1997.
9. C. H. Bennett and G. Brassard. Quantum public key distribution reinvented. *SIGACT News*, 18(4):51–53, July 1987.
10. D. Boneh and M. Zhandry. Quantum-secure message authentication codes. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013.
11. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, Aug. 2013.
12. Z. Brakerski, P. Christiano, U. Mahadev, U. V. Vazirani, and T. Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In M. Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, Oct. 2018.
13. Z. Brakerski, V. Vaikuntanathan, H. Wee, and D. Wichs. Obfuscating conjunctions under entropic ring LWE. In M. Sudan, editor, *ITCS 2016*, pages 147–156. ACM, Jan. 2016.
14. Y. Chen, C. Gentry, and S. Halevi. Cryptanalyses of candidate branching program obfuscators. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 278–307. Springer, Heidelberg, Apr. / May 2017.
15. J. H. Cheon, J. Jeong, and C. Lee. An algorithm for CSPR problems and cryptanalysis of the GGH multilinear map without an encoding of zero. Technical report, Cryptology ePrint Archive, Report 2016/139, 2016.
16. R. Colbeck. Quantum and relativistic protocols for secure multi-party computation, 2009.
17. J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493. Springer, Heidelberg, Aug. 2013.
18. R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering short generators of principal ideals in cyclotomic rings. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 559–585. Springer, Heidelberg, May 2016.

19. I. Damgård, J. Funder, J. B. Nielsen, and L. Salvail. Superposition attacks on cryptographic protocols. In C. Padró, editor, *ICITS 13*, volume 8317 of *LNCS*, pages 142–161. Springer, Heidelberg, 2014.
20. J. Ding and B.-Y. Yang. *Multivariates Polynomials for Hashing*, pages 358–371. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
21. E. Farhi, D. Gosset, A. Hassidim, A. Lutomirski, and P. W. Shor. Quantum money from knots. In S. Goldwasser, editor, *ITCS 2012*, pages 276–289. ACM, Jan. 2012.
22. S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.
23. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013.
24. S. Garg, H. Yuen, and M. Zhandry. New security notions and feasibility results for authentication of quantum data. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 342–371. Springer, Heidelberg, Aug. 2017.
25. D. Gavinsky. Quantum money with classical verification, 2011.
26. C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Heidelberg, Mar. 2015.
27. R. Goyal, V. Koppula, and B. Waters. Lockable obfuscation. In C. Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, Oct. 2017.
28. A. Lutomirski. An online attack against wiesner's quantum money, 2010.
29. A. Lutomirski, S. Aaronson, E. Farhi, D. Gosset, J. A. Kelner, A. Hassidim, and P. W. Shor. Breaking and making quantum money: Toward a new quantum cryptographic protocol. In A. C.-C. Yao, editor, *ICS 2010*, pages 20–31. Tsinghua University Press, Jan. 2010.
30. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, Apr. 2007.
31. M. Mosca and D. Stebila. Quantum coins. In *Error-Correcting Codes, Finite Geometries and Cryptography*, volume 523, pages 35–47, 2010.
32. M. C. Pena, J.-C. Faugère, and L. Perret. Algebraic cryptanalysis of a quantum money scheme: The noise-free case. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 194–213. Springer, Heidelberg, Mar. / Apr. 2015.
33. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
34. D. Unruh. Revocable quantum timed-release encryption. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, Heidelberg, May 2014.
35. D. Unruh. Computationally binding quantum commitments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.
36. D. Wichs and G. Zirdelis. Obfuscating compute-and-compare programs under LWE. In C. Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, Oct. 2017.
37. S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, Jan. 1983.
38. M. Zhandry. Quantum lightning never strikes the same state twice. Cryptology ePrint Archive, Report 2017/1080, 2017. https://eprint.iacr.org/2017/1080.