On Quantum Money and Evasive Obfuscation

Mark Zhandry
NTT Research
mzhandry@gmail.com

Abstract

We show a black box barrier against constructing public key quantum money from obfuscation for evasive functions. As current post-quantum obfuscators based on standard assumptions are all evasive, this shows a fundamental barrier to achieving public key quantum money from standard tools. Our impossibility applies to black box schemes where (1) obfuscation queries made by the mint are classical, and (2) the verifier only makes (possibly quantum) evaluation queries, but no obfuscation queries. This class seems to capture any natural method of using obfuscation to build quantum money.

1 Introduction

Wiesner's quantum money [Wie83] is the original quantum cryptographic protocol, dating back to 1970 (though not published until much later). Wiesner's scheme, however, has a number of undesirable drawbacks [Aar09, Lut10]. To remedy these problems, Aaronson [Aar09] proposes the goal of constructing *public key* quantum money, where banknotes can be verified by *anyone*, while only the mint has the ability to generate new notes. We will henceforth use the term "quantum money" to refer to its public key variant, which in the last 15 years has become one of the central goals in the field of quantum cryptography. Unfortunately, while there have been a number of proposals for quantum money [Aar09, FGH$^+$12, AC12, Kan18, Zha19, KSS21, KLS22, LMZ23, Zha24], many of them were subsequently broken [LAF$^+$10, CPDDF$^+$19, Rob21, LMZ23], and the bulk of the ones that remain (e.g. [FGH$^+$12, Kan18, KSS21, LMZ23, Zha24]) rely on new and poorly understood computational assumptions.

The closest we have to a proof relative to standard assumptions is using obfuscation. General purpose program obfuscation, formalized by [BGI$^+$01] and first constructed by [GGH$^+$13], has been a powerful tool for cryptography. Since its introduction, obfuscation has enabled numerous novel applications in cryptography and more generally computer science theory (e.g. [SW14, HSW14, BZ14, BPR15, CHN$^+$16, CPR17, GSWW22]). In the setting of quantum money, Ben-David and Sattath [BDS16] propose instantiating the hidden subspace framework of [AC12] with general purpose program obfuscation. Zhandry [Zha19] subsequently proves the scheme is secure assuming a notion of obfuscation called *indistinguishability obfuscation* (iO), or an even weaker form of obfuscation implied by iO referred to as subspace-hiding obfuscation. This, however, punts the analysis of quantum money security to the study of obfuscation, and while there are several candidate post-quantum iO constructions (e.g. [GGH15, BGMZ18, BDGM20, WW21]), they too rely on new and poorly understood computational assumptions, some of which have been challenged [HJL21, JLLS23].

The difficulty of building quantum money extends to many other related unclonable objects. For example, quantum copy protection [Aar09] for either programs producing digital signatures or for programs decrypting ciphertexts of a public key encryption scheme would imply quantum money. Likewise, certain forms of software leasing [AL21] imply quantum money. Essentially, any kind of unclonable object that admits public verification likely can be used to realize quantum money. As such, any attempt to build these objects would first have to build quantum money along the way.

In this work, we make progress towards justifying the difficulty of building public key quantum money by showing a *black box* impossibility for a natural class of protocols build from *evasive obfuscation*.

The Evasive Barrier. In a pair of beautiful concurrent works [GKW17, WZ17], it was shown how to obfuscate a class of functions called compute-and-compare (C&C) programs. These programs represent the state-of-the-art for what is achievable from standard post-quantum lattice-based assumptions, namely Learning with Errors (LWE). C&C programs are very general, encompassing many natural programs, such as pattern matching with wildcards, hyperplane membership, certain programs involving decryption functions, and more. This generality has led to a number of classical and quantum applications (e.g. [CVW$^+$18a, BKP19, BS20, AL21, ABG$^+$21, Zha21, CLLZ21]). Some works [Tsa22, VWW22] even construct "Null iO", a type of obfuscation for all-zeros circuits, which includes as a special case witness encryption [GGSW13]. Their assumption is a plausible yet non-standard variant of LWE called "evasive LWE."

Unfortunately, C&C obfuscation, Null iO, and witness encryption only apply to sub-classes of *evasive* functions. These are (distributions over) functions where any input is negligibly likely to be accepted by the function. For example, in hyperplane membership, a random hyperplane (with random offset from the origin) of low dimension is chosen. In this case, C&C obfuscation guarantees that nothing is learned about the hyperplane. But nothing is guaranteed if, say, there were a certain point p and the hyperplane is chosen from a non-uniform distribution that contains p with noticeable probability.

That LWE seems limited to evasive functions is a well-known barrier, and is closely related to other well-known barriers. An example is the fact that LWE can be used to realize predicate encryption [GVW15, WZ17], which can be seen as an evasive form of functional encryption, while general functional encryption from LWE remains out of reach. It should be noted that obfuscation for general evasive functions is at the same time much more extensive than what is currently achievable from LWE and even its non-standard variants.

Given that obfuscation is one of the most promising routes toward convincing quantum money schemes, but techniques from standard assumptions seem limited to evasive obfuscation, it is therefore natural to ask: *can quantum money be built from obfuscation for evasive function families?*

Possible strategies for using evasive obfuscation. The subspace-hiding obfuscation used to achieve quantum money in [Zha19] looks tantalizingly similar to the hyperplane obfuscation that can be achieved using C&C obfuscation. Indeed, both obfuscate subspaces/hyperplanes. However, the security notion is different in some important ways. Hyperplane obfuscation via C&C obfuscation must be evasive, and loses all security guarantees if a single accepting input is known. Meanwhile, subspace hiding obfuscation is inherently non-evasive. For starters, as defined in [Zha19], subspace hiding obfuscation obfuscates spaces S that pass through the origin. These can never be evasive, since the origin is an accepting input. The origin can be easily excluded in order to obfuscate with C&C obfuscation, and also later work [CLLZ21] extend to cosets, which do not pass through the origin at all. But even here, the notion of subspace-hiding obfuscation is still inherently non-evasive. This is because the security model for subspace-hiding obfuscation additionally gives the adversary a description of a non-trivial subspace T of S, and requires S to still be hidden. But such T trivially allows the adversary to find accepting inputs, violating evasiveness.

On the other hand, in the actual security game for quantum money, the adversary is never given such a subspace T. Instead, T is only an artifact of the security proof in [Zha19]. This means that subspace-hiding obfuscation is potentially stronger than what is actually needed to prove security. Given how close subspace-hiding is to what can be achieved by C&C obfuscation, it is natural to wonder if a better proof strategy could avoid giving the adversary points in S, thereby allowing a security proof under evasive subspace obfuscation.

Digging deeper, while the quantum money security game does not explicitly hand out accepting inputs to S, it *does* give the adversary the ability to compute an accepting input to S. Indeed, the adversary is given, in addition to an obfuscation of S (and also and obfuscation of $S^\perp$), a valid quantum money state, which is a uniform superposition over S. By measuring the money state, the adversary obtains an element in S. Likewise, the adversary can measure the money state in the Fourier basis to obtain an element in $S^\perp$. However, it does not appear possible to obtain an accepting input in both S and $S^\perp$ simultaneously[1]. Oone may hope that this means that *one* of S and $S^\perp$ may remain evasive.

Another direction would be to divide the subspace S into two pieces S_0, S_1, and obfuscate each piece separately. For instance, if S is a subspace of $\mathbb{F}_2^n$, then S_0, S_1 could actually be two affine sets. The adversary can measure the banknote superposition, which gives an accepting input to S_0 or S_1, violating the evasiveness of one of the spaces. But importantly, no inputs are generated for the other space. Therefore, it may seem that we can invoke evasive obfuscation for at least one of S_0, S_1.

It is also possible to move beyond subspaces, and perhaps obfuscate more complex sets S. For example, if one could find an (evasive) set S such that the Fourier transform of the uniform superposition over S has small support S', then the money state could be the uniform superposition over S, and the mint would output obfuscations of S and S'. Linear subspaces with $S' = S^\perp$ are but one example of such sets. Could more general sets be obfuscated sufficiently using evasive obfuscation to get secure quantum money?

A Black Box Impossibility. In this work, we make a significant step toward answering the above in the negative, by showing a black box barrier to achieving quantum money from evasive obfuscation. In particular, approaches like those above cannot work to yield quantum money from evasive obfuscation. Our main theorem is the following:

Theorem 1. *There is no black box construction of quantum money from evasive obfuscation where (1) obfuscation queries made by the mint are classical, and (2) the verifier makes no obfuscation queries.*

The class of quantum money schemes captured by Theorem 1 capture most natural approaches to building quantum money via obfuscation. Indeed, [AC12] has this form. Additionally, all existing

[1] Aside from the obvious point at the origin, which can be eliminated by moving to cosets.

constructions of copy protection using obfuscation [AL21, ALL+21, CLLZ21, LLQZ22] are also of a similar form. We give a more in-depth discussion of such schemes in Section 3 below.

Our notion of evasive obfuscation also encompasses one-way functions as a special case, and so as a corollary, we also separate a natural class of quantum money protocols from one-way functions.

Our results help explain the difficulties in constructing public key quantum money, and give guidance for future attempts at building it from evasive obfuscation – including one-way functions – by showing a wide class of natural schemes that must be avoided in any attempted construction. This is only the second known barrier for public key quantum money (following [AHY23]), and to the best of our knowledge the first barrier for constructing anything from evasive obfuscation, be it quantum protocols or otherwise. See Section 3 for a more thorough discussion of related work and interpretation of our results.

2 Proof Overview

2.1 How to model evasive obfuscation

We follow the typical approach in black box separation literature, and specify oracles relative to which evasive obfuscation exists, but quantum money of a particular flavor does not. The starting point is the natural approach to providing oracles for *ideal* obfuscation: provide a random oracle $\mathcal{O}$ mapping programs plus random coins to bit-strings that acts as the obfuscator, and an oracle $\mathcal{E}$ used to evaluate the obfuscated program. $\mathcal{E}$ will take as input an "obfuscated program" $\hat{P}$, which is just a bit-string produced by $\mathcal{O}$, and an input x. It inverts $\mathcal{O}$ to obtain the original un-obfuscated program P such that $\mathcal{O}(P) = \hat{P}$, and then outputs $P(x)$. Since we are interested in modeling quantum computation, we will allow these oracles to be accessed on quantum superpositions on inputs. It is straightforward to show that these oracles implement any reasonable notion of obfuscation, including evasive obfuscation but also the widely-used indistinguishability obfuscation (iO) or virtual black box (VBB) obfuscation, which is known to be impossible in the standard model [BGI+01].

The problem is that quantum money actually exists relative to this oracle! Indeed, this oracle can be used to implement the subspace membership oracles of [AC12] which are shown to give rise to quantum money. In fact, it was shown by [Zha19] that even iO suffices for building quantum money. So we need to ensure somehow that our oracles do not give iO.

Our solution is a simple tweak to $\mathcal{E}$ above. Whenever $P(x)$ would accept, in addition to outputting 1, $\mathcal{E}$ also outputs the code of the original un-obfuscated program P. Importantly, for rejecting inputs $\mathcal{E}$ still outputs only 0, giving no information about P other than the fact that $P(x) = 0$. For evasive obfuscation, the adversary will never be able to find an accepting input, meaning this change to $\mathcal{E}$ does not affect the adversary's view. Thus the oracles remain secure for evasive programs. However, for non-evasive programs, all security is compromised since a single accepting input reveals the original program in the clear.

Remark 1. *Our oracles are in fact not far from what happens in the case of LWE-based obfuscators for C&C programs and Null iO [GKW17, WZ17, Tsa22, VWW22], and related candidates for iO based on lattices [GGH15, HHSS17]. Indeed, in these constructions, an accepting input leads to certain leakage, which has been exploited in attacks [CGH17, CVW18b] to obtain information about the original program.*

As is typical in the black box separation literature, we will restrict the adversary to polynomially-many queries to $\mathcal{O}, \mathcal{E}$, but we allow for unbounded computation outside the queries. This is to rule out trivial constructions that simply assume a standard-model quantum money scheme (such as one based on existing candidates for iO) and ignore the oracle. By giving the adversary unbounded power outside the oracles, all such standard-model constructions are insecure, and the only way to obtain a secure scheme is to use the given oracles. Thus this model captures schemes that actually use evasive obfuscation as their source of security. Proving an impossibility in this model thus justifies the impossibility of using evasive obfuscation to construct quantum money.

2.2 Our impossibility

We now turn to actually proving that quantum money is impossible in our oracle model. Let Gen be the note generation procedure, and Ver the verifier. Per [AC12], we will only consider the case of a "mini-scheme", where there is only a single banknote. That is, Gen samples a serial number σ and money state \$, and Ver verifies that \$ is a valid banknote with respect to serial number σ. The adversary is given $\sigma, \$$, and tries to construct two (potentially entangled) banknotes $\$_1, \$_2$ that both pass verification with respect to σ.

Recall we restrict to protocols where the mint's obfuscation queries are classical, and the verifier makes no obfuscation queries. A standard argument shows we can assume the mint makes no verification queries, and the verifier only evaluates programs obfuscated by the mint. We call such schemes "respecting."

Step 1: Measuring a random query. Our starting point is the following: suppose Ver's queries to $\mathcal{E}$ place very little weight on any accepting inputs to programs queried by the mint. In this case, we can actually simulate Ver's queries to $\mathcal{E}$ by ourselves by simply responding to each query with reject. This will introduce some error related to the weight of accepting inputs in the queries, which is small by assumption. Thus we get an approximately correct verifier that makes no queries at all. It is easy to see that such a verifier is impossible in our model, since the adversary can use its unbounded computation to brute-force arbitrarily-many states that are accepted by the verifier.

On the other hand, if Ver's queries place significant weight on accepting inputs to programs queried by the mint, then we can measure a random query and we will find an accepting input with reasonable probability. Then we can run $\mathcal{E}$ on the measured query, and since $\mathcal{E}$ accepts, we actually learn one of the original un-obfuscated programs. If the mint only obfuscated a single program, we can again simulate Ver's queries since we now have de-obfuscated the only program and can evaluate it without making queries. As before, we can use our unbounded computation to brute-force many accepting money states.

But what if the mint obfuscates many programs, and Ver places significant weight on accepting inputs to the various different programs? For example, this occurs in our example of splitting the subspace S into two obfuscated spaces S_0, S_1. Ver will test membership in S by testing membership in $S_0 \cup S_1$, which involves making superposition queries to each of the spaces S_0, S_1.

When we measure a query, we are likely to find an accepting input — and hence the original un-obfuscated program — for *one* of the obfuscated programs. But the other programs so far will remain secure, meaning we cannot simulate Ver's queries on these programs. The problem is that our measurement has now destroyed the one money state we have. Thus, we may not be able to produce even a single valid money state from what we have, let alone two. And we cannot brute-force the state since we still have to make queries to $\mathcal{E}$ to evaluate the remaining secure programs, and we are query-bounded.

Step 2: State repair. Our solution to the above problem is to *repair* the quantum money state after measuring a random query. Specifically, we employ the quantum state repair procedure of [CMSZ22]. This procedure gives the following informal guarantee: suppose a state is accepted by some binary-outcome measurement M_1, and then is fed into another multi-outcome measurement M_2. Applying M_2 may have destroyed the state and M_1 may no longer accept, but by repairing the state using the procedure in [CMSZ22] we return to a state accepted by M_1, though it may be different than the original accepting state.

In our case, M_1 will be $\mathsf{Ver}(\sigma, \cdot)$, and M_2 will be our measurement of a random query to $\mathcal{E}$. After repairing the state, we therefore have simultaneously one of the original un-obfuscated programs, but also a quantum money state that passes verification. We can therefore repeat the process, measuring a random query again, potentially getting another one of the original programs. Since there are only a polynomial number of programs, after sufficiently-many iterations of this process, we will eventually stop obtaining new programs. At this point the query weight on programs we do not have is small, meaning we can approximately simulate Ver's queries and perform the brute-force search for money states.

There are several caveats to getting the repair procedure to work here. Some will be discussed below. But here, we briefly point out that the repair procedure runs in time that grows with the number of measurement outcomes of M_2. In particular, in our oracle setting, the number of queries the above strategy must make grows with the number of possible measurement outcomes, which must therefore be polynomial. This means we cannot naively measure the query, as the query domain contains exponentially many (obfuscated-program, input-to-program) pairs. However, we can take the number of possible query *responses* to actually be a polynomial: if the mint obfuscates P programs, the response can be 0 for a rejecting input, or 1 together with one of the P original un-obfuscated programs in the case of accepting inputs. Therefore, by only measuring the query response as opposed to the query itself, we obtain the information we are after (the original program) while only having polynomially-many outcomes.

When applied to our running example of splitting the space S into separate spaces S_0, S_1, the point is that it actually is true that one of the obfuscations of S_0, S_1 may remain secure if we are only given these two obfuscations, but fails since we must give out more than obfuscations of S_0, S_1 to have a secure scheme. In more detail, recall that by measuring the banknote (which recall is a uniform superposition over S), the adversary will learn an input to one of S_0, S_1 but not both. Using our evasive obfuscation oracle, the adversary will therefore learn one of the spaces S_0, S_1, but not both. With only knowledge of one of S_0, S_1, the adversary is unable to produce two valid banknotes. The problem is that in [AC12], having a membership tester for S (which in our case is obtained by membership testing for S_0, S_1) is not enough to give a secure quantum money scheme. Instead, the verifier also needs to be able to check membership in the dual space $S^\perp$ in order to give a secure scheme. But once we give out any means to verify banknotes securely (say through obfuscating $S^\perp$ or other means), we can run state repair to get back to a valid banknote. Once we have a valid banknote, we can measure again, and with reasonable probability we will learn the other space. Once we learn both S_0, S_1, we therefore know S and can easily forge new banknotes. Observe that this argument is very general, as it did not actually use that S_0, S_1 or $S^\perp$ were subspaces, or any details about how verification works.

Compatible verifiers. Besides requiring M_2 to have polynomially-many outcomes, there are other requirements we need to be aware of to get state repair from [CMSZ22] to work. First, we need that M_1 — that is, Ver — is projective, or at least a relaxation called *almost projective*. While many quantum money schemes such as [AC12] are indeed projective, this is not a normal requirement to be a valid money scheme. Using dilation, we can assume Ver is projective over the joint system of the money state and some ancilla qubits that are initialized to $|0\rangle$. We could try defining the money state as this joint system, but the naive solution may not be secure: an adversary may be able to fool Ver into accepting bad money states if those states have the ancilla qubit set to something other than $|0\rangle$. After all, Ver is only guaranteed to work correctly when the ancilla is initialized $|0\rangle$, but an adversary that supplies the ancilla can initialize it to anything. Of course, we could have Ver first perform a measurement to check that the ancilla are $|0\rangle$, but now Ver is not projective any more since measuring for $|0\rangle$ and evaluating Ver might be incompatible measurements.

A more subtle issue is that state repair requires M_2 to be projective as well, meaning measuring Ver's queries needs to be projective. In general, this is not the case. Indeed, the queries will be written to some ancilla qubits in order to be sent to the oracle. One can try to un-compute the query post-measurement by running the verifier in reverse, but because of the measurement, the ancilla in general will remain, and may be entangled with the money state. This means the measurement cannot be projective on the money state itself. The measurement will be projective on the joint system including the ancilla, but in this case we run into the issue above that Ver could be fooled by initializing a bad ancilla.

Our solution is to view Ver as a projective measurement over the joint system of money state and ancilla, but carefully re-design Ver to ensure that it remains secure. Essentially, Ver now performs a mixture of two projective measurements: the original verification viewed as a projection over the joint system, and the projection of the ancilla onto $|0\rangle$. By using techniques developed by [MW04, Zha20], we can implement a measurement that is a weighted average of both measurements. Importantly, by carefully choosing the relative weights of the two measurements, we can enforce that any state which passes our new verification has almost all its mass on states accepted by each of the two projections. This allows us to show that our new verifier is both correct and secure. The resulting verifier is also *almost* projective [Zha20], which is sufficient for state repair.

This solution also resolves the issue of M_2 being projective: M_2 considered as a measurement over the joint system including the ancilla is projective.

Remark 2. *An interesting consequence is that, once the verifier is (almost) projective, we can amplify correctness without further blowing up the note size: basically, when generating a note, check if the note is accepted by verification, and if not, discard the note and start over. This gives an alternative means to amplify correctness of weakly correct schemes, which may be more efficient in terms of storage requirements than the naive solution of parallel repetition.*

3 Discussion, Other Related Work, and Open Questions

Additional Motivation: black box separations for quantum protocols. Black box separations are a useful tool for understanding the (in)feasibility of constructing certain objects. Besides helping to explain why the community has not been able to give better constructions, black box separations also guide future research efforts by showing what types of constructions *cannot* work.

In our case, our results show that any quantum money construction from evasive obfuscation must either be non-respecting or make non-black-box use of the obfuscator.

While numerous separations are known in the classical setting, the literature on black box separations for *quantum* protocols is relatively sparse, with only a few prior results [HY20, ACC$^+$22, CLM23, AHY23]. As such, there are relatively few known techniques for proving such separations.

At a technical level, a common classical technique for black box separations (and the general approach we employ in our separation) is to run one or more of the cryptographic algorithms several times, view the queries the algorithms make to their oracles, then use this information in some way. For example, many separations will use this information to try to "compile out" the use of the oracle from the algorithm, by using the information to simulate the oracle responses. Unfortunately, viewing adversary's queries, and in particular the compiling out approach, can be difficult to adapt to the quantum setting. The reason is that, if the algorithm makes quantum queries to the oracle, then it is impossible to observe the query without perturbing the algorithm's state. Once even a small amount of information is extracted from the queries, this perturbation may make it hard or impossible to continue running the algorithm.

Perhaps because of challenges like these, existing quantum separations have been limited to separating from relatively simple objects like one-way functions or one-way permutations.

Our work demonstrates how quantum state repair can, at least in certain cases, be used to compile out oracle queries from quantum protocols. By using our new technique, we are able to separate quantum money from evasive obfuscation, a very powerful and structured object. Our ideas therefore further the set of techniques available for proving separations in the quantum setting.

On our restricted class of protocols. We briefly discuss our restrictions on protocols handled by Theorem 1, by arguing that such protocols capture essentially any natural construction using obfuscation, including all known obfuscation-based approaches to quantum money. First, it is unclear what use the verifier would have in actually obfuscating programs, since the point of obfuscating a program is to hide something, but the verifier does not need to hide anything from itself. So it seems natural verifiers would not make use of the obfuscation functionality, just the evaluation functionality. It is also unclear what the mint would accomplish by obfuscating a superposition of programs: since obfuscated programs are random strings, it is possible to show that the superposition of obfuscated programs is indistinguishable from being fully measured. In this case, we might as well have just measured the query in the first place, meaning only classical queries are required.

There may be some room to do non-trivial things by having the mint only produce a few bits of an obfuscated program. In this case, superpositions are *dis*tinguishable from measured. However, it is impossible to evaluate a program from just a few bits of its description, so it is unclear how to use program obfuscation in this way.

Implications. Our result separates a natural class of quantum money schemes from evasive obfuscation. This captures natural approaches to adapt [AC12] to the evasive setting. It also captures certain natural attempts to build quantum money from one-way functions. Indeed, any evasive obfuscator for, say, point functions is in particular trivially a one-way function. Applying our impossibility rules out construction of quantum money from a one-way function f, where the mint only makes classical queries to f, and the verifier cannot evaluate f, but can instead submit (superpositions of) pairs (x, y), to which it receives in response a (superposition of) bits indicating if $y = f(x)$.

More generally, our impossibility seems to capture natural constructions involving the evasive obfuscation of *non-cryptographic* functions. Care must be taken, however, when trying to interpret our results when applying evasive obfuscation to *cryptographic* programs. The motivation for considering cryptographic programs is that evasive obfuscation for such programs yields a number of powerful cryptographic objects, and we may hope to additionally get a separation from those objects through our impossibility. For example, evasive obfuscation applied to a one-way function (as opposed to considering the evasive obfuscator *as* the one-way function) gives public key encryption. We may therefore hope to conclude that there is no black box construction of quantum money from public key encryption. The problem is that our proof works in a model where we allow the adversary to be computationally unbounded. While we model the evasive obfuscator as an oracle and restrict the number of queries, any other cryptographic object that we may apply the evasive obfuscator to will be trivially broken in our model.

We could try to fix this issue by modeling the supposed cryptographic object as an oracle, and bound the queries to the oracle. This will require updating our model of evasive obfuscation to work with query-aided programs that may make queries to the cryptographic object. For example, we could model a one-way function by a random oracle, and then allow the evasive obfuscator to operate on programs that make random oracle queries. However, our impossibility will run the verifier for the quantum money scheme an exponential number of times. While we take care to first remove evasive obfuscation queries from the verifier (so that the impossibility only needs a polynomial number of queries to the evasive obfuscator), if the original verifier made any queries to the oracle implementing the additional cryptographic object, our impossibility will end up making an exponential number of queries to the object. Therefore, we need to restrict to protocols where the verifier makes no queries to the additional cryptographic object.

We now consider the application to public key encryption. Evasive obfuscation yields public key encryption as follows: the secret key is an input x to a one-way function f, and the public key is the output $y = f(x)$. The ciphertext is an obfuscation of the program that takes as input a candidate secret key x', and outputs the message m if and only if $f(x') = y$. As a consequence of the above discussion, we get an impossibility of constructing quantum money from public key encryption, subject to the following restrictions:

- The mint only encrypts using classical keys/messages/randomness.
- Verification *cannot* run key generation (as it would require making queries to f) and *cannot* encrypt messages (as it would require obfuscating programs).

While this gives *some* separation from public key encryption, it is unclear how meaningful it may be.

Existing black box separations for quantum money. The only prior black box separation for quantum money is the recent work of [AHY23]. That work shows it is impossible to build a black box construction of quantum money from random oracles, assuming that verification makes only classical queries to the underlying hash function. In particular, this shows that it is impossible to build quantum money of this form from collision resistance or one-way functions, the latter which can be seen as a special case of evasive obfuscation for point functions. Thus, our work complements theirs by offering more barriers to constructing quantum money from standard assumptions.

Other approaches to quantum money from lattices. Obfuscation is not the only approach to building quantum money. In particular, [Zha19] discusses a potential approach to building quantum money from lattice assumptions by constructing hash functions that are not *collapsing* [Unr16]. The recent candidate quantum money scheme of [KLS22] can also be seen as following this high-level idea. Unfortunately, [LZ19] show that Ajtai's [Ajt96] function based on lattices *is* collapsing. More recently, [LMZ23] show that a wide class of lattice-based quantum money schemes based on this approach cannot be secure. Combined with our work, this shows that all known approaches to building quantum money from tools implied by standard lattices are unlikely to work.

4 Preliminaries

4.1 Basic Quantum Preliminaries

Much of this section is taken almost verbatim from [CMSZ22]. A (pure) *quantum state* is a vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$ with $\||\psi\rangle\| = 1$; in this work, $\mathcal{H}$ is finite-dimensional. We denote by $\mathbf{S}(\mathcal{H})$ the space of Hermitian operators on $\mathcal{H}$. A *density matrix* is a positive semi-definite operator $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ with $\mathsf{Tr}(\boldsymbol{\rho}) = 1$. A density matrix represents a probabilistic mixture of pure states (a mixed state); the density matrix corresponding to the pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi|$. Typically we divide a Hilbert space into *registers*, e.g. $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$. We sometimes write, e.g., $\boldsymbol{\rho}^{\mathcal{H}_1}$ to specify that $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H}_1)$.

A unitary operation is a complex square matrix U such that $UU^\dagger = \mathbf{I}$. The operation U transforms the pure state $|\psi\rangle$ to the pure state $U|\psi\rangle$, and the density matrix $\boldsymbol{\rho}$ to the density matrix $U\boldsymbol{\rho}U^\dagger$.

A *projector* P is a Hermitian operator ($P^\dagger = P$) such that $P^2 = P$. A *projective measurement* is a collection of projectors $\mathbf{P} = (P_i)_{i\in S}$ such that $\sum_{i\in S} P_i = \mathbf{I}$. This implies that $P_iP_j = 0$ for distinct i and j in S. The application of $\mathbf{P}$ to a pure state $|\psi\rangle$ yields outcome $i \in S$ with probability $p_i = \|P_i|\psi\rangle\|^2$; in this case the post-measurement state is $|\psi_i\rangle = P_i|\psi\rangle/\sqrt{p_i}$. We refer to the post-measurement state $|\psi_i\rangle$ as the result of applying $\mathbf{P}$ to $|\psi\rangle$ and *post-selecting* (conditioning) on outcome i. A state $|\psi\rangle$ is an *eigenstate* of $\mathbf{P}$ if it is an eigenstate of every P_i.

A two-outcome projective measurement is called a *binary projective measurement*, and is written as $\mathbf{P} = (P, \mathbf{I} - P)$, where P is associated with the outcome 1, and $\mathbf{I} - P$ with the outcome 0.

General (non-unitary) evolution of a quantum state can be represented via a *completely-positive trace-preserving (CPTP)* map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H}')$. We omit the precise definition of these maps in this work; we only use the facts that they are trace-preserving (for every $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ it holds that $\mathsf{Tr}(T(\boldsymbol{\rho})) = \mathsf{Tr}(\boldsymbol{\rho})$) and linear.

For Hilbert spaces $\mathcal{A}, \mathcal{B}$ the *partial trace* over $\mathcal{B}$ is the unique CPTP map $\mathsf{Tr}_{\mathcal{B}} \colon \mathbf{S}(\mathcal{A}\otimes\mathcal{B}) \to \mathbf{H}(\mathcal{A})$ such that $\mathsf{Tr}_{\mathcal{B}}(\boldsymbol{\rho}_A \otimes \boldsymbol{\rho}_B) = \mathsf{Tr}(\boldsymbol{\rho}_B)\boldsymbol{\rho}_A$ for every $\boldsymbol{\rho}_A \in \mathbf{S}(\mathcal{A})$ and $\boldsymbol{\rho}_B \in \mathbf{S}(\mathcal{B})$.

For every CPTP map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H})$there exists a *unitary dilation* U that operates on an expanded Hilbert space $\mathcal{H} \otimes \mathcal{K}$, so that $T(\boldsymbol{\rho}) = \mathsf{Tr}_{\mathcal{K}}(U(\boldsymbol{\rho} \otimes |0\rangle\langle 0|^{\mathcal{K}})U^\dagger)$. This is not necessarily unique; however, if T is described as a circuit then there is a dilation U_T represented by a circuit of size $O(|T|)$.

A *general measurement* is a CPTP map $\mathbf{M} \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H} \otimes \mathcal{O})$, where $\mathcal{O}$ is an ancilla register holding a classical outcome. Specifically, given measurement operators $\{M_i\}_{i=1}^N$ such that $\sum_{i=1}^N M_i M_i^\dagger = \mathbf{I}$ and a basis $\{|i\rangle\}_{i=1}^N$ for $\mathcal{O}$, $\mathbf{M}(\boldsymbol{\rho}) = \sum_{i=1}^N (M_i \boldsymbol{\rho} M_i^\dagger \otimes |i\rangle\langle i|^{\mathcal{O}})$. We sometimes implicitly discard the outcome register. A projective measurement is a general measurement where the M_i are projectors. A measurement induces a probability distribution over its outcomes given by $\Pr[i] = \mathsf{Tr}(|i\rangle\langle i|^{\mathcal{O}}\mathbf{M}(\boldsymbol{\rho}))$; we denote sampling from this distribution by $i \leftarrow \mathbf{M}(\boldsymbol{\rho})$.

The *trace distance* between states $\boldsymbol{\rho}, \boldsymbol{\rho}'$, denoted $d(\boldsymbol{\rho}, \boldsymbol{\rho}')$, and is defined as $\frac{1}{2}\mathsf{Tr}(\sqrt{(\boldsymbol{\rho} - \boldsymbol{\rho}')^2})$. The trace distance is contractive under CPTP maps (for any CPTP map T, $d(T(\boldsymbol{\rho}), T(\boldsymbol{\rho}')) \leq d(\boldsymbol{\rho}, \boldsymbol{\rho}')$). It follows that for any measurement $\mathbf{M}$, the statistical distance between the distributions $\mathbf{M}(\boldsymbol{\rho})$ and $\mathbf{M}(\boldsymbol{\rho}')$ is bounded by $d(\boldsymbol{\rho}, \boldsymbol{\rho}')$.

We have the following *gentle measurement lemma*, which bounds how much a state is disturbed by applying a measurement whose outcome is almost certain.

Lemma 2 (Gentle Measurement [Win99])**.** *Let $\boldsymbol{\rho} \in \mathbf{S}(\mathcal{H})$ and $\mathbf{P} = (\Pi, \mathbf{I} - \Pi)$ be a binary projective measurement on $\mathcal{H}$ such that $\mathsf{Tr}(\Pi\rho) \geq 1 - \delta$. Let*

$$\boldsymbol{\rho}' = \frac{\Pi\rho\Pi}{\mathsf{Tr}(\Pi\rho)}$$

be the state after applying $\mathbf{P}$ to ρ and post-selecting on obtaining outcome 1. Then

$$d(\rho, \rho') \leq 2\sqrt{\delta}.$$

Quantum algorithms. In this work, a *quantum algorithm* is a quantum circuit built from some universal quantum gate set. We will usually consider the Hilbert space for a quantum algorithm as the product of of three registers $\mathcal{H}_{\mathsf{in}} \times \mathcal{H}_{\mathsf{out}} \times \mathcal{H}_{\mathsf{work}}$. Here, input x is mapped to $|x\rangle \in \mathcal{H}_{\mathsf{in}}$ where $\{|x\rangle\}$ is some basis for $\mathcal{H}_{\mathsf{in}}$, the output of the algorithm is obtained by measuring $\mathcal{H}_{\mathsf{out}}$ in some basis, and $\mathcal{H}_{\mathsf{work}}$ is an ancilla register used as work space.

One typically considers the universal gate set as being unitary, giving unitary algorithms. We can also consider algorithms with non-unitary gate sets, such as flipping random coins or making measurements. In this case, the algorithms are non-unitary, and we say they are *probabilistic*.

Black-box access A circuit C with black-box access to a unitary U, denoted C^U, is a standard quantum circuit with special gates that act as U and $U^\dagger$. We also use C^T to denote black-box access to a map T, which we interpret as C^{U_T} for a unitary dilation U_T of T; all of our results are independent of the choice of dilation. This allows, for example, the "partial application" of a projective measurement, and the implementation of a general measurement via a projective measurement on a larger space.

We model black-box (quantum) access to classical functions in the usual way: we make the classical function $f : \mathcal{X} \to \mathcal{Y}$ unitary by having it act on registers $\mathcal{H}_{\mathsf{in}} \times \mathcal{H}_{\mathsf{out}}$ by applying the map $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. We will slightly abuse notation and write programs having black box access to such a unitary as C^f.

Query weights and switching oracles. Consider an algorithm $\mathcal{A}^f$ making queries to a classical function f. During the ith query, the joint state of $\mathcal{A}$ and the query has the form $\sum_{x,y,z} \alpha_{x,y,z}|x, y, z\rangle$, where z is the state of $\mathcal{A}$. Let $w_i(x) = \sum_{y,z} |\alpha_{x,y,z}|^2$. $w_i(x)$ is then the "query weight" of x in the ith query to f. Let $w(x) = \sum_i w_i(x)$ where the sum ranges over all queries. $w(x)$ is then the "total query weight" of x. For a set S, we let $w(S) = \sum_{x \in S} w(x)$, which we call the total query weight of S. For a probabilistic (non-unitary) $\mathcal{A}$, we define $w_i(x), w(x), w(S)$ as the expectation of the query weight over the randomness of $\mathcal{A}$.

Another way to think of query weight is as follows: run $\mathcal{A}$ to query i, and measure $\mathcal{A}$'s state, outputting x. The probability of obtaining a given x is exactly $w_i(x)$. If i is chosen uniformly in $[q]$, then the probability of obtaining x is $w(x)/q$. The following result is paraphrased from [BBBV97]:

Lemma 3 ([BBBV97] Theorems 3.1 and 3.3)**.** *Let $\mathcal{A}^f$ be an algorithm making q queries to a function f. Let f' be another function, and S the set of points x such that $f(x) \neq f'(x)$. Suppose that the total query weight of $\mathcal{A}^f$ on S is at most ϵ. Then $|\Pr[\mathcal{A}^f() = 1] - \Pr[\mathcal{A}^{f'}() = 1]| \leq O(\sqrt{q\epsilon})$.*

[BBBV97] prove only the case of unitary algorithms, but the unitary case easy implies the probabilistic case via linearity of expectation and the concavity of $\sqrt{\cdot}$. We also note a simple corollary:

Corollary 4. *Let $\mathcal{A}^f$ be an algorithm making q queries to a function f. Let $\mathcal{A}^f$ be an algorithm making q queries to a function f. Let f' be another function, and S the set of points x such that $f(x) \neq f'(x)$. Suppose that the total query weight of $\mathcal{A}^f$ on S is at most ϵ. Then the total query weight of $\mathcal{A}^{f'}$ on S is at most $O(\sqrt{q^3\epsilon})$*

Proof. Let $\mathcal{B}^f$ be the algorithm which runs $\mathcal{A}$, samples a random $i \in [q]$, measures the ith query to obtain x, and outputs 1 if and only if $x \in S$. Then $\Pr[\mathcal{B}^f() = 1]$ is exactly $w(S)/q = \epsilon/q$. Meanwhile $\Pr[\mathcal{B}^{f'}() = 1]$ is exactly $w'(S)/q$ where w' is the total query weight of S in $\mathcal{B}^{f'}$. By Lemma 3 we know that $|w(S)/q - w'(S)/q| \leq O(\sqrt{q\epsilon})$, meaning $w'(S) \leq w(S) + q \times O(\sqrt{q\epsilon}) = O(\sqrt{q^3\epsilon})$ □

Almost Projective Measurements. We state a property of general measurements due to [Zha20] that captures when a measurement is "close" to being projective, in the sense that sequential applications of the measurement yield similar outcomes.

Definition 5. *A real-valued measurement $\mathbf{M} = (M_p)_p$ is (ϵ, δ)-*almost projective *if applying $\mathbf{M}$ twice in a row to a register $\mathcal{H}$ (initially containing any state $\boldsymbol{\rho}$) produces measurement outcomes p, p' where $\Pr[|p - p'| \leq \epsilon] \geq 1 - \delta$.*

Quantum State Repair. We now recall quantum state repair from [CMSZ22].

Lemma 6 ([CMSZ22], Lemma 4.10)**.** *Given a projective measurement $\mathbf{P}$ on register $\mathcal{H}$ that has outcomes in set S of size N, an (ϵ, δ)-almost projective measurement $\mathbf{M}$ on $\mathcal{H}$, and $T \in \mathbb{N}, s \in S, p \in [0,1]$, there exists a procedure $\mathsf{QRepair}^{\mathbf{M},\mathbf{P}}_{T,p,s}$ on $\mathcal{H}$ such that:*

- *(State is repaired with respect to $\mathbf{M}$) Consider applying the following operations to register $\mathcal{H}$ initially containing state $\boldsymbol{\rho}$:*

  1. *First apply $\mathbf{M}$ to obtain $p \in [0,1]$,*
  2. *Then apply $\mathbf{P}$ to obtain outcome $s \in S$,*
  3. *Then apply $\mathsf{QRepair}^{\mathbf{M},\mathbf{P}}_{T,p,s}$,*
  4. *And finally, apply $\mathbf{M}$ once more to obtain $p' \in [0,1]$.*

  Then $\Pr[|p - p'| > 2\epsilon] \leq N\delta + N/T + 4\sqrt{\delta}$.

- *(Efficiency) The expected total number of calls that $\mathsf{QRepair}$ makes to $\mathbf{M}$ and $\mathbf{P}$ is at most $N + 4T\sqrt{\delta}$.*

In other words, since $\mathbf{M}$ is almost projective, applying $\mathbf{M}$ twice in a row will give outcomes p, p' that are close. However, if $\mathbf{P}$ is applied in between these two measurements, there are no more guarantees on the closeness of p, p'. However, by applying $\mathsf{QRepair}$ before the second application of $\mathbf{M}$, we can once again ensure closeness of p, p'.

Mixtures of Projective Measurements. The following is taken from [Zha20]. We consider the following abstract setup. We have a collection $\mathcal{P} = \{\mathcal{P}_i\}_{i\in\mathcal{I}}$ of binary outcome projective measurements $\mathcal{P}_i = (P_i, Q_i)$ over the same Hilbert space $\mathcal{H}$. Here, P_i corresponds to output 0, and Q_i corresponds to output 1. We will assume we can efficiently measure the $\mathcal{P}_i$ for superpositions of i, meaning we can efficiently perform the following projective measurement over $\mathcal{I} \otimes \mathcal{H}$:

$$\left(\sum_i |i\rangle\langle i| \otimes P_i \ , \ \sum_i |i\rangle\langle i| \otimes Q_i \right) \tag{1}$$

Here, we call $\mathcal{P}$ a *collection of projective measurements*, and call $\mathcal{I}$ the *control*. For a distribution D over $\mathcal{I}$, let $\mathcal{P}_D$ be the POVM which samples a random $i \leftarrow D$, applies the measurement $\mathcal{P}_i$, and outputs the resulting bit. We call $\mathcal{P}_D$ a *mixture of projective measurements*. The POVM is given by the matrices (P_D, Q_D) where

$$P = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow D] P_i \quad \text{and} \quad Q = \sum_{i\in\mathcal{I}} \Pr[i \leftarrow D] Q_i$$

Next, for $a \in \mathbb{R}$ and interval $[b, c] \subseteq \mathbb{R}$, denote the distance between a and $[b, c]$ as $|a - [b, c]| := \min_{x\in[b,c]} |a - x|$. For $a \in [b, c]$, the distance is 0 and for $a \notin [b, c]$, the distance is $\max(a - c, b - a)$. Let D_0, D_1 be two distributions over $\mathbb{R}$, with cumulative density functions f_0, f_1, respectively. Let $\epsilon \in \mathbb{R}$. The Shift distance with parameter ϵ is defined as:

$$\Delta_\epsilon(D_0, D_1) := \sup_{x\in\mathbb{R}} \left| f_0(x) - [f_1(x - \epsilon), f_1(x + \epsilon)] \right|$$

Let $\mathbf{M} = (M_i)_{i\in\mathcal{I}}$ and $\mathbf{N} = (N_j)_{j\in\mathcal{J}}$ be real-valued quantum measurements over the same quantum system $\mathcal{H}$. The shift distance between $\mathbf{M}, \mathbf{N}$, denoted $\Delta_\epsilon(\mathbf{M}, \mathbf{N})$ is defined as

$$\Delta_\epsilon(\mathbf{M}, \mathbf{N}) := \sup_{|\psi\rangle} \Delta_\epsilon(\ \mathbf{M}(|\psi\rangle)\ ,\ \mathbf{N}(|\psi\rangle)\ )$$

Now, if $\mathcal{P}_D = (P_D, Q_D)$ is a mixture of projective measurements, we note that $Q_D = \mathbf{I} - P_D$, and therefore P_D, Q_D commute. In this case, $\mathcal{P}_D$ has a *projective implementation*, denoted $\mathsf{ProjImp}(\mathcal{P}_D)$, which is defined as follows. Let S be the set of eigenvalues of P_D, and R_i for i the projectors onto the associated eigenspaces. Then $\mathsf{ProjImp}(\mathcal{P}_D)$ is the projective measurement $(P_i)_{i\in S}$. Note that $S \subseteq [0, 1]$. Also note that applying $\mathcal{P}_D$ is equivalent to the following: first apply $\mathsf{ProjImp}(\mathcal{P}_D)$ to obtain outcome p, then interpret p as a probability and output 1 with probability p.

Lemma 7 ([Zha20], Theorem 6.2)**.** *For any $\epsilon, \delta, \mathcal{P}, D$, there exists an algorithm $\mathsf{API}^{\mathcal{P},D}_{\epsilon,\delta}$ operating on $\mathcal{H}$ and making quantum queries to $\mathcal{P}, D$ which additionally outputs a number in some set $S \subseteq [0, 1]$ such that:*

- *There is a function $R = \mathsf{poly}(1/\epsilon, \log(1/\delta))$ such that the expected number of calls $\mathsf{API}^{\mathcal{P},D}_{\epsilon,\delta}$ makes to $\mathcal{P}$ and D, the running time of $\mathsf{API}^{\mathcal{P},D}_{\epsilon,\delta}$, and $|S|$ are all bounded by R.*
- *$\mathsf{API}^{\mathcal{P},D}_{\epsilon,\delta}$ is (ϵ, δ)-almost projective.*
- *$\Delta_\epsilon(\mathsf{API}^{\mathcal{P},D}_{\epsilon,\delta}, \mathsf{ProjImp}(\mathcal{P}_D)) \leq \delta$.*

4.2 Cryptographic Notions

Let λ be a security parameter. All quantities below will be implicit functions of λ. When we say "polynomial" or "negligible", we mean polynomial or negligible in λ.

4.2.1 Quantum Money

We now define what a quantum money scheme. As shown by [AC12], it suffices to consider "mini-schemes", where there is only a single banknote. A quantum money scheme is a pair of algorithms:

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

Correctness. We define correctness. Usually one considers negligible correctness error, but in this work it will be convenient to consider relaxed forms of correctness where an honest banknote may fail to verify with non-negligible but still bounded probability.

Definition 8. *A quantum money scheme* $(\mathsf{Gen}, \mathsf{Ver})$ *is* c-correct *if* $\Pr[\mathsf{Ver}(\mathsf{Gen}())] \geq c$. $(\mathsf{Gen}, \mathsf{Ver})$ *is* correct *if it is c-correct for some c such that $1 - c$ is negligible.*

Security. We now define security. Usually one considers negligible security, but in this work it will be convenient to consider relaxed forms of security where an adversary may forge with non-negligible but still bounded probability.

Definition 9. *A quantum money scheme* $(\mathsf{Gen}, \mathsf{Ver})$ *is ϵ-secure if, for all quantum polynomial-time oracle algorithms $\mathcal{A}$, $\mathcal{A}$ wins the following game with probability at most ϵ:*

- *Run* $(\sigma, \$) \leftarrow \mathsf{Gen}()$.
- *Run* $\rho_{1,2} \leftarrow \mathcal{A}(\sigma, \$)$, *where $\rho_{1,2}$ is a joint state over registers $\mathcal{H}_1, \mathcal{H}_2$.*
- *Apply* $\mathsf{Ver}(\sigma, \cdot)$ *twice, once each to $\mathcal{H}_1, \mathcal{H}_2$, to get bits b_1, b_2.*
- *$\mathcal{A}$ wins if $b_1 = b_2 = 1$.*

$(\mathsf{Gen}, \mathsf{Ver})$ *is* secure *if it is ϵ-secure for some negligible ϵ.*

4.2.2 Evasive Obfuscation

We now define evasive obfuscation. There are many possible notions of security for evasive obfuscation, as explored by [BBC+14]. Our separation will apply to any reasonable notion, but for concreteness we focus on only one definition, a composable and auxiliary-input version of input-hiding obfuscation. We first define evasive circuits.

Let $\mathcal{C}$ be a family of circuits with domain $\{0,1\}^n$ and range $\{0,1\}$ such that $\log |\mathcal{C}|$ is polynomial.

Definition 10 (Evasive Distribution of Circuit Tuples)**.** *A distribution $\mathcal{D}$ over $\mathcal{C}^s \times \{0,1\}^*$ is* evasive *with auxiliary input if, for any QPT $\mathcal{A}$, there exists a negligible function* negl *such that* $\Pr[\exists i, C_i(\mathcal{A}(\mathsf{aux})) = 1 : (C_1, \cdots, C_s, \mathsf{aux}) \leftarrow \mathcal{D}] \leq \mathsf{negl}$.

Definition 11. *A composable input-hiding obfuscator for the class of circuits $\mathcal{C}$ is a pair of probabilistic polynomial time algorithms* $\mathsf{Obf}, \mathsf{Eval}$ *with the following properties:*

- **Correctness.** *There exists a negligible function* negl *such that, for any circuit $C \in \mathcal{C}$ and any input $x \in \{0,1\}^n$, $\Pr[\mathsf{Eval}(\hat{C}, x) = C(x) : \hat{C} \leftarrow \mathsf{Obf}(C)] \geq 1 - \mathsf{negl}$, where the probability is over the random choice of C and the randomness of* Obf*,*

- **Composable Input-hiding (with auxiliary input).** *For any evasive distribution over circuit tuples $\mathcal{D}$, any QPT oracle-aided adversary $\mathcal{A}$, and any polynomial ℓ, there exists a negligible* negl *such that for any $i \in [s]$, $\Pr\left[C_i(\mathcal{A}(\hat{C}_1, \cdots, \hat{C}_s, \mathsf{aux})) = 1 : \substack{(C_1, \cdots, C_s, \mathsf{aux}) \leftarrow \mathcal{D} \\ \hat{C}_j \leftarrow \mathsf{Obf}(C_j) \text{ for } j \in [s]}\right] \leq$* negl.

The usual notion of input-hiding evasive obfuscation corresponds to the restriction to $s = 1$ and aux being empty. Thus our notion is a significant generalization. This makes our separation stronger.

5 The Model

Let $n = n(\lambda), t = t(\lambda), w = w(\lambda)$ be polynomials, and let $\mathcal{C} = (\mathcal{C}_\lambda)_\lambda$ be any family of circuits such that $\log |\mathcal{C}_\lambda|$ is polynomial. From now on, we will drop the security parameter λ, leaving it implicit as with our cryptographic definitions. Assume $w \geq t + \log_2 |\mathcal{C}|$. Consider the following oracles:

- $\mathcal{O} : \mathcal{C} \times \{0,1\}^t \to \{0,1\}^w$ is a random injection. Here, $\mathcal{O}$ will play the role of obfuscator, with the input $r \in \{0,1\}^t$ being the random coins of the obfuscator. The outputs $\hat{C} \in \{0,1\}^w$ will therefore play the role of obfuscated program

- $\mathcal{E} : \{0,1\}^w \times \{0,1\}^n \to \mathcal{C} \cup \{\bot\}$ for special symbol $\bot$. $\mathcal{E}$ is defined defined as follows. On input $(\hat{C}, x) \in \{0,1\}^w \times \{0,1\}^n$, first check if $\hat{C}$ is in the image of $\mathcal{O}$. If not, output $\bot$. Otherwise, run $(C, r) \leftarrow \mathcal{O}^{-1}(\hat{C})$ and compute $b = C(x)$. If $b = 1$, output C; otherwise output $\bot$. $\mathcal{E}$ plays the role of evaluation algorithm for the obfuscator. Note that the value of b can be inferred from the output of $\mathcal{E}$.

Definition 12. *A quantum polynomial time (QPT) algorithm in our model is a quantum oracle algorithm $\mathcal{A}^{\mathcal{O},\mathcal{E}}$ that makes polynomially-many queries to $\mathcal{O}, \mathcal{E}$, but may have unbounded computation outside the oracle.*

$\mathcal{O}, \mathcal{E}$ give evasive obfuscation for the class $\mathcal{C}$. We discuss how the oracles above give obfuscation for evasive circuits. The oracles will give evasive obfuscation for any meaningful notion, but for concreteness we show that they give our notion of composable input-hiding.

Theorem 13. *For any t, w such that $t = \omega(\log \lambda)$ and $w \geq t + \log_2 |\mathcal{C}| + \omega(\log \lambda)$, the obfuscator* $\mathsf{Obf} = \mathcal{O}, \mathsf{Eval} = \mathcal{E}$ *is a composable input-hiding obfuscator for $\mathcal{C}$.*

Proof. Let $\mathcal{D}$ be an evasive distribution of circuit tuples. Let $\mathcal{A}$ be a potential adversary against $\mathsf{Obf}, \mathsf{Eval}$ relative to $\mathcal{D}$. Let ϵ be the probability $\mathcal{A}$ wins the composable input-hiding experiment. We consider the following experiments on $\mathcal{A}$:

- **Experiment 1.** Sample $\mathcal{O}, \mathcal{E}$. Sample $(C_1, \ldots, C_s, \mathsf{aux}) \leftarrow \mathcal{D}$. Then run $\hat{C}_j \leftarrow \mathsf{Obf}(C_j) = \mathcal{O}(C_j, r_j)$ for $j \in [t]$. where $r_j \in \{0,1\}^t$ are uniform. Now run $x \leftarrow \mathcal{A}^{\mathcal{O},\mathcal{E}}(\hat{C}_1, \cdots, \hat{C}_s, \mathsf{aux})$. The experiment outputs 1 if $C_i(x) = 1$ for any $i \in [s]$; that is, if $\mathcal{A}$ wins the composable input-hiding experiment. Thus the probability Experiment 1 outputs 1 is $\epsilon_1 = \epsilon$.

- **Experiment 2.** This is identical to Experiment 1, except we condition on the r_j being distinct. Let ϵ_2 be the probability the experiment outputs 1. Note that the r_j are distinct except with probability $O(s^2/2^t)$. Thus, $|\epsilon_1 - \epsilon_2| \leq O(s^2/2^t) = \mathsf{negl}$.

- **Experiment 3.** Here, we delay the sampling of $\mathcal{O}, \mathcal{E}$. Sample $(C_1, \ldots, C_s) \leftarrow \mathcal{D}$. Then sample uniform distinct $r_1, \ldots, r_j$ and uniform distinct $\hat{C}_j \leftarrow \{0,1\}^w$ for each $j \in [s]$. Only now do we sample $\mathcal{O}, \mathcal{E}$: we sample $\mathcal{O}$ as a random injection conditioned on $\hat{C}_j = \mathcal{O}(C_j, r_j)$, and define $\mathcal{E}$ accordingly. Finally we run $x \leftarrow \mathcal{A}^{\mathcal{O},\mathcal{E}}(\hat{C}_1, \cdots, \hat{C}_s, \mathsf{aux})$. The experiment outputs 1 if $C_i(x) = 1$ for any $i \in [s]$. Let ϵ_3 be the probability Experiment 3 outputs 1. Experiments 2 and 3 are identically distributed, so $\epsilon_3 = \epsilon_2$.

  Note that in Experiment 3, we can equivalently sample $\mathcal{O}, \mathcal{E}$ as follows. First sample a uniform distinct $r_j \in \{0,1\}^t$ and $\hat{C}_j \in \{0,1\}^w$. Then sample random injective $\mathcal{O}' : (\mathcal{C} \times \{0,1\}^t) \setminus \{(C_j, r_j)\}_j \to \{0,1\}^w \setminus \{\hat{C}_j\}_j$, and then define

$$\mathcal{O}(C,r) = \begin{cases} \hat{C}_j & \text{if } (C,r) = (C_j, r_j) \\ \mathcal{O}'(C,r) & \text{otherwise} \end{cases}$$

$$\mathcal{E}(\hat{C}, x) = \begin{cases} (C_j, r_j) & \text{if } \hat{C} = \hat{C}_j \text{ for some } j \text{ and } C_j(x) = 1 \\ (C, r) & \text{if } \hat{C} \text{ is in the image of } \mathcal{O}',\ (C,r) = (\mathcal{O}')^{-1}(\hat{C}), \text{ and } C(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

- **Experiment 4.** This is identical to our alternate view of Experiment 3, except that $\mathcal{O}'$ is sampled as a random injection from $\mathcal{O}' : (\mathcal{C} \times \{0,1\}^t) \setminus \{(C_j, r_j)\}_j \to \{0,1\}^w$. Let ϵ_4 be the probability Experiment 3 outputs 1. We see that the distribution of each output of $\mathcal{O}'$ is changed from uniform over $\{0,1\}^w \setminus \{\hat{C}_j\}_j$ to uniform over $\{0,1\}^w$. The statistical distance between these distributions is $s/2^w$. Then a union bound over all $|\mathcal{C}| \times 2^t$ gives a distance of $|\epsilon_3 - \epsilon_2| \leq s|\mathcal{C}|2^t/2^w$. This is negligible by our assumption that $w \geq t + \log_2 |\mathcal{C}| + \omega(\log \lambda)$, since s is polynomial. Thus, $|\epsilon_4 - \epsilon_3| \leq \mathsf{negl}$.

  Note that in Experiment 4, we might as well sample $\mathcal{O}'$ as a random injection on $\mathcal{O}' : \mathcal{C} \times \{0,1\}^t \to \{0,1\}^w$, since the values of $\mathcal{O}'$ on $\{(C_j, r_j)\}_j$ are ignored.

- **Experiment 5.** This is identical to Experiment 4, except that we define $\mathcal{O}, \mathcal{E}$ in terms of $\mathcal{O}'$ as follows:

$$\mathcal{O} = \mathcal{O}'$$

$$\mathcal{E}(\hat{C}, x) = \begin{cases} \perp & \text{if } \hat{C} = \hat{C}_j \text{ for some } j \\ (C, r) & \text{if } \hat{C} \text{ is in the image of } \mathcal{O}',\ (C,r) = (\mathcal{O}')^{-1}(\hat{C}), \text{ and } C(x) = 1 \\ \perp & \text{otherwise} \end{cases}$$

  Let ϵ_5 be the probability Experiment 5 outputs 1. Notice that the only differences between Experiment 4 and Experiment 5 are the following:

- We changed $\mathcal{E}$ on inputs $(\hat{C}, x)$ such that both (1) $\hat{C} = \hat{C}_j$ for some j, corresponding to program C_j, and (2) $C_j(x) = 1$.
- We changed $\mathcal{O}$ on inputs (C_j, r_j)

Observe that, in Experiment 5, the oracles are independent of the C_j. This allows us to conclude the ϵ_5 is negligible. Indeed, we can construct an algorithm $\mathcal{A}'(\mathsf{aux})$ which is given aux sampled from $(C_1, \dots, C_s, \mathsf{aux}) \leftarrow \mathcal{D}$. $\mathcal{A}'(\mathsf{aux})$ runs $\mathcal{A}$, simulating the view of $\mathcal{A}$ in Experiment 5. Then it outputs whatever $\mathcal{A}$ outputs. Then $\mathcal{A}'$ outputs an x such that $C_j(x) = 1$ for some j with probability exactly ϵ_5. But this probability must be negligible by the evasiveness of $\mathcal{D}$.

It remains to prove that $|\epsilon_4 - \epsilon_5|$ is negligible, which would then allow us to conclude ϵ is negligible, a contradiction. We prove this by showing that the total query weight by $\mathcal{A}$ in Experiment 5 on inputs where Experiment 4 and Experiment 5 differ is negligible; by Lemma 3 this means $|\epsilon_4 - \epsilon_5|$ is negligible as desired.

We must handle two different kinds of differing points.

- Queries to $\mathcal{O}$ on inputs (C_j, r_j). Notice that in Experiment 5, the view of $\mathcal{A}$ is independent of the r_j. Since the r_j are uniform in a super-polynomial-sized set, this means the total expected query weight in Experiment 5 on points (C_j, r_j) is negligible.
- Queries to $\mathcal{E}$ on inputs $(\hat{C}, x)$ such that both (1) $\hat{C} = \hat{C}_j$ for some j, corresponding to program C_j, and (2) $C_j(x) = 1$. We prove the query weight on such points is negligible in Experiment 5 by constructing a different adversary $\mathcal{A}''(\mathsf{aux})$ which is given aux sampled from $(C_1, \dots, C_s, \mathsf{aux}) \leftarrow \mathcal{D}$. $\mathcal{A}''(\mathsf{aux})$ chooses a random $i \in [Q]$, where Q is the total number of queries $\mathcal{A}$ makes to $\mathcal{E}$. Then it runs $\mathcal{A}$, simulating the view of $\mathcal{A}$ in Experiment 5 up until $\mathcal{A}$ makes the ith query. It then measures the query, obtaining $(\hat{C}, x)$; it outputs x.

  The probability $C_j(x) = 1$ for some j is exactly $1/Q$ times the total query weight on such x. By the evasiveness of $\mathcal{D}$, this probability must be negligible. The query weight must therefore also be negligible since Q is a polynomial.

This completes the proof of Theorem 13. □

Remark 3. *The requirement that $t = \omega(\log \lambda)$ is necessary, as otherwise the adversary can detect whether two programs are equal by testing if their obfuscated versions are equal. Such an equality pattern could reveal an accepting input that would otherwise not be available just given* aux.

However, if one restricts to the case of a single program ($s = 1$), then t can be taken to be 0. We can also consider the class $\mathcal{C}$ of point functions I_x. In this case, the oracle $\mathcal{E}$ can be simulated just given $\mathcal{O}$: to test if $\hat{C}(x) = 1$, simply check if $\mathcal{O}(I_x) = \hat{C}$. We are left with just a random injection $\mathcal{O}$, which acts as an injective one-way function. Thus, we see that our model of evasive obfuscation also captures (injective) one-way functions.

6 The Canonical Verifier

Here, we discuss what we call a canonical verifier for a quantum money scheme. The goal is to make the verifier almost projective, while allowing for the measurements of various queries to be projective (the latter feature to be discussed in Section 7). Along the way, we give a new tool for amplifying the correctness of a quantum money scheme, which may result in smaller banknotes than the straightforward approach of parallel repetition.

Assume a c-correct quantum money scheme $\mathsf{Gen}, \mathsf{Ver}$. Let $\mathcal{H}$ be the register of the quantum money state. For a serial number σ, let Ver_σ be the measurement corresponding to applying $\mathsf{Ver}(\sigma, \cdot)$ to $\mathcal{H}$. By dilation, we can make Ver_σ projective over the joint system $\mathcal{H} \otimes \mathcal{Z}$, where $\mathcal{Z}$ is initialized to $|0\rangle$. Throughout we will let \$ refer to the state over $\mathcal{H}$ produced by Gen and $\$'$ the state over the joint system $\mathcal{H} \times \mathcal{Z}$. Let Z be the 1-dimensional projective measurement on $\mathcal{Z}$ which accepts $|0\rangle$. We will slightly abuse notation, and also consider Ver_σ, Z as being projections on the joint system $\mathcal{H} \times \mathcal{Z}$.

Fix parameters $p, \epsilon, \delta, \gamma \in [0, 1]$. Here, p, ϵ, γ will be inverse polynomials; the precise relationships will be discussed in Section 7, which will be based in part on what we develop here. δ is negligible. Let $\mathcal{P}(\sigma) = \{\mathsf{Ver}_\sigma, Z\}$ be the set of projective measurements on $\mathcal{H} \otimes \mathcal{Z}$, and D_p the distribution over $\mathcal{P}$ which selects Ver_σ with probability p and Z with probability $1-p$. We define a new verifier for the quantum money scheme:

Algorithm 14 (Canonical Verifier)**.** *Let* $\mathsf{Ver}_{*,\sigma,p,\epsilon,\delta} = \mathsf{API}_{\epsilon,\delta}^{\mathcal{P}(\sigma),D_p}$ *which operates on* $\mathcal{H} \otimes \mathcal{Z}$*, and define* $\mathsf{Ver}_*(\sigma, \$ \mid c, p, \epsilon, \delta, \gamma)$ *to be the following algorithm:*

1. *Apply* $\mathsf{Ver}_{*,\sigma,p,\epsilon,\delta}$ *to* $\mathcal{H} \otimes \mathcal{Z}$*, obtaining outcome* R
2. *Accept if and only if* $R \geq 1 - p(1-c) - \gamma - \epsilon$

We also define a new note generator:

Algorithm 15 (Canonical Gen)**.** *Define* $\mathsf{Gen}_*(\mid c, p, \epsilon, \delta, \gamma)$ *to be the following algorithm:*

1. *Run* $(\sigma, \$) \leftarrow \mathsf{Gen}()$*. Let* $\mathcal{H}$ *be the register containing* \$*. As above, we will initialize a register* $\mathcal{Z}$ *with* $|0\rangle$*.*
2. *Apply* $\mathsf{Ver}_*(\sigma, \cdot \mid c, p, \epsilon, \delta, \gamma)$ *to* $\mathcal{H} \otimes \mathcal{Z}$*, letting* $\$'$ *be the state remaining in* $\mathcal{H} \otimes \mathcal{Z}$*.*
3. *If* Ver_* *accepts, output* $(\sigma, \$')$*. Otherwise discard* $(\sigma, \$')$ *and go to Step 1.*

In the rest of this section, we prove some general properties of $\mathsf{Gen}_*, \mathsf{Ver}_*$.

Correctness. We first consider the quantum money scheme $(\mathsf{Gen}, \mathsf{Ver}_*)$. This is a slight abuse of notation, since notes from Gen lie in $\mathcal{H}$, whereas notes for Ver_* lie in $\mathcal{H} \otimes \mathcal{Z}$. When considering such a scheme, we therefore interpret Gen as initializing $\mathcal{Z}$ to $|0\rangle$, and outputting $\$' = \$ \otimes |0\rangle \in \mathcal{H} \otimes \mathcal{Z}$ as the note.

Lemma 16. *If* $\mathsf{Gen}, \mathsf{Ver}$ *is* c*-correct, then* $\mathsf{Gen}, \mathsf{Ver}_*(\cdot, \cdot \mid c, p, \epsilon, \delta, \gamma)$ *is* c_**-correct, where* $c_* = \frac{\gamma}{p(1-c)+\gamma} - \delta$*.*

Proof. Consider sampling $(\sigma, \$) \leftarrow \mathsf{Gen}()$ and letting $\$' = \$ \otimes |0\rangle \in \mathcal{H} \otimes \mathcal{Z}$. Since $\mathsf{Gen}, \mathsf{Ver}$ is c-correct, we know that applying Ver_σ to $\$'$ will accept with probability at least c (over the randomness of both Gen and Ver_σ). Applying Z will accept with probability 1. Therefore, the mixture $\mathcal{P}_{D_p}(\sigma)$ will accept with probability at least $(1-p) + pc = 1 - p(1-c)$.

Now consider applying $\mathsf{ProjImp}(\mathcal{P}_{D_p}(\sigma))$ to $\$'$, resulting in outcome R. We know that $\mathbb{E}[R]$ is equal to the probability $\mathcal{P}_{D_p}(\sigma)$ accepts, and so $\mathbb{E}[R] \geq 1 - p(1-c)$. Since $R \leq 1$, we have that $1 - R \geq 0$ and $\mathbb{E}[1-R] \leq p(1-c)$. By Markov's inequality, we therefore have that $\Pr[1 - R > p(1-c) + \gamma] \leq \frac{p(1-c)}{p(1-c)+\gamma}$. Equivalently, $\Pr[R \geq 1 - p(1-c) - \gamma] \geq \gamma / (p(1-c) + \gamma)$.

Now we imagine computing R from $\mathsf{API}_{\epsilon,\delta}^{\mathcal{P}(\sigma),D_p}$ as in Ver_*. By Lemma 7, we conclude that $\Pr[R \geq 1 - p(1-c) - \gamma - \epsilon] \geq \gamma/(p(1-c)+\gamma) - \delta = c_*$, and thus Ver_* accepts with at last this probability. □

Lemma 17. *Let T be the combined run time of $\mathsf{Gen}, \mathsf{Ver}_*$. If $\mathsf{Gen}, \mathsf{Ver}$ is c-correct, then the expected running time of $\mathsf{Gen}_*(\ | \ c, p, \epsilon, \delta, \gamma)$ is at most $T/\left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right)$. Moreover, $\mathsf{Gen}_*(\ | \ c, p, \epsilon, \delta, \gamma), \mathsf{Ver}_*(\cdot, \cdot \mid c, p, \epsilon, \delta, \gamma + \epsilon)$ is $(1-\delta)$-correct.*

Proof. Each trial of Gen_* takes time T, and succeeds with probability at least $c_* = \frac{\gamma}{p(1-c)+\gamma} - \delta$. This proves the running time of Gen_*. Then $(1-\delta)$-correctness follows from the fact that $\mathsf{Ver}_{*,\sigma,p,\epsilon,\delta}$ is (ϵ, δ)-almost projective. □

Security. We first need the following technical lemma characterizing the kind of states accepted by Ver_*:

Lemma 18. *Suppose $p/(1-p) \leq c/3$ and $2\epsilon + \gamma + \delta \leq pc/3$. Consider applying $\mathsf{Ver}_*(\sigma, \cdot \mid c, p, \epsilon, \delta, \gamma)$ to a quantum state over $\mathcal{H} \otimes \mathcal{Z}$, and post-select on acceptance. Let \$ be the resulting state contained in $\mathcal{H} \otimes \mathcal{Z}$. Then $\mathsf{Ver}(\sigma, \$)$ accepts with probability at least $W \geq c/9$.*

Proof. Let R be the outcome of the measurement $\mathsf{Ver}_{*,\sigma,p,\epsilon,\delta}$ applied during running Ver_*. By definition, since Ver_* accepted, we have $R \geq 1 - p(1-c) - \gamma - \epsilon$. Now let v be be the probability Ver_σ accepts \$, and z the probability Z accepts \$.

Now consider applying $\mathsf{ProjImp}(\mathcal{P}_{D_p}(\sigma))$ to a state $\$'$ accepted by Ver_*, obtaining outcome R'. By Lemma 7, we know that $\Pr[|R' - R| \geq \epsilon] \leq \delta$. Therefore, we have that $R' \geq 1 - p(1-c) - \gamma - 2\epsilon$, except with probability δ. Since R' must be non-negative, this implies that $\mathbb{E}[R'] \geq (1 - p(1-c) - \gamma - 2\epsilon)(1-\delta) \geq 1 - p(1-c) - \gamma - 2\epsilon - \delta$.

On the other hand, know that $\mathbb{E}[R'] = pv + (1-p)z$ by the definition of v, z. In particular, this means $z \in [1 - \frac{p(1-c)+\gamma+2\epsilon+\delta}{1-p}, 1]$. Now consider the state $\$''$, which is the normalized state $Z\$'$ resulting from applying Z to $\$'$ and then post-selecting on acceptance. Let v' be the probability Ver_σ accepts $\$''$. Then by gentle measurement $d(\$', \$'') \leq 2\sqrt{1-z}$, meaning $|v - v'| \leq 2\sqrt{1-z}$. This means $\$''$ is accepted by Ver_σ with probability at least

$$\begin{aligned} v - 2\sqrt{1-z} &= z + (\mathbb{E}[R'] - z)/p - 2\sqrt{1-z} \\ &\geq z + \frac{1 - z - p(1-c) - \gamma - 2\epsilon - \delta}{p} - 2\sqrt{1-z} := f(z) \end{aligned}$$

We now observe that $f(z)$ is convex (its second derivative is $(1-z)^{-3/2}/2$). Moreover, its first derivative vanishes when $z = (1-2p)/(1-p)^2$, meaning $f(z)$ is minimized at this point. Thus, $\$''$ is accepted by Ver_σ with probability at least

$$f((1-2p)/(1-p)^2) = c - \frac{p}{1-p} - \frac{2\epsilon + \gamma + \delta}{p} \tag{2}$$

Now consider the following operation on $\$'$. We discard the $\mathcal{Z}$ register, and then feed the resulting state ρ contained in $\mathcal{H}$ into Ver. Since Ver is equivalent to applying Ver_σ on $\mathcal{H}$ with a zero'd out $\mathcal{Z}$ register, we know that Ver will at least accept if (1) applying Z accepts, and then (2) subsequently applying Ver_σ accepts (it may also accept if (1) does not happen). We already computed the probability of (2) in Equation 2. Meanwhile, the probability of (1) is simply z, which is lower-bounded by $1-\frac{p(1-c)+\gamma+2\epsilon+\delta}{1-p}$. Therefore, Ver accepts ρ with probability at least

$$W := \left(1-\frac{p(1-c)+\gamma+2\epsilon+\delta}{1-p}\right)\times\left(c-\frac{p}{1-p}-\frac{\gamma+2\epsilon+\delta}{p}\right)$$

Now we use our assumption that $p/(1-p)\leq c/3$ and $2\epsilon+\gamma+\delta\leq pc/3$ to lower-bound W as

$$W\geq(1-c/3-c/9)\,(c-c/3-c/3)\geq c/9$$

This completes the proof of Lemma 18 □

Now we prove security of Ver_*.

Lemma 19. *If* $\mathsf{Gen},\mathsf{Ver}$ *is* α*-secure and* $p/(1-p)\leq c/3$ *and* $2\epsilon+\gamma+\delta\leq pc/3$*, then the pair* $\mathsf{Gen},\mathsf{Ver}_*(\cdot,\cdot\mid c,p,\epsilon,\delta,\gamma)$ *is* α_**-secure, where* $\alpha_*=81\alpha/c^2$*.*

Proof. Consider any adversary $\mathcal{A}_*$ for the α_*-security of Ver_* . We construct an adversary $\mathcal{A}$ for the α-security of Ver. $\mathcal{A}$ simply runs $\mathcal{A}_*$ to get two (potentially entangled) quantum banknotes $\rho^*_{1,2}$. It runs Ver_* on each of ρ^*_1,ρ^*_2. Then for each note ρ^*_1,ρ^*_2, it runs Ver_*, then discards the $\mathcal{Z}$ register, and outputs the two $\mathcal{H}$ registers as $\rho_{1,2}$.

The probability ρ^*_1,ρ^*_2 both pass Ver_* is greater than α_* by assumption. Then each of the resulting $\mathcal{H}$ registers pass Ver with probability at least $c/9$ by Lemma 18. Overall, then, $\mathcal{A}$ produces two accepting banknotes with probability greater than $\alpha_*(c/9)^2=\alpha$, breaking the α-security of Ver.

Independence. One may be concerned that the events that the two H registers pass verification are correlated, meaning though each passes Ver with probability $c/9$, both simultaneously pass verification with probability somewhat larger than $(c/9)^2$. Indeed, the two events are correlated. However, while the events are correlated, it is straightforward to show that $\mathcal{A}$ still succeeds with probability α. Toward that end, consider the following operations:

- O_b for $b\in\{1,2\}$: the measurement Ver_* that $\mathcal{A}$ applies to ρ^*_b produced by $\mathcal{A}_*$.
- N_b: the measurement Ver that the experiment performs on ρ_b produced by $\mathcal{A}$

Note that O_1,O_2 commute as do O_1,N_2 and O_2,N_1, since the elements in each pair act on different registers. Thus we can shuffle the order without affecting the outcome probabilities; the only constraint is that N_b must come after O_b. So consider applying O_2,O_1,N_1 in that order, and only proceeding if the previous measurement accepts. We know that O_2,O_1 accept jointly with probability more than α_*. But now we have a state that was just accepted by O_1 and gets fed into N_1; we know via the above analysis that N_1 accepts this state with probability at least W. Thus, O_2,O_1,N_1 all jointly accept with probability at least αW.

Next consider applying O_1,N_1,O_2,N_2 in that order. By commutativity and what we just analyzed, O_1,N_1,O_2 jointly accept with probability at least αW. But now we have a state that was just accepted by O_2 and gets fed into N_2; we know that N_2 accepts this state with probability at least W. Thus O_1,N_1,O_2,N_2 jointly accept with probability at least αW^2. This completes the proof of Lemma 19. □

Lemma 20. *If* $\mathsf{Gen}, \mathsf{Ver}$ *is* α*-secure and* $p/(1-p) \leq c/3$ *and* $2\epsilon + \gamma + \delta \leq pc/3$*, then* $\mathsf{Gen}_*(\ |\ c, p, \epsilon, \delta, \gamma), \mathsf{Ver}_*(\cdot, \cdot \mid c, p, \epsilon, \delta, \gamma + \epsilon)$ *is* α_{**}*-secure, where* $\alpha_{**} = \alpha_* \times \left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right)^{-1} = (81\alpha/c^2) \times \left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right)^{-1}$.

Proof. Let $\mathcal{A}_{**}$ be an adversary breaking the α_{**}-security of $\mathsf{Gen}_*, \mathsf{Ver}_*$. We will construct a new adversary $\mathcal{A}_*$ for $\mathsf{Gen}, \mathsf{Ver}_*$. $\mathcal{A}_*(\sigma, \$)$ does the following:

1. Using our abuse of notation, we will interpret \$ as a state over the joint system $\mathcal{H} \otimes \mathcal{Z}$, where $\mathcal{Z}$ is initialized to $|0\rangle$.

2. Apply $\mathsf{Ver}_*(\sigma, \cdot \mid c, p, \epsilon, \delta, \gamma)$ to $\mathcal{H} \otimes \mathcal{Z}$. If it rejects, abort and output $\perp$.

3. Otherwise, apply $\mathcal{A}_{**}(\sigma, \cdot)$ to $\mathcal{H} \otimes \mathcal{Z}$, obtaining state $\rho_{1,2}$. Output $\rho_{1,2}$.

By Lemma 19, the success probability of $\mathcal{A}_*$ is at most α_*. Conditioned on not rejecting, the state given to $\mathcal{A}_{**}$ is identically distributed to the state $(\sigma, \$) \leftarrow \mathsf{Gen}_*(\ |\ c, p, \epsilon, \delta, \gamma)$, meaning conditioned on not rejecting the success probability of $\mathcal{A}_*$ is more than α_{**}. On the other hand, the probability of rejection is bounded by $\left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right)$. Therefore, the overall success probability of $\mathcal{A}_*$ must be more than $\alpha_{**} \times \left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right) = \alpha_*$, a contradiction. □

In particular, if $\mathsf{Gen}, \mathsf{Ver}$ is negligibly secure, then $\mathsf{Gen}_*, \mathsf{Ver}_*$ is negligibly secure while also being $(1 - \mathsf{negl})$-correct. This gives a new approach to amplifying the correctness of a quantum money scheme. Previously, the only known approach was parallel repetition: each banknote consists of several independent notes, and the overall note is accepted as long as at least one of the constituent notes is valid. This simple correctness amplification technique, however, blows of the note size be $\omega(\log \lambda)/c$. Our solution also blows up the note size, since the verifier's workspace qubits become part of the note. But for schemes with very low correctness and/or little workspace qubits, our approach will yield smaller banknotes. Certainly the serial numbers will be shorter with our approach.

7 Impossibility for Respecting Quantum Money

Here, we prove that there is no black box construction of a respecting quantum money scheme from evasive obfuscation.

Quantum Money Relative to Evasive Obfuscation Oracles. It is straightforward to adapt the definition of quantum money (Definition 9) to a relativized model where $\mathsf{Gen}, \mathsf{Ver}$ and the adversary may make queries to an oracle. Likewise, the results of Section 6 are also easily adapted to a relativized model. From this point forward, we will consider the case where $\mathsf{Gen}, \mathsf{Ver}$ and the adversary have oracle access to our evasive obfuscation oracles from Section 5. We now define what it means to be respecting:

Definition 21. *A quantum money scheme* $(\mathsf{Gen}, \mathsf{Ver})$ *relative to the evasive obfuscation oracles* $\mathcal{O}, \mathcal{E}$ *is* respecting *if the following are true: (1)* Gen *makes only classical queries to* $\mathcal{O}$*, and (2)* Ver *makes no queries to* $\mathcal{O}$*.*

Since the outputs of $\mathcal{O}$ are sparse, a standard argument shows that we can assume (incurring negligible error) all queries to $\mathcal{E}(\hat{C}, x)$ were of $\hat{C}$ previously outputted by $\mathcal{O}$. Since Gen "knows" all queries made to $\mathcal{O}$, it can answer all $\mathcal{E}$ queries for itself. Thus, we can additionally assume for respecting schemes that Gen makes no queries to $\mathcal{E}$.

Our main theorem. We now give the main theorem of this section and the paper:

Theorem 22. *If $w \geq t + \log_2 |\mathcal{C}| + \omega(\log \lambda)$, there does not exist a respecting quantum money scheme relative to the evasive obfuscation oracles $\mathcal{O}, \mathcal{E}$.*

The rest of this section is devoted to the proof of Theorem 22.

7.1 Measuring Queries

Before describing our attack, we describe how to use canonical verifiers to measure queries. Let $\mathsf{Gen}^{\mathcal{O},\mathcal{E}}, \mathsf{Ver}^{\mathcal{O},\mathcal{E}}$ be a quantum money scheme relative to evasive obfuscation oracles $\mathcal{O}, \mathcal{E}$. We will consider Ver taking as input a money state in system $\mathcal{H}$, and using ancilla register $\mathcal{Z}$ as work space, where $\mathcal{Z}$ is initialized to $|0\rangle$. Let Q be the number of queries that Ver makes to $\mathcal{E}$. We define projections $\Pi_{\sigma,q,f}$ on $\mathcal{H} \otimes \mathcal{Z}$ where $q \in [Q]$ and f is a function with domain $\{0,1\}^w \times \{0,1\}^n$, corresponding to the query inputs for a query to $\mathcal{E}$:

- Let $U_{\sigma,q}$ be the unitary which runs $\mathsf{Ver}(\sigma, \cdot)$ until the qth query to $\mathcal{E}$. Apply $U_{\sigma,q}$ to $\mathcal{H} \otimes \mathcal{Z}$.
- The query to $\mathcal{E}$ interprets $\mathcal{H} \otimes \mathcal{Z}$ as containing $\sum_{\hat{C},x,y,a} \mu_{\hat{C},x,y,a} |\hat{C}, x, y, a\rangle$, where $\hat{C}, x$ are the inputs to the query, the output is XORed into y, and a is the remaining part of the quantum system.

  $\Pi_{\sigma,q,f}$ does the following: construct $\sum_{\hat{C},x,y,a} \mu_{\hat{C},x,y,a} |\hat{C}, x, y, a\rangle |f(\hat{C}, x)\rangle$ by applying f in superposition to the query registers, writing the response to a new register $\mathcal{I}$.
- $\Pi_{\sigma,q,f}$ now measures $\mathcal{I}$ to obtain measurement outcome I. It discards the register $\mathcal{I}$.
- Finally, $\Pi_{\sigma,q,f}$ applies the unitary $U^\dagger_{\sigma,q}$ to $\mathcal{H} \otimes \mathcal{Z}$.

It is clear that $\Pi_{\sigma,q,f}$ is projective over $\mathcal{H} \otimes \mathcal{Z}$. This will be important for our application of state repair, since state repair [CMSZ22] requires projections.

7.2 The Attack

Let $\mathsf{Gen}, \mathsf{Ver}$ be a respecting scheme that is c-correct. Recall that a respecting scheme is one where Gen makes only classical queries to $\mathcal{O}$, and Ver makes no queries to $\mathcal{O}$. Let $\mathsf{Ver}^{\mathcal{E}}_{*,\sigma,p,\epsilon,\delta}$ be the almost projective measurement over the joint system $\mathcal{H} \otimes \mathcal{Z}$ from Section 6, and $\mathsf{Ver}^{\mathcal{E}}_{*}(\cdot, \cdot \mid c, p, \epsilon, \delta, \gamma)$ the associated canonical verifier. Let P be the number of queries Gen makes to $\mathcal{O}$ and Q be the number of queries Ver makes to $\mathcal{E}$. We now describe our attack

Algorithm 23. *Let $\mathcal{A}^{\mathcal{E}}(\sigma, \$)$ be the following algorithm:*

1. *Let $\mathcal{H}$ be the register containing $\$$, and initialize a new register $\mathcal{Z}$ to $|0\rangle$.*
2. *Initialize an empty list L.*

3. *Apply $\mathsf{Ver}^{\mathcal{E}}_{*,\sigma,p,\epsilon,\delta}$ to $\mathcal{H} \otimes \mathcal{Z}$, obtaining outcome R_0.*

4. *Let f be the function that on input $(\hat{C}, x) \in \{0,1\}^w \times \{0,1\}^n$ computes $y = \mathcal{E}(\hat{C}, x)$. If $y = C \neq \perp$, then output $(C, \hat{C})$; otherwise output $\perp$.*

5. *Let U be a parameter to be chosen later. For $i = 1, \dots, U$, do the following:*

   (a) *Choose a random $q_i \in [Q]$*

   (b) *Apply the measurement $\Pi_i = \Pi_{\sigma,q_i,f}$ to $\mathcal{H} \otimes \mathcal{Z}$, and let y_i be the outcome of the measurement.*

   (c) *Then apply $\mathsf{QRepair}^{\mathsf{Ver}^{\mathcal{E}}_{*,\sigma,p,\epsilon,\delta},\Pi_i}_{T,R_{i-1},y_i}$ where $T = \sqrt{1/\delta}$.*

   (d) *If y_i has the form $(C, \hat{C}) \neq \perp$, add $(C, \hat{C})$ to L (if it not already present).*

   (e) *Finally, run $\mathsf{Ver}^{\mathcal{E}}_{*,\sigma,p,\epsilon,\delta}$ on $\mathcal{H} \otimes \mathcal{Z}$, obtaining outcome R_i.*

6. *Let $\mathcal{E}'$ be the following function:*

$$\mathcal{E}'(\hat{C}, x) = \begin{cases} C & \textit{if there is a pair } (C, \hat{C}) \in L \textit{ and } C(x) = 1 \\ \perp & \textit{otherwise} \end{cases}$$

7. *Inefficiently construct $\$_1, \$_2$, two copies of a state ρ over $\mathcal{H}$ satisfying the following, assuming it exists:*

   - $\Pr[\mathsf{Ver}^{\mathcal{E}'}(\sigma, \rho) = 1] \geq V := c/9 - Q\sqrt{P/(U-1)}$.
   - *The query weight $\mathsf{Ver}^{\mathcal{E}'}(\sigma, \rho)$ makes on terms $(\hat{C}, x)$ such that $\hat{C}$ does not appear in L is at most $W := Q^2\sqrt{P/(U-1)}$*

8. *Output $\$_1, \$_2$.*

7.3 Analysis

We prove that $\mathcal{A}$ in Algorithm 23 succeeds with non-negligible probability. We first show that, if $|\psi\rangle$ exists, then the states outputted by $\mathcal{A}$ pass verification:

Lemma 24. *Suppose a state ρ (as in Step 7) exists, and consider running $\mathsf{Ver}^{\mathcal{E}}(\sigma, \rho)$. Then Ver accepts with probability at least $V - O(\sqrt{QW})$.*

Proof. Let S be the set of terms $(\hat{C}, x)$ such that $\hat{C}$ does not appear in L. By assumption, we have that $\Pr[\mathsf{Ver}^{\mathcal{E}'}(\sigma, \rho) = 1] \geq V$ and the total query weight on terms in S is at most W. Observe that $\mathcal{E}(\hat{C}, x) = \mathcal{E}'(\hat{C}, x)$ for all $(\hat{C}, x) \notin S$. Therefore, by Lemma 3, we have that $\Pr[\mathsf{Ver}^{\mathcal{E}}(\sigma, \rho) = 1] \geq V - O(\sqrt{QW})$. □

We now demonstrate that $|\psi\rangle$ exists with reasonable probability.

Lemma 25. *Let P be the number of queries made by Gen. Assuming $w \geq t + \log_2 |\mathcal{C}| + \omega(\log \lambda)$, then with probability at least $X = \left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right) - \left(U(2P+6)\sqrt{\delta}\right) - (\mathsf{negl})$, $R_i \geq 1 - p(1-c) - \gamma - 2(i+1)\epsilon$ for all $i \in [0, U]$*

Proof. By the $\frac{\gamma}{p(1-c)+\gamma} - \delta$ correctness of $\mathsf{Gen}, \mathsf{Ver}_*$ (Lemma 17), $R_0 \geq 1 - p(1-c) - \gamma - 2\epsilon$ except with probability at most δ. This gives the first term in the statement of Lemma 25.

Let O be the set of query responses from the queries Gen makes to $\mathcal{O}$. We first imagine replacing the measurements $\Pi_i = \Pi_{\sigma,q_i,f}$ with $\Pi_i = \Pi_{\sigma,q_i,f'}$, where

$$f'(\hat{C},x) = \begin{cases} f(\hat{C},x) & \text{if } \hat{C} \in O \\ \bot & \text{otherwise} \end{cases}$$

A routine argument shows that $\mathsf{Ver}^{\mathcal{E}}$ only has negligible query mass to $\mathcal{E}$ on $(\hat{C},x)$ such that both $\hat{C} \notin O$ and $f(\hat{C},x) \neq \bot$. This is because $\hat{C}$ that are in the image of $\mathcal{O}$ are negligibly sparse in $\{0,1\}^w$, and Ver can only have information about the $\hat{C}$ that were the result of previous queries to $\mathcal{O}$, namely the set O.

Then we observe that f and f' coincide except on $(\hat{C},x)$ such that both $\hat{C} \notin O$ and $\mathcal{E}(\hat{C},x) \neq \bot$. Since the weight on these points is negligible, Lemma 3 shows that the effect of switching from f to f' is negligible. We therefore proceed assuming the measurements applied are $\Pi_i = \Pi_{\sigma,q_i,f'}$. This introduces an error that gives the third term in the statement of Lemma 25.

Suppose that $R_{i-1} \geq 1 - p(1-c) - \gamma - 2i\epsilon$. Applying Lemma 6 gives $R_i \geq R_{i-1} - 2\epsilon \geq 1 - p(1-c) - \gamma - 2(i+1)\epsilon$, except with probability $Z = N\delta + N/T + 4\sqrt{\delta}$ where N is the number of outcomes of the measurement $\Pi_i = \Pi_{\sigma,q_i,f'}$. But the number of outcomes is $|O|+1 = P+1$, as the outcome can either be $\bot$ or one of the P circuits C such that are queried by Gen. Recall that $T = \sqrt{1/\delta}$. Thus $Z = (P+1)(\delta + \sqrt{\delta}) + 4\sqrt{\delta} \leq (2P+6)\sqrt{\delta}$. Union-bounding over all i gives the the second term in the statement of Lemma 25, proving the lemma. □

Lemma 26. *Consider the state ρ_i of $\mathcal{H} \otimes \mathcal{Z}$ at the end of Step 5e during the ith iteration. Assume $R_i \geq 1 - p(1-c) - \gamma - 2(i+1)\epsilon$, and assume $p/(1-p) \leq c/3$ and $2(U+1)\epsilon + \gamma + \delta \leq pc/3$. Then $\Pr[1 \leftarrow \mathsf{Ver}^{\mathcal{E}}(\sigma, \rho_i)] \geq c/9$.*

Proof. The requirement that $R_i \geq 1 - p(1-c) - \gamma - 2(i+1)\epsilon$ is the same as saying that $\mathsf{Ver}_*(\sigma, \cdot \mid c, p, \epsilon, \delta, \gamma_i)$ accepts ρ_i, where $\gamma_i = \gamma + 2i\epsilon$. Then Lemma 26 follows immediately from Lemma 18. □

Lemma 27. *Let S be the set of terms $(\hat{C},x)$ such that (1) $\hat{C}$ was the result of an obfuscation query by Gen, and (2) $\hat{C}$ does* not *appear in L by the end of running $\mathcal{A}$. Consider applying $\mathsf{Ver}^{\mathcal{E}}(\sigma,\rho_i)$ where ρ_i is as in Lemma 26. Then there exists some iteration $i \in [1, U-1]$ such that the query weight $\mathsf{Ver}^{\mathcal{E}}(\sigma,\rho_i)$ places on S is at most $PQ/(U-1)$*

Proof. Let L_i be the list L at the end of the ith iteration, and let $L_0 = \{\}$ be the initial L. Let S_i be the set of terms $(\hat{C},x)$ such that $\hat{C}$ does *not* appear in L_i. Since $L_i \subseteq L_U$, we have $S = S_U \subseteq S_i$. Let $w_i(S)$ and $w_i(S_i)$ be the query weight $\mathsf{Ver}^{\mathcal{E}}(\sigma,\rho_i)$ places on S and S_i respectively. Observe that, in iteration $i+1$, we apply $\Pi_{\sigma,q_{i+1},f}$ to ρ_i. Therefore, the probability of obtaining a $\hat{C} \notin L_i$ in iteration $i+1$ is $w_i(S_i)/Q$; in such an event, we have that $|L_{i+1}| = |L_i| + 1$.

Therefore, $\mathbb{E}[L_U] = \sum_{i=1}^{U-1} w_i(S_i)/Q \geq Q^{-1} \sum_{i=1}^{U-1} w_i(S)$. But we also have that $\mathbb{E}[L_U] \leq P$. Therefore, there must exist some $i \in [1, U-1]$ such that $w_i(S) \leq PQ/(U-1)$. □

Now fix an i guaranteed by Lemma 27. Then assuming $R_i \geq 1 - p(1-c) - \gamma - 2(i+1)\epsilon$ (which happens with probability at least X by Lemma 25) we have that $\Pr[\mathsf{Ver}^{\mathcal{E}}(\sigma,\rho_i)] \geq c/9$. Moreover, the query weight $\mathsf{Ver}^{\mathcal{E}}(\sigma,\rho_i)$ places on S is at most $PQ/(U-1)$. Therefore, by Lemma 3, we have that $\Pr[\mathsf{Ver}^{\mathcal{E}'}(\sigma,\rho_i)] \geq c/9 - Q\sqrt{P/(U-1)} = V$, and by Corollary 4, the query weight $\mathsf{Ver}^{\mathcal{E}'}(\sigma,\rho_i)$ places on S is at most $Q^2\sqrt{P/(U-1)} = W$. Thus, conditioned on $R_i \geq 1-p(1-c)-\gamma-2(i+1)\epsilon$, the state ρ_i satisfies the conditions needed in Step 7, and so the necessary state ρ exists. Thus $\mathcal{A}^{\mathcal{E}}(\sigma,\$)$ will output two copies of this state, which by Lemma 24 will be accepted by $\mathsf{Ver}^{\mathcal{E}}(\sigma,\cdot)$ with probability $V - O(\sqrt{QW})$. These events are independent since they are two independent copies of the same state. Putting it all together, the overall success probability of our adversary is at least $X(V-\sqrt{QW})^2$.

It remains to choose values for $p, \gamma, \epsilon, U, \delta$. We need the following:

- $p/(1-p) \leq c/3$ for Lemma 26.
- $2(U+1)\epsilon + \gamma + \delta \leq pc/3$ for Lemma 26.
- X is non-negligible, where $X = \left(\frac{\gamma}{p(1-c)+\gamma} - \delta\right) - \left(U(2Q+6)\sqrt{\delta}\right) - (\mathsf{negl})$.
- $V - z\sqrt{QW}$ is non-negligible, where $V = c/9 - Q\sqrt{P/(U-1)}$ and $W = Q^2\sqrt{P/(U-1)}$ and z is the constant hidden in Lemma 24.
- p, γ, ϵ, U are polynomial (but δ can be super-polynomial).

The following values suffice:

- $\delta = 2^{-\lambda}$. Then $X = \frac{\gamma}{p(1-c)+\gamma} - \mathsf{negl}$, which is inverse polynomial for any polynomials γ, p and inverse polynomial c.
- $p = c/(3+c)$, giving $p/(1-p) = c/3$.
- $U = z^4 3^{12} Q^6 c^{-4} P + 1$, giving $V - z\sqrt{QW} \geq c/27$.
- $\gamma = \epsilon = pc/4(2U+3)$, giving $2(U+1)\epsilon + \gamma + \delta = pc/4 + \delta \leq pc/3$.

This completes the proof of Theorem 22. □

References

[Aar04] Scott Aaronson. Limitations of quantum advice and one-way communication. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 320–332. IEEE, 2004.

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

[ABG+21] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 435–464. Springer, Cham, October 2021.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012.

[ACC$^+$22] Per Austrin, Hao Chung, Kai-Min Chung, Shiuan Fu, Yao-Ting Lin, and Mohammad Mahmoody. On the impossibility of key agreements from quantum random oracles. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 165–194. Springer, Cham, August 2022.

[AHY23] Prabhanjan Ananth, Zihan Hu, and Henry Yuen. On the (im)plausibility of public-key quantum money from collision-resistant hash functions. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part VIII*, volume 14445 of *LNCS*, pages 39–72. Springer, Singapore, December 2023.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[AL21] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 501–530. Springer, Cham, October 2021.

[ALL$^+$21] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 526–555, Virtual Event, August 2021. Springer, Cham.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, Oct 1997.

[BBC$^+$14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 26–51. Springer, Berlin, Heidelberg, February 2014.

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

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

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, August 2001.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 544–574. Springer, Cham, November 2018.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1091–1102. ACM Press, June 2019.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In Venkatesan Guruswami, editor, *56th FOCS*, pages 1480–1498. IEEE Computer Society Press, October 2015.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 269–279. ACM Press, June 2020.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Berlin, Heidelberg, August 2014.

[CGH17] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 278–307. Springer, Cham, April / May 2017.

[CHN$^+$16] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 1115–1127. ACM Press, June 2016.

[CLLZ21] Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 556–584, Virtual Event, August 2021. Springer, Cham.

[CLM23] Kai-Min Chung, Yao-Ting Lin, and Mohammad Mahmoody. Black-box separations for non-interactive classical commitments in a quantum world. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 144–172. Springer, Cham, April 2023.

[CMSZ22] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *62nd FOCS*, pages 49–58. IEEE Computer Society Press, February 2022.

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

[CPR17] Ran Canetti, Oxana Poburinnaya, and Mariana Raykova. Optimal-rate non-committing encryption. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 212–241. Springer, Cham, December 2017.

[CVW$^+$18a] Yilei Chen, Vinod Vaikuntanathan, Brent Waters, Hoeteck Wee, and Daniel Wichs. Traitor-tracing from LWE made simple and attribute-based. In Amos Beimel and

Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 341–369. Springer, Cham, November 2018.

[CVW18b] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 577–607. Springer, Cham, August 2018.

[FGH$^+$12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter W. Shor. Quantum money from knots. In Shafi Goldwasser, editor, *ITCS 2012*, pages 276–289. ACM, January 2012.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Berlin, Heidelberg, March 2015.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.

[GSWW22] Rachit Garg, Kristin Sheridan, Brent Waters, and David J. Wu. Fully succinct batch arguments for NP from indistinguishability obfuscation. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 526–555. Springer, Cham, November 2022.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Berlin, Heidelberg, August 2015.

[HHSS17] Shai Halevi, Tzipora Halevi, Victor Shoup, and Noah Stephens-Davidowitz. Implementing BP-obfuscation using graph-induced encoding. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 783–798. ACM Press, October / November 2017.

[HJL21] Samuel B. Hopkins, Aayush Jain, and Huijia Lin. Counterexamples to new circular security assumptions underlying iO. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 673–700, Virtual Event, August 2021. Springer, Cham.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 201–220. Springer, Berlin, Heidelberg, May 2014.

[HY20] Akinori Hosoyamada and Takashi Yamakawa. Finding collisions in a quantum world: Quantum black-box separation of collision-resistance and one-wayness. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 3–32. Springer, Cham, December 2020.

[JLLS23] Aayush Jain, Huijia Lin, Paul Lou, and Amit Sahai. Polynomial-time cryptanalysis of the subspace flooding assumption for post-quantum io. 2023.

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

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

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

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

[LLQZ22] Jiahui Liu, Qipeng Liu, Luowen Qian, and Mark Zhandry. Collusion resistant copy-protection for watermarkable functionalities. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 294–323. Springer, Cham, November 2022.

[LMZ23] Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 611–638. Springer, Cham, April 2023.

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

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 326–355. Springer, Cham, August 2019.

[MW04] C. Marriott and J. Watrous. Quantum arthur-merlin games. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 275–285, 2004.

[Rob21] Bhaskar Roberts. Security analysis of quantum lightning. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 562–567. Springer, Cham, October 2021.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 535–559. Springer, Cham, August 2022.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Berlin, Heidelberg, May 2016.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-IO from evasive LWE. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 195–221. Springer, Cham, December 2022.

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

[Win99] A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Trans. Inf. Theor.*, 45(7):2481–2485, November 1999.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Cham, October 2021.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 408–438. Springer, Cham, May 2019.

[Zha20] Mark Zhandry. Schrödinger's pirate: How to trace a quantum decoder. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 61–91. Springer, Cham, November 2020.

[Zha21] Mark Zhandry. White box traitor tracing. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 303–333, Virtual Event, August 2021. Springer, Cham.

[Zha24] Mark Zhandry. Quantum money from abelian group actions. In Venkatesan Guruswami, editor, *ITCS 2024*, volume 287, pages 101:1–101:23. LIPIcs, January / February 2024.