Franchised Quantum Money

Bhaskar Roberts[1] and Mark Zhandry[2,3]

[1] UC Berkeley
[2] Princeton University
[3] NTT Research

Abstract. The construction of public key quantum money based on standard cryptographic assumptions is a longstanding open question. Here we introduce franchised quantum money, an alternative form of quantum money that is easier to construct. Franchised quantum money retains the features of a useful quantum money scheme, namely unforgeability and local verification: anyone can verify banknotes without communicating with the bank. In franchised quantum money, every user gets a unique secret verification key, and the scheme is secure against counterfeiting and sabotage, a new security notion that appears in the franchised model. Finally, we construct franchised quantum money and prove security assuming one-way functions.

1 Introduction

The application of quantum information to unforgeable currency was first envisioned by Wiesner [Wie83], and these early ideas laid the foundation for the field of quantum cryptography. However, Wiesner's scheme for quantum money has a major drawback: verifying that a banknote is valid requires a classical description of the state, so the banknote must be sent back to the bank for verification.

The key properties that make cash (paper bills) useful are that anyone can verify banknotes *locally*, without communicating with the bank, and the banknotes are hard to counterfeit. In a classical world, digital currency cannot hope to achieve these properties because any classical bitstring can be duplicated. In a quantum world, we have hope for uncounterfeitable money because of the no-cloning theorem.

Recent works [Aar09, FGH$^+$12, AC12, Zha19] have sought a *public* test to verify banknotes. A scheme with such a test is called public key quantum money (or PKQM). Unfortunately, a convincing construction of public key quantum money has been notoriously elusive. Most proposals have been based on new ad hoc complexity assumptions, and in many cases those assumptions were broken [FGH$^+$12, PFP15, Aar16]. Recently, Zhandry [Zha19] showed that the [AC12] scheme can be instantiated using recent indistinguishability obfuscators. However, the quantum security of such obfuscators is currently unclear. Zhandry also proposed a new quantum money scheme in [Zha19], but the security of his scheme was also called into question [Rob21].

Franchised Quantum Money: In this work, we introduce franchised quantum money (FQM), which is useful as a currency system, easier to construct than public key quantum money, and potentially a stepping stone to PKQM. In franchised quantum money, every user receives a unique secret verification key. With their key, a user can verify banknotes locally, but they cannot create counterfeit money that would fool another user. Our main result is to show how to realize franchised quantum money under essentially minimal assumptions, namely one-way functions.

Franchised quantum money is a secret key scheme that approximates the functionality of a public key scheme. In particular, franchised quantum money achieves local verification[4].

The franchised verification model is broadly useful for approximating the security guarantees of public key verification. Building off of an earlier, unpublished version of this paper, [KNY21] proposed a franchised verification model for quantum lightning, and combined with a lattice assumption that we also proposed, they constructed a scheme for secure software leasing.

[4] [BS20] also propose a quantum money scheme that tries to approximate the functionality of PKQM. However, their scheme does not achieve local verification: their banknotes must be periodically sent back to the bank for verification. Furthermore, the way they define security is hard to justify.

The central feature of franchised quantum money is that each user has a unique secret key. Furthermore, we only require that an adversary cannot trick a *different* user into accepting a counterfeit banknote.

The difficulty with PKQM is that if the adversary knows the verification key, they know what properties of the state will be tested during verification. It is hard to design a verification procedure that reveals just enough information to verify banknotes, without giving enough information to create fake banknotes that fool the verifier.

Franchised quantum money does not have this issue. The adversary does not know any other user's key, so they don't know what properties the other user will test during verification. Therefore it is hard for the adversary to trick the other user into accepting a counterfeit banknote.

1.1 Technical Details

Definition of Franchised Quantum Money: In franchised quantum money, there is a trusted party, called the bank, that administers the currency system by generating verification keys and banknotes. A banknote is valid if it was generated by the bank.

The other participants in the system are untrusted users, who send and receive banknotes among each other. Each user can request a unique secret verification key from the bank. The key allows the user to verify any banknote they receive, and valid banknotes are accepted by verification with overwhelming probability.

Some users (the adversaries) are malicious and try to trick other users into accepting invalid banknotes. However it's hard for an adversary to create invalid banknotes that another user would accept.

Security: In order to be considered secure, a franchised quantum money scheme must be secure against both counterfeiting and sabotage.

Security against counterfeiting: We say that the scheme is *secure against counterfeiting* if it is hard for an adversary with m valid banknotes to get any other users to accept $m + 1$ banknotes. The key difference from public key quantum money lies in the word *other*. We don't care if the adversary can produce $m + 1$ banknotes that they themself would accept.

In fact in our construction, it's easy for the adversary to "trick themself" into accepting invalid banknotes, because if they know what key will be used in verification, they can create invalid banknotes that will be accepted. However, a different user with a key that is unknown to the adversary will recognize these banknotes as invalid.

Security against sabotage: Because each user has a different key, there is a second kind of security we need to consider. We don't want one user to accept an invalid banknote that another user would reject.

We call this attack sabotage:[5] the adversary takes a valid banknote and modifies it. Then they give it to one user, who accepts it even though the banknote is invalid. But when the first user tries to spend the banknote with a second user, the second user rejects the banknote.

How could sabotage be possible if the scheme is secure against counterfeiting? The adversary does not need to spend more banknotes than they received in order to succeed at sabotage.

A scheme is *secure against sabotage* if the adversary cannot produce a banknote that one other user accepts but which a second other user rejects.

Remark 1. We note that sabotage attacks are also a potential concern for public key quantum money schemes. Even though all users run the same verification procedure, technically two successive runs of the procedure may not output the same result. However, this problem can always be avoided by implementing verification as a projective measurement.

Furthermore, in practice, decoherence between runs may cause successive runs to behave differently. In this case too, sabotage attacks may be relevant.

To the best of our knowledge, this is the first work to point out these potential problems.

[5] We borrow this name from [BS20].

If an FQM scheme is secure against counterfeiting and sabotage, then it is practically useful as currency. This is because users can trust that any banknote they accept will be accepted by all other users, and the money supply will not increase unless the bank produces more banknotes. Therefore, these banknotes can hold monetary value. Quantum money does not need to be public key in order to be useful as a currency system.

Construction from Hidden Subspaces: Our construction of FQM is based on [AC12]'s proposal for PKQM from black-box subspace oracles. Below is a simplified version of our construction. A less-simplified version is given in section 4, and the full version is given in section 5.

Banknote: The banknote is an n-qubit quantum state. We can think of its computational basis states as vectors in $\mathbb{Z}_2^n$. The banknote $|A\rangle$ is a superposition over some random subspace $A \leq \mathbb{Z}_2^n$ such that $dim(A) = dim(A^\perp) = n/2$. We call this state a subspace state.

$$|A\rangle = \frac{1}{\sqrt{|A|}} \sum_{\mathbf{x} \in A} |\mathbf{x}\rangle$$

Verification key: For a given banknote $|A\rangle$, each verification key is a pair of random subspaces (V, W). $V \leq A$ and $W \leq A^\perp$, and the dimension of V and W is $t := \Theta(\sqrt{n})$. Each verifier gets an independently random (V, W).

Verification: To verify a banknote, the verifier performs two tests, one in the computational basis, and one in the Fourier basis.

First we test that the classical basis states of $|A\rangle$ are in $W^\perp$.

Then we take the quantum Fourier transform of the banknote. If the banknote is valid, the resulting state, $|\widetilde{A}\rangle$, is a superposition over $A^\perp$ ([AC12]):

$$|\widetilde{A}\rangle = \left|A^\perp\right\rangle = \frac{1}{\sqrt{|A^\perp|}} \sum_{\mathbf{y} \in A^\perp} |\mathbf{y}\rangle$$

Next, in the Fourier basis, we test that the vectors in $|\widetilde{A}\rangle$'s superposition are in $V^\perp$. Finally we take the inverse quantum Fourier transform, and return the resulting state. We accept the banknote if both tests passed. If the banknote was valid, the final state is the same as the initial one.

Discussion: A verifier will accept any subspace state $|B\rangle$ where $V \leq B \leq W^\perp$. Note that the adversary can easily construct a $|B\rangle$ based on their key (V, W) that they themself would accept.

However, an adversary cannot trick other users into accepting an invalid banknote. With probability overwhelming in n, the other user's (V, W) include dimensions of A and $A^\perp$, respectively, that are unknown to the adversary. Any banknote the adversary tries to produce, other than an honest banknote, will almost certainly get "caught" by these other dimensions and rejected.

Multiple banknotes. In the simplified construction above, one verification key (V, W) cannot verify multiple banknotes. Each banknote uses a different subspace A, and (V, W) depend on the choice of A.

However in the full construction, one verification key needs to verify every banknote the user receives. To achieve this, we assume the existence of one-way functions, which implies CPA-secure encryption. First, (V, W) are encrypted and appended to the banknote as a classical ciphertext. Then the decryption key serves as the verification key – the verifier decrypts the ciphertext to get (V, W), which they use to verify the banknote.

It is straightforward to see that *some* computational assumptions are necessary for franchised quantum money, since given an unlimited number of banknotes, the bank's master secret key is information-theoretically determined. So our construction of franchised quantum money uses essentially minimal assumptions.

Franchised vs. Obfuscated Verification: The franchised verification model allows us to avoid using obfuscation when constructing quantum money, and the model may be useful beyond quantum money as a way to avoid obfuscation.

[AC12, Zha19]'s construction of PKQM relies on strong forms of obfuscation, such as post-quantum-secure iO, for which we we have no convincing construction. The PKQM construction is like our FQM construction, except every verifier uses $V = A$ and $W = A^{\perp}$. We call this *full* verification, in contrast to franchised verification. Additionally, the oracles checking membership in A and $A^{\perp}$ are obfuscated so the adversary can't learn A.

In the franchised model, there is no need for obfuscation. The adversary only gets query access to the verifier, and they do not know the other users' verification keys. It is therefore feasible to construct FQM from assumptions weaker than obfuscation.

Finally, the franchised verifiers enjoy essentially the same security as full verifiers. We will show that the adversary cannot distinguish whether they're interacting with a full verifier or a franchised verifier, so our FQM construction inherits the security guarantees of the PKQM construction.

Colluding adversaries: As we defined FQM above, each user receives one verification key. But in the real world, it's possible that multiple adversaries collude: they pool their verification keys to gain more counterfeiting or sabotage power.

In our construction, each key gives a small number of dimensions of A and $A^{\perp}$. If the adversary has unlimited verification keys, then they can learn all of A and $A^{\perp}$ and produce as many copies of $|A\rangle$ as they want. So we will impose a collusion bound: no more than $C = \frac{n}{4t}$ adversaries can work together. This means no adversary learns more than $n/4$ dimensions of A (or $A^{\perp}$). With this collusion bound, the scheme is secure.

Although our scheme needs large banknotes to handle a large collusion bound, this may be reasonable in any scenario where the number of users is small – for example, in markets for certain financial securities, event tickets, etc. [6]

Additionally, collusion bounds are commonplace in cryptography, for example in traitor tracing. Our construction is analogous to the early days of traitor tracing, where the initial schemes [CFN94] had ciphertexts with size linear in the collusion bound, and the main goal became to shrink the ciphertext size. Eventually, [GKW18] essentially removed the collusion bound, giving a construction that is secure against exponentially many colluding adversaries, as a function of the ciphertext size.

Finally, we expect that any FQM scheme will require a collusion bound of some kind or else it would likely yield PKQM. See section 1.2 for more detail.

1.2 Next Steps

Increase the collusion bound: The main open problem is to increase the collusion bound, while maintaining small banknotes and verification keys. In our construction of FQM, the size of the banknotes (n) grows faster than the collusion bound ($C = \Theta(\sqrt{n})$). A reasonable next step is to construct a scheme whose banknote size grows slower than the collusion bound.

Here are two possible approaches: first, we might use LWE or similar assumptions to add noise to the verification keys. Given many noisy keys, an adversary would hopefully be unable to learn the secret information needed for counterfeiting. LWE has been used in traitor tracing [GKW18] to increase the collusion bound while achieving short ciphertexts and secret keys (which are analogous to banknotes and verification keys).

Second, we can use combinatorial techniques, such as those used for traitor tracing in [BN08]. [BN08]'s techniques have resulted in optimally short ciphertexts and might be used to achieve short banknotes. However, combinatorial techniques in traitor tracing usually come at the cost of much larger secret keys, and we might expect something similar for franchised quantum money.

Work up to public key quantum money: Franchised quantum money is a potential stepping stone to PKQM. Intuitively, the larger the collusion bound, the more the scheme behaves like PKQM, and we expect that PKQM can be easily constructed from an FQM construction that has unbounded collusion.

[6] We thank an anonymous reviewer for suggesting these applications.

Hypothetically, how would we prove security for an FQM scheme with unbounded collusion? The reduction would have to generate the adversary's verification keys, and somehow use the adversary's forgery for honest keys to break some underlying hard problem. But if the reduction could generate new verification keys for itself, then the construction might also be able to generate these new keys. If this were the case, we would easily get a public key quantum money scheme: to verify a banknote, generate a new verification key for yourself, and use that key.

Franchised semi-quantum money: We can make the mint in our scheme entirely classical, similar to the semi-quantum money scheme of [RS19], which is a secret key scheme. This follows from the fact that anyone can create new (un-signed) banknotes. To create and send a new banknote to a recipient, the recipient will generate a new un-signed banknote $|\$\rangle$ with serial number $\mathbf{y}$ on its own. It will then send $\mathbf{y}$ to the mint, who will sign $\mathbf{y}$ with a classical signature scheme.

2 Preliminaries

Subspaces

- For any subspace $A \leq \mathbb{Z}_2^n$, A will also refer to a matrix whose columns are a basis of the subspace A. The matrix serves as a description of the subspace.
- Let $A^\perp = \{\mathbf{x} \in \mathbb{Z}_2^n \ | \forall \mathbf{a} \in A, \langle \mathbf{x}, \mathbf{a} \rangle = 0\}$ be the orthogonal complement of A.
- Let $|A\rangle = \frac{1}{\sqrt{|A|}} \sum_{\mathbf{x} \in A} |\mathbf{x}\rangle$
- Let $O_A : \mathbb{Z}_2^n \to \{0,1\}$ decide membership in A. That is, $\forall \mathbf{x} \in \mathbb{Z}_2^n$:

$$O_A(\mathbf{x}) = \mathbb{1}_{\mathbf{x} \in A}$$

  Given a basis B of $A^\perp$, we can compute O_A as follows:

$$O_A(\mathbf{x}) = \mathbb{1}_{B^T \cdot \mathbf{x} = \mathbf{0}}$$

Quantum computation.

Here we recall the basics of quantum computation, and refer to Nielsen and Chuang [NC00] for a more detailed overview.

A quantum system is a Hilbert space $\mathcal{H}$ and an associated inner product $\langle \cdot | \cdot \rangle$. The state of the system is given by a complex unit vector $|\psi\rangle$. Given quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the joint quantum system is given by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$. Given $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$, we denote the product state by $|\psi_1\rangle|\psi_2\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$. A quantum state $|\psi\rangle$ can be "measured" in an orthonormal basis $B = \{|b_0\rangle, ..., |b_{d-1}\rangle\}$ for $\mathcal{H}$, which gives value i with probability $|\langle b_i|\psi\rangle|^2$. The quantum state then collapses to the basis element $|b_i\rangle$.

For a state over a joint system $\mathcal{H}_1 \otimes \mathcal{H}_2$, we can also perform a partial measurement over just, say, $\mathcal{H}_1$. Let $\{|a_0\rangle, ...\rangle\}$ be a basis for $\mathcal{H}_1$ and $\{|b_0\rangle, ...\rangle\}$ a basis for $\mathcal{H}_2$. Then for a general state $|\psi\rangle = \sum_{i,j} \alpha_{i,j} |a_i\rangle|b_j\rangle$, measuring in $\mathcal{H}_1$ will give the outcome i with probability $p_i = \sum_j |\alpha_{i,j}|^2$. In this case, the state collapses to $\sqrt{1/p_i} \sum_j \alpha_{i,j} |a_i\rangle|b_j\rangle$.

Operations on quantum states are given by unitary transformations over $\mathcal{H}$. An efficient quantum algorithm is a unitary U that can be decomposed into a polynomial-sized circuit, consisting of unitary matrices from some finite set.

Miscellaneous

A function $f(\lambda)$ is *negligible*, written as $f(\lambda) = \mathsf{negl}(\lambda)$, if $f(\lambda) = o(\lambda^{-c})$ for any constant c. $\mathsf{poly}(\lambda)$ is a generic polynomial in λ. A probability p is *overwhelming* if $1 - p = \mathsf{negl}(\lambda)$. Finally $[\lambda] = \{1, \ldots, \lambda\}$, for any $\lambda \in \mathbb{N}$. Numbers are assumed to be in $\mathbb{N}$ unless otherwise stated.

3 Definition of Franchised Quantum Money

Here we'll define franchised quantum money and its notions of security in detail.

Definition 1 (Main Variables).

- *Let $\lambda \in \mathbb{N}$ be the security parameter.*
- *Let $N \in \mathbb{N}$ be the number of verification keys that the bank distributes. $N = O(\mathsf{poly}(\lambda))$ in the security game because the adversary cannot query more than polynomially-many users.*
- *Let $C \in [N]$ be the collusion bound, the maximum number of verification keys that the adversary can receive.*
- *Let msk be the master secret key, known only by the bank.*
- *Let svk be a secret verification key given to a user.*
- *Let $|\$\rangle$ be a valid banknote. Let $|P\rangle$ be a purported banknote, which may or may not be valid.*
- *After verification, $|\$\rangle$ becomes $|\$'\rangle$, and $|P\rangle$ becomes $|P'\rangle$.*

Definition 2. *A **franchised quantum money scheme** $\mathcal{F}$ comprises four polynomial-time quantum algorithms: Setup, $\mathsf{Franchise}$, Mint, and Ver.*

1. **Setup***: The bank runs Setup to initialize the FQM scheme.*
$$msk \leftarrow \mathsf{Setup}(1^\lambda)$$
2. **Franchise***: The bank runs $\mathsf{Franchise}$ whenever a user requests a secret verification key. Then the bank sends svk to the user.*
$$svk \leftarrow \mathsf{Franchise}(msk)$$
3. **Mint***: The bank runs Mint to create a new banknote $|\$\rangle$. Then the bank gives $|\$\rangle$ to someone who wants to spend it.*
$$|\$\rangle \leftarrow \mathsf{Mint}(msk)$$
4. **Ver***: Any user with a secret verification key can run Ver to check whether a purported banknote $|P\rangle$ is valid. Ver accepts $|P\rangle$ ($b = 1$) or rejects $|P\rangle$ ($b = 0$). Finally, $|P\rangle$ becomes $|P'\rangle$ after it is processed by Ver.*
$$b, |P'\rangle \leftarrow \mathsf{Ver}(svk, |P\rangle)$$

In order to function as money, $|\$\rangle$ should be accepted by Ver with overwhelming probability, and $|\$'\rangle$ should be close to $|\$\rangle$. This way, we can verify the state in future transactions. The following definition, for correctness, achieves these properties.

Definition 3. *$\mathcal{F}$ is **correct** if for any $svk \leftarrow \mathsf{Franchise}(msk)$, any $|\$\rangle \leftarrow \mathsf{Mint}(msk)$, and any N and C that are polynomial in λ,*

1. *$\mathsf{Ver}(svk, |\$\rangle)$ accepts with probability overwhelming in λ, and*
2. *The trace distance between $|\$\rangle$ and $|\$'\rangle$ is $\mathsf{negl}(\lambda)$.*

Next, franchised quantum money needs two forms of security: security against counterfeiting and sabotage. Security against counterfeiting, defined below, means that an adversary given m banknotes cannot produce $m + 1$ banknotes that pass verification, except with $\mathsf{negl}(\lambda)$ probability.

Definition 4. *$\mathcal{F}$ is **secure against counterfeiting** if for any polynomial-time quantum adversary, the probability that the adversary wins the following security game is $\mathsf{negl}(\lambda)$:*

1. ***Setup**: The challenger is given $\lambda, N,$ and C, where $N, C = \mathsf{poly}(\lambda)$. Then the challenger runs $\mathsf{Setup}(1^\lambda)$ to get msk, and finally creates N verification keys $(svk_1, \ldots, svk_N)$ by running $\mathsf{Franchise}(msk)$ N times.*
2. ***Queries:** The adversary makes any number of franchise, mint, and verify queries, in any order:*
   - ***Franchise**: the challenger sends a previously unused key to the adversary. By convention, let the last C keys be sent to the adversary: $svk_{N-C+1}, \ldots, svk_N$.*
   - ***Mint**: The challenger samples $|\$\rangle \leftarrow \mathsf{Mint}(msk)$ and sends $|\$\rangle$ to the adversary.*

- ***Verify***: *The adversary sends a state $|P\rangle$ and an index $id \in [N-C]$ to the challenger. The challenger runs $\mathsf{Ver}(svk_{id}, |P\rangle)$, and sends the results $(b, |P'\rangle)$ back to the adversary.*

Let m be the number of mint queries made, which represents the number of valid banknotes the adversary receives.

3. ***Challenge***: *The adversary tries to spend $m+1$ banknotes. The adversary sends to the challenger $u > m$ purported banknotes, possibly entangled, each with an $id \in [N-c]$:*

$$(id_1, |P\rangle_1), (id_2, |P\rangle_2), \ldots, (id_u, |P\rangle_u)$$

Then for each purported banknote $|P\rangle_k$, the challenger runs Ver:

$$b_k, |P'\rangle_k \leftarrow \mathsf{Ver}(svk_{id_k}, |P\rangle_k)$$

The adversary wins the game if at least $m+1$ of the purported banknotes are accepted.

The second form of security is security against sabotage. Sabotage is when the adversary tricks one user into accepting an invalid banknote that is then rejected by a second user.

Definition 5. *$\mathcal{F}$ is **secure against sabotage** if for any polynomial-time quantum adversary, the probability that the adversary wins the following security game is $\mathsf{negl}(\lambda)$:*

1. ***Setup:*** *same as in definition 4*
2. ***Queries:*** *same as in definition 4*
3. ***Challenge:*** *The adversary sends to the challenger a banknote $|P\rangle$ and two distinct indices $id_1, id_2 \in [N-c]$.*
   The challenger runs Ver using svk_{id_1}, then svk_{id_2}:

$$b_1, |P'\rangle \leftarrow \mathsf{Ver}(svk_{id_1}, |P\rangle)$$
$$b_2, |P''\rangle \leftarrow \mathsf{Ver}(svk_{id_2}, |P'\rangle)$$

The adversary wins the game if the first verification accepts ($b_1 = 1$) and the second verification rejects ($b_2 = 0$).

4 Simple Construction

Here we give a simpler version of our construction of FQM in order to illustrate the main ideas. The simple construction is correct and secure, but only if the adversary gets just one banknote. The full construction of FQM is given in section 5.

Variables and Parameters

- Let N be any $\mathsf{poly}(\lambda)$.
- Let $n = \Omega(\lambda)$ be the dimension of the ambient vector space: $\mathbb{Z}_2^n$.
- Let $A < \mathbb{Z}_2^n$ be a subspace, and let $dim(A) = dim(A^\perp) = n/2$.
- Let $V \leq A$ and $W \leq A^\perp$ be two subspaces given by an svk.
- Let $t = \Theta(\sqrt{n})$ be an upper bound on the dimension of V and W.
- Let $C = \frac{n}{4t}$.

Setup

Input: 1^λ

1. Choose values for $N, n,$ and t.
2. Sample $A \leq \mathbb{Z}_2^n$ such that $dim(A) = dim(A^\perp) = n/2$.
3. For each $id \in [N]$: sample t indices uniformly and independently from $[n/2]$. Call this set I_{id}. Then sample another set called J_{id} from the same distribution.
4. Sample $\mathbf{v}_1, \ldots, \mathbf{v}_{n/2} \in A$ independently and uniformly at random.
   Sample $\mathbf{w}_1, \ldots, \mathbf{w}_{n/2} \in A^\perp$ independently and uniformly at random.
5. $$\text{Let } msk = \left(A, \{\mathbf{v}_i\}_{i\in[n/2]}, \{\mathbf{w}_j\}_{j\in[n/2]}, \{I_{id}, J_{id}\}_{id\in[N]}\right)$$
   and **output** msk.

Franchise

Input: msk

1. Choose an $id \in [N]$ that hasn't been chosen before.
2. Let $svk_{id} = \big(I_{id}, J_{id}, \{\mathbf{v}_i\}_{i\in I_{id}}, \{\mathbf{w}_j\}_{j\in J_{id}}\big)$, and **output** svk_{id}.

Mint

Input: msk

1. Generate and **output** $|\$\rangle = |A\rangle$.

Ver

Input: $svk, |P\rangle$

Let $svk = \big(I, J, \{\mathbf{v}_i\}_{i\in I}, \{\mathbf{w}_j\}_{j\in J}\big)$. Then let

$$V := span(\{\mathbf{v}_i\}_{i\in I}) \text{ and } W = span(\{\mathbf{w}_j\}_{j\in J})$$

1. **Computational basis test:** Check that $O_{W^\perp}\big(|P\rangle\big) = 1$. Now $|P\rangle$ becomes $|P_1\rangle$.
2. Take the quantum Fourier transform of $|P_1\rangle$ to get $\widetilde{|P_1\rangle}$.
3. **Fourier basis test:** Check that $O_{V^\perp}\big(\widetilde{|P_1\rangle}\big) = 1$. Now $\widetilde{|P_1\rangle}$ becomes $\widetilde{|P_2\rangle}$.
4. Take the inverse quantum Fourier transform of $\widetilde{|P_2\rangle}$ to get $|P_2\rangle$. Let $|P'\rangle = |P_2\rangle$. **Output** 1 (accept) if both tests pass, and 0 (reject) otherwise. Also output $|P'\rangle$.

Proofs of Correctness and Security

Theorem 1. *The simple FQM construction is correct.*

Proof. We will show that for any valid banknote $|\$\rangle = |A\rangle$, $\mathsf{Ver}(svk, |\$\rangle)$ outputs $(1, |\$\rangle)$ with probability 1.

1. The computational basis test passes with probability 1. $W \leq A^\perp$, so $A \leq W^\perp$, and $O_{W^\perp}(|A\rangle) = 1$ with probability 1. Also the banknote is unchanged by this test.
2. The quantum Fourier transform of the banknote is $\big|A^\perp\big\rangle$ ([AC12]).
3. The Fourier basis test also passes with probability 1. Since $V \leq A$, then $A^\perp \leq V^\perp$, and $O_{V^\perp}(\big|A^\perp\big\rangle) = 1$ with probability 1. The banknote is also unchanged by this test.
4. Finally, the inverse quantum Fourier transform restores the banknote to its initial state $|A\rangle$, and the banknote is accepted by Ver with probability 1.

□

Theorem 2. *The simple FQM construction is secure against counterfeiting if the adversary receives only $m = 1$ banknote.*

Proof.
1) Preliminaries
Let's say without loss of generality that the adversary receives C verification keys, which correspond to the last C identities: $id \in \{N-C+1, \ldots, N\}$. Then they receive 1 banknote, and then they make any polynomial number of verification queries. Finally, they attempt the counterfeiting challenge.

We can define the subspaces $V_{adv} \leq A$ and $W_{adv} \leq A^\perp$ as the subspaces known to the adversary. We also define V_{id} and W_{id} analogously for each $id \in [N]$:

Definition 6.

- *Let* $I_{adv} = \bigcup_{id>N-C} I_{id}$ *and* $J_{adv} = \bigcup_{id>N-C} J_{id}$.

- *For any $id \in [N]$, let $V_{id} = span\big(\{\mathbf{v}_i\}_{i \in I_{id}}\big)$. Let W_{id}, V_{adv}, and W_{adv} be defined analogously.*

Let's assume for simplicity that

$$dim(V_{adv}) = dim(W_{adv}) =: d$$

where d is fixed. This assumption isn't necessary for proving security, but it does make the proof simpler. Also note that $d \leq n/4$.

2) We'll use a hybrid argument to reduce the counterfeiting game to [AC12]'s security game for secret key quantum money:

- **h0** is the counterfeiting security game for the simple FQM construction. In particular, the adversary receives one banknote $|A\rangle$, along with C franchised verification keys.
- **h1** is the same as h0, except the challenger simulates full verifiers: whenever the adversary makes a verification query $(id, |P\rangle)$, the challenger verifies the state using O_A and $O_{A^\perp}$ instead of $O_{W_{id}^\perp}$ and $O_{V_{id}^\perp}$.
- **h2** is essentially [AC12]'s security game for secret key quantum money: let $A \leq \mathbb{Z}_2^{n-2d}$ be a uniformly random subspace such that $dim(A) = dim(A^\perp) = n/2 - d$. Next, the adversary gets a banknote $|A\rangle$ but no verification keys. They can make verification queries, and the challenger will run Ver using full verifiers: (O_A and $O_{A^\perp}$).

Lemma 1. *For any polynomial-time adversary $\mathcal{A}$, their success probabilities in $h0$ and in $h1$ differ by a $\mathsf{negl}(\lambda)$ function.*

We'll defer the proof of lemma 1 to section 6.

Lemma 2. *If $\mathcal{A}$ is a polynomial-time adversary with non-negligible success probability in $h1$, then there is a polynomial-time adversary $\mathcal{A}'$ with non-negligible success probability in $h2$.*

Proof. We can reduce the security game in $h2$ to the security game in $h1$. Let $\mathcal{A}'$ be given an $h2$ banknote $|A\rangle$, where $A \leq \mathbb{Z}_2^{n-2d}$ and $dim(A) = dim(A^\perp) = n/2 - d$. We will turn $|A\rangle$ into an $h1$ banknote $|B\rangle$, where $B \leq \mathbb{Z}_2^n$, and $dim(B) = dim(B^\perp) = n/2$:

1. Prepend $|A\rangle$ with $|0\rangle^{\otimes d} |+\rangle^{\otimes d}$:
$$\text{Let } |A'\rangle = |0\rangle^{\otimes d} |+\rangle^{\otimes d} |A\rangle$$
$|A'\rangle$ is a subspace state, a uniform superposition over the subspace
$$A' := span[\hat{e}_{d+1}, \ldots, \hat{e}_{2d}, (0^{\times 2d} \times A)]$$
where $0^{\times 2d} \times A$ is all vectors in $\mathbb{Z}_2^n$ for which the first $2d$ bits are 0 and the rest form a vector in A. Also, $dim(A') = dim(A'^\perp) = n/2$.
2. Sample an invertible matrix $M \in \mathbb{Z}_2^{n \times n}$ uniformly at random. Then apply M to $|A'\rangle$:
$$\text{Let } B = M \cdot A' \text{ and } |B\rangle = M(|A'\rangle)$$
Observe that $|B\rangle$ is a uniformly random $h1$ banknote.

Additionally, the adversary knows d dimensions of B and d dimensions of $B^\perp$:

$$V_{adv} = M \cdot span(\hat{e}_{d+1}, \ldots, \hat{e}_{2d})$$
$$W_{adv} = M \cdot span(\hat{e}_1, \ldots, \hat{e}_d)$$

$\mathcal{A}'$ derives C $h1$-verification keys whose vectors span V_{adv} and W_{adv}. Finally, $\mathcal{A}'$ runs $\mathcal{A}$, giving it the banknote $|B\rangle$ along with the verification keys.

When $\mathcal{A}$ makes a verification query $(id, |P\rangle)$, $\mathcal{A}'$ simulates the $h1$ challenger's response as follows, by converting $|P\rangle$ into an $h2$ banknote:

1. Let $|P'\rangle = M^{-1}(|P\rangle)$.

2. Check that the first $2d$ qubits of $|P'\rangle$ are $|0\rangle^{\otimes d}|+\rangle^{\otimes d}$.
3. Query the $h2$ challenger with the remaining $n-2d$ qubits of $|P'\rangle$. Let $|P''\rangle$ be the state returned by the challenger. Accept the banknote if and only if the first $2d$ qubits passed their test, and the challenger accepted as well.
4. Return $M(|0\rangle^{\otimes d}|+\rangle^{\otimes d}|P''\rangle)$ to the $h1$ adversary.

This procedure simulates $h1$ for $\mathcal{A}$. Also, note that the probability that $|P'\rangle$ passes $h2$ verification is at least the probability that $|P\rangle$ passes $h1$ verification.

Finally, when $\mathcal{A}$ attempts to win the challenge by outputting several purported $h1$ banknotes, $\mathcal{A}'$ converts these into $h2$ banknotes. If $\mathcal{A}$ wins in $h1$ with non-negligible probability, then $\mathcal{A}'$ wins in $h2$ with at least that probability. □

Lemma 3. *In $h2$, any polynomial-time adversary has negligible success probability.*

Proof. [AC12]'s security game is similar to $h2$, except the adversary can query both O_A and $O_{A^\perp}$. They proved the following:

Theorem 3 ([AC12], Theorem 25). *Let the adversary get $|A\rangle$, a random n'-qubit banknote, along with quantum query access to O_A and $O_{A^\perp}$. If the adversary prepares two possibly entangled banknotes that both pass verification with probability $\geq \varepsilon$, for all $1/\varepsilon = o(2^{n'/2})$, then they make at least $\Omega(\sqrt{\varepsilon}2^{n'/4})$ oracle queries.*

Let $n' = n - 2d$, the size of the banknote in $h2$. Note that $n' \geq n/2$. Next, let $\varepsilon = 2^{-n'/3}$. Note that $\varepsilon = \mathsf{negl}(\lambda)$. Finally, the number of queries needed to win with probability $\geq \epsilon$ is

$$\Omega(\sqrt{\varepsilon}2^{n'/4}) = \Omega(2^{n'/4-n'/6}) = \Omega(2^{n'/12})$$

Any polynomial-time adversary makes fewer than that many queries, so no polynomial-time adversary can win with non-negligible probability. □

Putting together lemmas 1, 2, 3, we get that any polynomial-time adversary has negligible probability of winning the counterfeiting security game for the simple construction of FQM. □

Theorem 4. *The simple FQM construction is secure against sabotage if the adversary receives only $m = 1$ banknote.*

Proof. The proof of this theorem follows the proof of 2, except at the end. We need to show that in $h2$, any polynomial-time adversary has negligible probability of succeeding at sabotage. To show this, we need the following lemma:

Lemma 4 ([AC12], Lemma 21). *In $h2$, Ver projects $|P\rangle$ onto $|A\rangle$ if it accepts and onto a state orthogonal to $|A\rangle$ if it rejects.*

That means that if a purported banknote is verified twice, it is either accepted both times or rejected both times. Therefore, sabotage is not possible in $h2$.

Again, by lemmas 1 and 2, any polynomial-time adversary has negligible probability of winning the sabotage security game for the simple construction of FQM. □

5 Full Construction

The full construction of FQM adds a signature scheme and a secret key encryption scheme, which let us hand out the subspaces V_{id}, W_{id} as part of the banknote. As a result, a user can verify many banknotes, each for a different subspace A, without needing to call $\mathsf{Franchise}$ for each banknote.

The signature and encryption schemes have the following syntax.

Definition 7. *([KL14], Definition 12.1) A **signature scheme** comprises the following three probabilistic polynomial-time algorithms:*

- SigKeyGen *takes a security parameter λ, and returns (sig_pk, sig_sk), the public and secret keys.*

$$sig\_pk, sig\_sk \leftarrow \mathsf{SigKeyGen}(1^\lambda)$$

- Sign *takes a message $msg \in \{0,1\}^*$ and the secret key and produces σ, the signature for msg.*

$$\sigma \leftarrow \mathsf{Sign}(sig\_sk, msg)$$

- SigVer *takes msg, σ, and the public key, and outputs a bit b to indicate the decision to accept ($b = 1$) or reject ($b = 0$) the signature-message pair. Also,* SigVer *is deterministic.*

$$b := \mathsf{SigVer}(sig\_pk, msg, \sigma)$$

The signature scheme is *existentially unforgeable under an adaptive chosen-message attack*. Such a signature scheme can be constructed from one-way functions ([KL14]).

Definition 8. *([KL14], Definition 3.7). A* ***secret key encryption scheme*** *comprises the following three probabilistic polynomial-time algorithms:*

- *EncKeyGen takes a security parameter λ and produces a secret key enc_k.*

$$enc\_k \leftarrow \mathit{EncKeyGen}(1^\lambda)$$

- *Enc encrypts a message $msg \in \{0,1\}^*$ using the key enc_k to produce a cyphertext c.*

$$c \leftarrow \mathit{Enc}(enc\_k, msg)$$

- *Dec decrypts c, again using enc_k. Dec is deterministic, so for any enc_k produced by EncKeyGen, Dec always decrpyts c correctly.*

$$msg := \mathit{Dec}(enc\_k, c)$$

The secret key encryption is *CPA-secure*, and it can also be constructed from one-way functions ([KL14]).

Variables

- Let $|\$\rangle$, a valid banknote, comprise a quantum state $|\Sigma\rangle$ and some classical bits.
- Let $|P\rangle$, a purported banknote, comprise a quantum state $|\Pi\rangle$ and some classical bits.

Setup

Input: 1^λ

1. Choose values for the parameters: $n = \Omega(\lambda), t = \Theta(\sqrt{n})$.
2. Set up one signature scheme and n encryption schemes by computing:

$$(sig\_pk, sig\_sk) \leftarrow \mathsf{SigKeyGen}(1^\lambda)$$
$$(enc\_k_1, \ldots, enc\_k_n) \leftarrow \mathsf{EncKeyGen}(1^\lambda), \ldots, \mathsf{EncKeyGen}(1^\lambda)$$

3. Let $msk = (sig\_pk, sig\_sk, enc\_k_1, \ldots, enc\_k_n)$, and then **output** msk.

Franchise

Input: msk

1. Sample t indices uniformly and independently from $[n/2]$. Call this set I. Then sample another set called J from the same distribution.
2. Let $svk = (sig\_pk, I, J, \{enc\_k_i\}_{i\in I}, \{enc\_k_{j+n/2}\}_{j\in J})$, and then **output** svk.

Mint

Input: msk

1. Sample a subspace $A < \mathbb{Z}_2^n$ such that $dim(A) = dim(A^\perp) = n/2$, uniformly at random.
2. Create the subspace state for A, and let $|\Sigma\rangle = |A\rangle$.
3. Sample $n/2$ random vectors in A: $\{\mathbf{v}_1, \ldots, \mathbf{v}_{n/2}\} \in_R A$. And sample $n/2$ random vectors in $A^\perp$: $\{\mathbf{w}_1, \ldots, \mathbf{w}_{n/2}\} \in_R A^\perp$.
4. Encrypt the $\mathbf{v}$s and $\mathbf{w}$s, each with a different enc_k:

$$\text{Let } c_1, \ldots, c_{\frac{n}{2}} = \left[\mathsf{Enc}(enc\_k_1, \mathbf{v}_1), \ldots, \mathsf{Enc}(enc\_k_{\frac{n}{2}}, \mathbf{v}_{\frac{n}{2}})\right.$$
$$c_{\frac{n}{2}+1}, \ldots, c_n = \left[\mathsf{Enc}(enc\_k_{\frac{n}{2}+1}, \mathbf{w}_1), \ldots, \mathsf{Enc}(enc\_k_n, \mathbf{w}_{\frac{n}{2}})\right.$$

5. Sign the ciphertexts. Let $\sigma \leftarrow \mathsf{Sign}[sig\_sk, (c_1, \ldots, c_n)]$.
6. Construct the banknote. Let $|\$\rangle = (|\Sigma\rangle, c_1, \ldots, c_n, \sigma)$. Finally, **output** $|\$\rangle$.

Ver

Inputs: $svk_{id}, |P\rangle$

1. Check the signature: $\mathsf{SigVer}(sig\_pk, (c_1, \ldots, c_n), \sigma)$.
2. Decrypt any ciphertexts for which the key is available. For every $i \in I_{id}$ compute $\mathbf{v}_i = \mathsf{Dec}(enc\_k_i, c_i)$, and for every $j \in J_{id}$, compute $\mathbf{w}_j = \mathsf{Dec}(enc\_k_{j+n/2}, c_{j+n/2})$.

   Additionally, define two subspaces, V_{id}, W_{id}:

$$V_{id} := span(\{\mathbf{v}_i\}_{i \in I_{id}})$$
$$W_{id} := span(\{\mathbf{w}_j\}_{j \in J_{id}})$$

3. Recall that $|P\rangle$ comprises a quantum state $|\Pi\rangle$ and some classical bits.
   Computational basis test: Check that $O_{W_{id}^\perp}(|\Pi\rangle) = 1$. After this step, $|\Pi\rangle$ becomes $|\Pi_1\rangle$.
4. Take the quantum Fourier transform of $|\Pi_1\rangle$ to get $|\widetilde{\Pi_1}\rangle$.
5. **Fourier basis test:** Check that $O_{V_{id}^\perp}(|\widetilde{\Pi_1}\rangle) = 1$. After this step, $|\widetilde{\Pi_1}\rangle$ becomes $|\widetilde{\Pi_2}\rangle$.
6. Take the inverse quantum Fourier transform of $|\widetilde{\Pi_2}\rangle$ to get $|\Pi_2\rangle$.
   Let $|P'\rangle$ be the state that $|P\rangle$ has become, with $|\Pi\rangle$ replaced with $|\Pi_2\rangle$.
   Output 1 (accept) if both tests pass, and 0 (reject) otherwise. Also output $|P'\rangle$.

Proofs of Correctness and Security

Theorem 5. *The full construction of franchised quantum money is correct.*

Proof. In steps 1 and 2 of Ver, we check the signature and decrypt the ciphertexts. With probability 1, the signature check passes, and the ciphertexts are correctly decrypted. This follows from the correctness of the signature and encryption schemes.

After the first two steps, Ver is the same as it was in the simple construction. Because the simple construction is correct, the full construction is correct as well. □

Theorem 6. *The full construction of franchised quantum money is secure against counterfeiting and sabotage.*

Proof. We will use a hybrid argument to show that the adversary's success probability at counterfeiting or sabotage with the full construction is close to what it is with the simple construction. Since the simple construction is secure against counterfeiting and sabotage, the full construction is secure as well.

1) Preliminaries
Without loss of generality, let us say that the adversary receives C svks, then receives m valid banknotes from the challenger, and finally makes multiple Ver queries.

Furthermore, let the challenger keep a record of all the banknotes and svks it generated. Finally let the ciphertexts $(c_1, \ldots, c_n)$ of each valid banknote be unique. This occurs with overwhelming probability.

2) Next, we'll use a sequence of hybrids to simplify the situation and remove the need for the signature and encryption schemes.

- **h0** uses the full FQM construction in the counterfeiting or sabotage security game.
- **h1** is the same as $h0$, except Ver only accepts a purported banknote if its ciphertexts $(c_1, \ldots, c_n)$ match those of one of the m valid banknotes given to the adversary.
- **h2** is the same as $h1$, except for any ciphertext c_i for which the adversary does not have the decryption key, c_i is replaced with junk: the encryption under enc_k_i of a random message.

The adversary has $\mathsf{negl}(\lambda)$ advantage in distinguishing $h0$ and $h1$. The signature scheme is existentially unforgeable under an adaptive chosen-message attack, so except with $\mathsf{negl}(\lambda)$ probability, any banknote that passed Ver in $h0$ had ciphertexts that matched one of the m valid banknotes.

The adversary has $\mathsf{negl}(\lambda)$ advantage in distinguishing $h1$ and $h2$ because the encryption scheme is CPA-secure. For any i for which the adversary does not have the decryption key, the adversary receives either m ciphertexts of random messages or m ciphertexts of potentially useful messages. CPA security is equivalent to left-or-right security ([KL14]), which implies that the adversary cannot distinguish these two cases.

3) Next, we'll use another set of hybrids to relate the full construction with the simple construction.

- **h3** is the same as $h2$, except we do not use the signature or encryption schemes. Each valid banknote comprises a subspace state $|\psi_A\rangle$ and a set of plaintext $\mathbf{v}$ vectors in A and $\mathbf{w}$ vectors in $A^\perp$. Finally, to verify a purported banknote, the challenger checks that the $\mathbf{v}$ and $\mathbf{w}$ vectors associated with a purported banknote match those of a valid banknote. Then they use whatever svks were recorded along with the valid banknote to verify the subspace state.
- **h4** is the simple FQM construction with just one banknote. This is the same as $h3$, except the adversary receives only 1 valid banknote, and the $\mathbf{v}$ and $\mathbf{w}$ vectors are given by $\mathsf{Franchise}$ and are not included with the banknote.

The adversary's best success probability is the same in h2 and h3 because the signature and encryption schemes were not necessary in h2, so h3 presents essentially the same security game to the adversary.

Lemma 5. *The best success probability for an adversary in h3 is at most m times the best success probability in h4.*

Proof. Given any $h3$ adversary $\mathcal{A}$, there is an $h4$ adversary $\mathcal{A}'$ that simulates $\mathcal{A}$. $\mathcal{A}'$ receives one valid banknote and generates $m-1$ other banknotes. Then $\mathcal{A}'$ runs $\mathcal{A}$ with the m banknotes. When $\mathcal{A}$ makes a verification query, $\mathcal{A}'$ simulates the verifier for the $m-1$ banknotes it generated and queries the $h4$ verifier for the banknote that it received. Finally, $\mathcal{A}$ outputs some purported banknotes at the challenge step, which $\mathcal{A}'$ outputs as well.

If $\mathcal{A}$ wins in $h3$, then there are at least $m+1$ purported banknotes that pass verification, and at least two of them have the same $\mathbf{v}$ and $\mathbf{w}$ vectors. $\mathcal{A}'$ wins in $h4$ if the two banknotes with matching vectors also match the vectors of the banknote given to $\mathcal{A}'$. This happens with probability $\frac{1}{m}$, by the symmetry of the m banknotes. Therefore, $\mathcal{A}'$'s success probability is $\frac{1}{m}$ times $\mathcal{A}$'s. □

4) In $h4$, the adversary has negligible probability of winning the counterfeiting or sabotage games, by theorems 2 and 4. Since $m = O(\mathsf{poly}(\lambda))$, for any polynomial-time adversary, then any polynomial-time adversary has negligible probability of winning the counterfeiting or security games for the full FQM construction. □

6 Distinguishing Game

In order to prove lemma 1, we will use the adversary method of [Amb02]. We will study the *distinguishing game*, in which an adversary that is more powerful than the one in lemma 1 tries to distinguish full and franchised verifiers. Then we show that the more-powerful adversary still has negligible advantage.

In the distinguishing game, the adversary is given a classical description of A, along with other information that is more than what they receive in the security game. However, one piece of information remains hidden to them: the verification keys used by the franchised verifiers. More formally, we say the adversary is given the msk, which includes every (V_{id}, W_{id}). But the verifiers will actually use $(M \cdot V_{id}, M \cdot W_{id})$ for some random matrix M. The next two definitions make this precise.

Definition 9. *Let $\mathcal{M}(A)$ be the set of all matrices $M \in \mathbb{Z}_2^{n \times n}$ such that:*

- *M is invertible*
- *If $\mathbf{x} \in A$, then $M^T\mathbf{x} \in A$, and if $\mathbf{x} \in A^\perp$, then $M^T\mathbf{x} \in A^\perp$.*

Definition 10. *For any $M \in \mathcal{M}(A)$, we also treat M as a function mapping one master secret key to another. Essentially, M is applied to every $\mathbf{v}$ or $\mathbf{w}$ vector that the adversary did not receive. More formally, for any msk:*

$$M(msk) = \Big(A, \{\mathbf{v}_i\}_{i \in I_{adv}}, \{M \cdot \mathbf{v}_i\}_{i \notin I_{adv}}, \{\mathbf{w}_j\}_{j \in J_{adv}}, \{M \cdot \mathbf{w}_j\}_{j \notin J_{adv}}, \{I_{id}, J_{id}\}_{id \in [N]}\Big)$$

Let $msk' = M(msk)$, and let $V'_{adv}, W'_{adv}, V'_{id}$, and W'_{id} be defined analogously. Then $V'_{adv} = V_{adv}$ and $W'_{adv} = W_{adv}$ because the adversary's $\mathbf{v}$ and $\mathbf{w}$ vectors are not changed by M. Therefore, in the counterfeiting and sabotage security games, the adversary receives the same information, whether the master secret key is msk or msk'.

Next, the adversary in the distinguishing game can also query $O_{W_{id}^\perp}$ and $O_{V_{id}^\perp}$, rather than just Ver. The following definitions bundle together the oracles that the adversary can query.

Definition 11. *The **franchised verification oracle** for a given msk is $O_{Fran}[msk]$. It takes as input an $id \in [N - C]$, a selection bit $s \in \{0, 1\}$, and a vector $\mathbf{x} \in \mathbb{Z}_2^n$. Then*

$$O_{Fran}[msk](id, s, \mathbf{x}) = \begin{cases} O_{W_{id}^\perp}(\mathbf{x}) & s = 0 \\ O_{V_{id}^\perp}(\mathbf{x}) & s = 1 \end{cases}$$

Definition 12. *The **full verification oracle** for a given msk is $O_{Full}[msk]$ or $O_{Full}[A]$. It takes as input $id \in [N - C]$, $s \in \{0, 1\}$, and $\mathbf{x} \in \mathbb{Z}_2^n$. Then*

$$O_{Full}[A](id, s, \mathbf{x}) = \begin{cases} O_A(\mathbf{x}) & s = 0 \\ O_{A^\perp}(\mathbf{x}) & s = 1 \end{cases}$$

Now we can define the distinguishing game precisely.

Definition 13. *The **distinguishing game** takes as input an msk, which is given to the challenger and the adversary. Then:*

1. *The challenger samples $b \in_R \{0, 1\}$ and $M \in_R \mathcal{M}(A)$.*
2. *The adversary makes quantum queries to the challenger. If $b = 0$, the challenger uses $O_{Full}[A]$ to answer the queries; if $b = 1$, the challenger uses $O_{Fran}[M(msk)]$.*
3. *The adversary outputs a bit b', and they win if and only if $b' = b$.*

Theorem 7. *Any polynomial-time quantum adversary $\mathcal{A}$ has negligible advantage in the distinguishing game. That is:*

$$\Big|P[\mathcal{A} = 1 | b = 0] - P[\mathcal{A} = 1 | b = 1]\Big| \leq \mathsf{negl}(\lambda)$$

where the probabilities are over the choice of $M \in \mathcal{M}(A)$ and $\mathcal{A}$'s randomness.

We'll prove theorem 7 later using the adversary method, but assuming theorem 7 for now, we can prove lemma 1.

Proof of lemma 1

We want to show that for any polynomial-time adversary $\mathcal{A}$, their success probabilities in $h0$ and in $h1$ differ by a $\mathsf{negl}(\lambda)$ function. Recall that $h0$ uses franchised verifiers, whereas $h1$ uses full verifiers.

Assume toward contradiction that $\mathcal{A}$'s success probabilities in $h0$ and $h1$ differ by a non-negligible amount. Then we can construct an adversary $\mathcal{A}'$ that has non-negligible advantage in the distinguishing game.

$\mathcal{A}'$ simulates the counterfeiting security game and runs $\mathcal{A}$ on it. Given msk, $\mathcal{A}'$ constructs $|A\rangle$ and the C franchised verification keys. When $\mathcal{A}$ queries a verifier, $\mathcal{A}'$ simulates this by querying either $O_{Full}[A]$ (if we're in $h1$) or $O_{Fran}[M(msk)]$ (if we're in $h0$). $\mathcal{A}'$ can even simulate the counterfeiting challenge, checking if $\mathcal{A}$ successfully counterfeited. Finally, $\mathcal{A}'$ outputs 1 if $\mathcal{A}$ won the security game, and 0 otherwise. $h0$ and $h1$ for the counterfeiting game correspond to $b = 1$ and $b = 0$ in the distinguishing game, so $\mathcal{A}'$ has non-negligible advantage in the distinguishing game.

This is a contradiction, by theorem 7, so in fact, the success probabilities of $\mathcal{A}$ in the two hybrids must be negligibly close.

The Adversary Method

Now we'll prove theorem 7 using the adversary method[7]. First, we'll define the scenario that [Amb02] considered, which is an abstract version of the distinguishing game, and then we'll state their main theorem.

Definition 14. *Let $\mathcal{O}$ be a set of oracles, each of which has range $\{0,1\}$. Let $f : \mathcal{O} \to \{0,1\}$ be a predicate that takes an oracle as input. Let X, Y partition $\mathcal{O}$ such that $f(O_x) = 0$, for all $O_x \in X$, and $f(O_y) = 1$, for all $O_y \in Y$.*

Next, the adversary will try to compute f on every input, so it must distinguish oracles in X from oracles in Y.

Definition 15. *Let $\mathcal{A}^O$ be a quantum algorithm with query access to an $O \in \mathcal{O}$. We say that $\mathcal{A}$ **approximately computes** f if for every $O \in \mathcal{O}$, $P[\mathcal{A}^O = f(O)] \geq 2/3$.*

Definition 16. *Let u, u' be upper bounds that satisfy:*

- *For any $O_x \in X$ and any input i to O_x, $P_{O_y \in Y}[O_x(i) \neq O_y(i)] \leq u$.*
- *For any $O_y \in Y$ and any input i to O_y, $P_{O_x \in X}[O_x(i) \neq O_y(i)] \leq u'$.*

Theorem 8 ([Amb02], Thm. 2). *If $\mathcal{A}$ approximately computes f, then $\mathcal{A}$ makes at least $\Omega\left(\frac{1}{\sqrt{u \cdot u'}}\right)$ queries to O.*

Proof of theorem 7

The distinguishing game's format matches the format considered by the adversary method. For a given msk, let X comprise only the full verification oracle, $\{O_{Full}[A]\}$. Let Y comprise all possible franchised verification oracles: $Y = \{O_{Fran}[M(msk)] | M \in \mathcal{M}(A)\}$. And let $\mathcal{O} = X \bigcup Y$. Then f equals b from the distinguishing game.

Next, we will assume that each honest verifier gets at least $t/4$ dimensions of V_{id} and $t/4$ dimensions of W_{id} that are unknown to the adversary. As a result, each verifier accepts a negligible fraction of the vectors in $\mathbb{Z}_2^n$. So it is hard for the adversary to find an $\mathbf{x} \in \mathbb{Z}_2^n$ on which the full and franchised oracles behave differently, which makes distinguishing them hard. The next definition and next two lemmas expand on this argument.

Definition 17. *An $msk \leftarrow \mathsf{Setup}(1^\lambda)$ is **good** if for every $id \in [N - C]$,*

- *$dim[span(V_{adv}, V_{id})] \geq dim(V_{adv}) + t/4$*
- *$dim[span(W_{adv}, W_{id})] \geq dim(W_{adv}) + t/4$*

[7] Our proof is inspired by [AC12].

Lemma 6. *With overwhelming probability in λ, $msk \leftarrow \mathsf{Setup}(1^\lambda)$ is good.*

Proof.

1) With overwhelming probability, $|I_{id} \backslash I_{adv}| \geq t/4$ for all $id \in [N - C]$.
First, $|I_{adv}| \leq Ct = n/4$, so the probability that a uniformly random $i \in [n/2]$ is in I_{adv} is $\leq 1/2$. Then

$$\text{Let } \mu = \mathbb{E}_{I_{id}}[|I_{id} \backslash I_{adv}|] \geq t/2$$

Next we use the multiplicative Chernoff bound:

$$\begin{aligned} P\big[|I_{id} \backslash I_{adv}| \leq t/4\big] &\leq P\big[|I_{id} \backslash I_{adv}| \leq \mu/2\big] \\ &< \left(\frac{e^{-1/2}}{(1/2)^{1/2}}\right)^{\mu} = \left(\frac{2}{e}\right)^{\mu/2} \leq \left(\frac{2}{e}\right)^{t/4} \\ &= \left(\frac{2}{e}\right)^{\Theta(\sqrt{n})} = \mathsf{negl}(\lambda) \end{aligned}$$

Then by the union bound, the probability that $|I_{id} \backslash I_{adv}| \geq t/4$ for all $id \in [N - C]$ is $1 - (N - C) \cdot \mathsf{negl}(\lambda) = 1 - \mathsf{negl}(\lambda)$.

2) For convenience, let's say that $I_{id} \backslash I_{adv} = \big[|I_{id} \backslash I_{adv}|\big]$. Given that $|I_{id} \backslash I_{adv}| \geq t/4$, the following event E occurs with overwhelming probability:

$$E: \quad dim\big[span(V_{adv}, \mathbf{v}_1, \ldots, \mathbf{v}_{t/4})\big] = dim(V_{adv}) + t/4$$

$$\begin{aligned} P_{\{v_i\}_{i \in [t/4]}}(E) &\geq 1 - P(\mathbf{v}_1 \in V_{adv}) - \ldots - P[\mathbf{v}_{t/4} \in span(V_{adv}, \mathbf{v}_1, \ldots, \mathbf{v}_{t/4-1})] \\ &\geq 1 - 2^{n/4-n/2} - \ldots - 2^{n/4+t/4-1-n/2} \\ &\geq 1 - \frac{t}{4} \cdot 2^{(t/4-n/4)} = 1 - 2^{-\Theta(n)} = 1 - \mathsf{negl}(\lambda) \end{aligned}$$

3) Putting together steps 1 and 2, we have that with overwhelming probability in λ,

$$dim\big[span(V_{adv}, V_{id})\big] \geq dim(V_{adv}) + t/4$$

□

Lemma 7. *Let msk be good, let $M \in_R \mathcal{M}(A)$, and let $msk' = M(msk)$. Then for any $id \in [N - C]$ and any $\mathbf{x} \in \mathbb{Z}_2^n$,*

- *If $\mathbf{x} \notin A$, then $P\big(\mathbf{x} \in {W'_{id}}^{\perp}\big) = 2^{-\Omega(\sqrt{n})}$.*
- *If $\mathbf{x} \notin A^\perp$, then $P\big(\mathbf{x} \in {V'_{id}}^{\perp}\big) = 2^{-\Omega(\sqrt{n})}$.*

The probability is over the choice of $M \in_R \mathcal{M}(A)$.

Proof. We'll prove the first claim – the second claim's proof is similar.

1) Let $S = span(\{w_j\}_{j \in J_{id} \backslash J_{adv}})$. This is the random subspace that verifier id has that the adversary cannot predict. We know from lemma 6 that $dim(S) \geq t/4$. Also $M \cdot S \leq W'_{id}$, so $W'^{\perp}_{id} \leq (M \cdot S)^\perp$. Then:

$$P_M\big(\mathbf{x} \in W'_{id}\big) \leq P_M\big(\mathbf{x} \in (M \cdot S)^\perp\big) = P_M\big(\mathbf{x}^T \cdot M \cdot S = \mathbf{0}\big)$$

2) $M^T\mathbf{x}$ is a random vector satisfying $M^T\mathbf{x} \notin A$. First, M^T maps A to A and $A^\perp$ to $A^\perp$. Since $\mathbf{x} \notin A$, $\mathbf{x}$ has a non-zero component in $A^\perp$, which M^T maps to a non-zero component in $A^\perp$. Therefore, $M^T\mathbf{x} \notin A$.

$$\begin{aligned} P_M\big(\mathbf{x}^T \cdot M \cdot S = \mathbf{0}\big) = P_M\big(M^T\mathbf{x} \in S^\perp\big) &\leq \frac{|S^\perp|}{|\mathbb{Z}_2^n \backslash A|} \\ &= \frac{2^{dim(S^\perp)}}{2^n - 2^{n/2}} \leq \frac{2^{n-t/4}}{2^{n-1}} = 2^{1-t/4} = 2^{-\Omega(\sqrt{n})} \end{aligned}$$

□

Lemma 8. *If msk is good, then any quantum algorithm that approximately computes f needs at least $2^{\Omega(\sqrt{n})}$ oracle queries.*

Proof.

1) If O_{Full} and O_{Fran} differ on an input, then O_{Full} rejects the input, and O_{Fran} accepts it.

For any input $(id, s, \mathbf{x})$ to an oracle, if $O_{Full}[A](id, s, \mathbf{x}) = 1$, then $O_{Fran}[M(msk)](id, s, \mathbf{x}) = 1$ as well. When $s = 0$, O_{Full} accepts iff $\mathbf{x} \in A$. Since $A \leq W_{id}^{\perp}$, O_{Fran} accepts as well. Similar reasoning shows that when $s = 1$, if O_{Full} accepts, then O_{Fran} accepts as well.

Therefore, the only way for O_{Full} and O_{Fran} to give different responses to an input is if:

$$O_{Full}[A](id, s, \mathbf{x}) = 0, \text{ and } O_{Fran}[M(msk)](id, s, \mathbf{x}) = 1$$

2) Lemma 7 says that if $O_{Full}[A](id, s, \mathbf{x}) = 0$, then

$$P_{M \leftarrow \mathcal{M}(A)}\Big(O_{Fran}[M(msk)](id, s, \mathbf{x}) = 1\Big) = 2^{-\Omega(\sqrt{n})}$$

so we can set $u = 2^{-\Omega(\sqrt{n})}$. Also, we can set $u' = 1$ because 1 is greater than or equal to any probability.

Finally, in order to approximately compute f, the number of oracle queries needed is $\Omega\Big(\frac{1}{\sqrt{u \cdot u'}}\Big) = 2^{\Omega(\sqrt{n})}$. □

Lemma 9. *For any polynomial-time quantum algorithm $\mathcal{A}$, and any good msk, there exists an $M \in \mathcal{M}(A)$ such that:*

$$\Big|P(\mathcal{A}^{O_{Full}[A]} = 1) - P(\mathcal{A}^{O_{Fran}[M(msk)]} = 1)\Big| \leq 2^{-\Theta(\sqrt[3]{n})}$$

Proof.

1) Let Δ be the minimum value of

$$\Big|P(\mathcal{A}^{O_{Full}[A]} = 1) - P(\mathcal{A}^{O_{Fran}[M(msk)]} = 1)\Big|$$

over all M, and let $p = P(\mathcal{A}^{O_{Full}[A]} = 1)$.

Next, assume toward contradiction that there is some polynomial-time algorithm $\mathcal{A}$ and some good msk such that $\Delta > 2^{-\Theta(\sqrt[3]{n})}$. Then we'll construct an algorithm $\mathcal{A}'$ that approximately computes f using $2^{\Theta(\sqrt[3]{n})}$ queries (by lemma 8, we know this is not possible).

$\mathcal{A}'$ runs $4n/\Delta^2$ independent iterations of $\mathcal{A}$ and averages the outputs. Let $\bar{p}$ be the average number of iterations of $\mathcal{A}$ that output 1. Next, $\mathcal{A}'$ outputs 0 if $|\bar{p} - p| \leq \Delta/2$ and outputs 1 otherwise.

2) $\mathcal{A}'$ gives the incorrect value for f if:

1. $|\bar{p} - p| \leq \Delta/2$, but the oracle is franchised.
2. $|\bar{p} - p| > \Delta/2$, but the oracle is full.

In the first case, $\big|\mathbb{E}[\bar{p}] - p\big| > \Delta$, so $\big|\bar{p} - \mathbb{E}[\bar{p}]\big| \geq \Delta/2$. In the second case as well, $\big|\bar{p} - \mathbb{E}[\bar{p}]\big| \geq \Delta/2$.

The probability of an error is bounded by the Hoeffding inequality:

$$P\Big(\big|\bar{p} - \mathbb{E}[\bar{p}]\big| \geq \Delta/2\Big) \leq 2e^{-2(\Delta/2)^2 \cdot (4n/\Delta^2)} = 2e^{-2n}$$

Next, $\mathcal{A}'$ approximately computes f because for any $O \in \mathcal{O}$, $\mathcal{A}'$ computes $f(O)$ with probability $\geq 1 - 2e^{-2n} > 2/3$.

3) Finally, $\mathcal{A}'$ makes $2^{\Theta(\sqrt[3]{n})}$ queries. First, $\mathcal{A}$ makes $2^{O(\log n)}$ queries because it runs in polynomial time. So the number of queries that $\mathcal{A}'$ makes is:

$$\frac{4n}{\Delta^2} \cdot 2^{O(\log n)} = 2^{O(\log n) + O(\sqrt[3]{n})} = 2^{O(\sqrt[3]{n})}$$

Since no algorithm can approximately compute f using $2^{O(\sqrt[3]{n})}$ queries, this is a contradiction. So for any polynomial-time $\mathcal{A}$, and any good msk, there exists an M such that

$$\left|P(\mathcal{A}^{O_{Full}[A]}=1)-P(\mathcal{A}^{O_{Fran}[M(msk)]}=1)\right|\leq 2^{-\Theta(\sqrt[3]{n})}$$

□

Lemma 10. *For any polynomial-time quantum algorithm $\mathcal{A}$, any good msk, and a uniformly random $M\in_R \mathcal{M}(A)$,*

$$\left|P(\mathcal{A}^{O_{Full}[A]}=1)-P(\mathcal{A}^{O_{Fran}[M(msk)]}=1)\right|\leq 2^{-\Theta(\sqrt[3]{n})}$$

The probability is over $\mathcal{A}$'s randomness and the choice of M.

Note that lemma 10 is equivalent to theorem 7.

Proof. The problem of distinguishing full and franchised oracles is random self-reducible. Since lemma 9 says the algorithm's distinguishing advantage is negligible in the worst case, then their advantage is also negligible in the average case.

Assume toward contradiction that there exists a polynomial-time quantum algorithm $\mathcal{A}$ such that for a uniformly random $M\in_R \mathcal{M}(A)$,

$$\delta := \left|P(\mathcal{A}^{O_{Full}[A]}=1)-P(\mathcal{A}^{O_{Fran}[M(msk)]}=1)\right| = 2^{-o(\sqrt[3]{n})}$$

Then we'll construct a polynomial-time algorithm $\mathcal{A}'$ that runs $\mathcal{A}$ as a subroutine and achieves $\delta = 2^{-o(\sqrt[3]{n})}$ for all M (by lemma 9, this is impossible).

Given any $M\in\mathcal{M}(A)$, $\mathcal{A}'$ samples a uniformly random $R\in_R\mathcal{M}(A)$. Then $R[M(msk)]$ is an "average-case" master secret key in the sense that $R[M(msk)]=(R\cdot M)(msk)$, and $R':=R\cdot M$ is uniformly random in $\mathcal{M}(A)$.

$\mathcal{A}'$ gives msk to $\mathcal{A}$ and simulates the distinguishing game in which the franchised verifiers are using $R[M(msk)]$. Whenever $\mathcal{A}$ queries the oracle, $\mathcal{A}'$ uses R as a change-of-basis for the query before forwarding it to the challenger. In $\mathcal{A}$'s view, it is dealing with a uniformly random $R'\in\mathcal{M}(A)$, so $\mathcal{A}$ has distinguishing advantage δ. Therefore, $\mathcal{A}'$ has the same advantage $\delta=2^{-o(\sqrt[3]{n})}$, but for every M. This contradicts lemma 9, so in fact, lemma 10's claim is true. □

Lemma 10 proves theorem 7.

Acknowledgements

This work is supported in part by NSF. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

This work is also supported by MURI Grant FA9550-18-1-0161 and ONR award N00014-17-1-3025.

We thank Zeph Landau, Umesh Vazirani, and the Princeton Writing Center for helpful feedback on various drafts of this paper.

References

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

[Aar16] Scott Aaronson, 2016. http://www.scottaaronson.com/blog/?p=2854.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. *Proceedings of the Annual ACM Symposium on Theory of Computing*, 03 2012.

[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, June 2002.

[BN08] Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, page 501–510, New York, NY, USA, 2008. Association for Computing Machinery.

[BS20] Amit Behera and Or Sattath. Almost public quantum coins, 2020.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 257–270, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[FGH+12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 276–289, New York, NY, USA, 2012. Association for Computing Machinery.

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 660–670, New York, NY, USA, 2018. Association for Computing Machinery.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Htall/CRC, 2nd edition, 2014.

[KNY21] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions, 2021.

[NC00] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. *American Journal of Physics*, 70(5):558, 2000.

[PFP15] Marta Conde Pena, Jean-Charles Faugère, and Ludovic Perret. Algebraic cryptanalysis of a quantum money scheme the noise-free case. In Jonathan Katz, editor, *Public-Key Cryptography – PKC 2015*, pages 194–213, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[Rob21] Bhaskar Roberts. Security analysis of quantum lightning. Springer-Verlag, 2021.

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

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

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 408–438, Cham, 2019. Springer International Publishing.